3GPP TSG SA WG3 Security — S3#22 **S3-020066**

25 - 28 February, 2002

Bristol, UK

CR-Form-v4

CHANGE REQUEST

⌘ **33.200 CR** ⌘ ev **-** ⌘ Current version: **4.2.0** ⌘

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

Proposed change affects: ⌘ (U)SIM □ ME/UE □ Radio Access Network □ Core Network ☒

Title:	⌘	Automatic Key Management		
Source:	⌘	MAP Rapporteur		
Work item code:	⌘	MAPsec	***Date:*** ⌘	18-02-02
Category:	⌘	**B**	***Release:*** ⌘	Rel-5
		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)*

Reason for change:	⌘	The Release 4 version of the specification only included manual key management. This CR add automatic key management to the specification for the Release 5 version.
Summary of change:	⌘	The change introduces Key Administaration Centres (KACs) and their associated interactions to themselves and other elements to the specification. KACs enable automatic key management.
Consequences if not approved:	⌘	There will be no automatic key management in the specification.

Clauses affected:	⌘	1, 2, 3.2, 3.3, 4, 5.1, 5.2, 5.3, 5.4, 5.6.1, 5.6.2, 7, 8, A, B	
Other specs affected:	⌘	□ Other core specifications □ Test specifications □ O&M Specifications	⌘
Other comments:	⌘		

**************** First Modified Section ****************

1 Scope

This technical specification covers the security mechanisms and procedures necessary to protect the MAP protocol. The complete set of enhancements and extensions to facilitate security protection for the MAP protocol is termed MAPsec and it covers transport security in the MAP protocol itself and the security management procedures.

The security mechanisms specified for MAP are on the application layer. This means that MAPsec is independent of the network and transport protocols to be used. This specification also includes automatic key management mechanisms to update the security associations used to protect the MAP signalling.

This technical specification contains the stage-2 specification for security protection of the MAP protocol. The actual implementation (stage-3) specification can be found in the MAP stage-3 specification, TS 29.002 [4].

~~NOTE: It is explicitly noted that automated key management and key distribution is not part of Rel-4. All key management and key distribution in Rel-4 must therefore be carried out by other means. (See Annex A)~~

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3G TS 21.133: Security Threats and Requirements.

[2] 3G TS 21.905: 3G Vocabulary.

[3] 3G TS 23.060: General Packet Radio Service (GPRS); Service description; Stage 2.

[4] 3G TS 29.002: Mobile Application Part (MAP) specification.

[5] NIST Special Publication 800-XX "Recommendation for Block Cipher Modes of Operation" July 2001.

[6] ISO/IEC 9797: "Information technology -- Security techniques -- Message Authentication Codes (MACs) -- Part 1: Mechanisms using a block cipher", Ed.1, 1999-12-16.

[7] draft-arkko-map-doi-04-pa2.txt: The MAP Security Domain of Interpretation for ISAKMP

***************** Next Modified Section *****************

3.2 Symbols

For the purposes of the present document, the following symbols apply:

f6	MAP encryption algorithm.
f7	MAP integrity algorithm.
Zd	MAPsec interface between KACs belonging to different PLMNs
Ze	MAPsec interface between KACs and MAP-NEs within the same PLMN
Zf	The MAP application layer security interface between MAP-NEs engaged in security protected signalling.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AES	Advanced Encryption Standard
DoI	Domain of Interpretation
ESP	Encapsulating Security Payload
FALLBACK	Fallback to unprotected mode indicator
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
IP	Internet Protocol
IPsec	IP security - a collection of protocols and algorithms for IP security incl. key mngt.
ISAKMP	Internet Security Association Key Management Protocols
IV	Initialisation Vector
KAC	Key Administration Centre
~~MEK~~	~~MAP Encryption Key~~
MAC	Message Authentication Code
MAC-M	MAC used for MAP
MAP	Mobile Application Part
MAP-NE	MAP Network Element
MAPsec	MAP security – the MAP security protocol suite
MEA	MAP Encryption Algorithm identifier
MEK	MAP Encryption Key
MIA	MAP Integrity Algorithm identifier
MIK	MAP Integrity Key
NDS	Network Domain Security
NE	Network Entity
PPI	Protection Profile Indicator
PPRI	Protection Profile Revision Identifier
PROP	Proprietary field
SA	Security Association
SADB	Security Association DataBase (also referred to as SAD)
SPD	Security Policy Database (sometimes also referred to as SPDB)
SPI	Security Parameters Index
TVP	Time Variant Parameter

***************** Next Modified Section *****************

4 Principles of MAP application layer security

This technical specification defines mechanisms for protecting the MAP protocol at the application layer. The MAP protocol may also be protected at the network layer when IP is used as the transport protocol. However, whenever inter-working with networks using SS7-based transport is necessary, protection at the application layer shall be used.

The security measures specified in this TS are only fully useful if all interconnected operators use them. In order to prevent active attacks all interconnected operators must at least use MAPsec with the suitable protection levels as indicated in this specification and treat the reception of all MAP messages (protected and unprotected) in a uniform way in the receiving direction.

Before protection can be applied, Security Associations (SA) needs to be established between the respective MAP network elements. Security associations define, among other things, which keys, algorithms, and protection profiles to use to protect MAP signalling. The necessary MAP-SAs between networks are negotiated between the respective Key Administration Centres (KACs) of the networks~~network operators~~. The negotiated SA will be effective PLMN-wide and distributed to all network elements which implement MAP application layer security within the PLMN. Signalling traffic protected at the application layer will, for routing purposes, be indistinguishable from unprotected traffic to all parties except for the sending and receiving entities.

Figure 1 gives an overview of the architecture used for MAPsec.

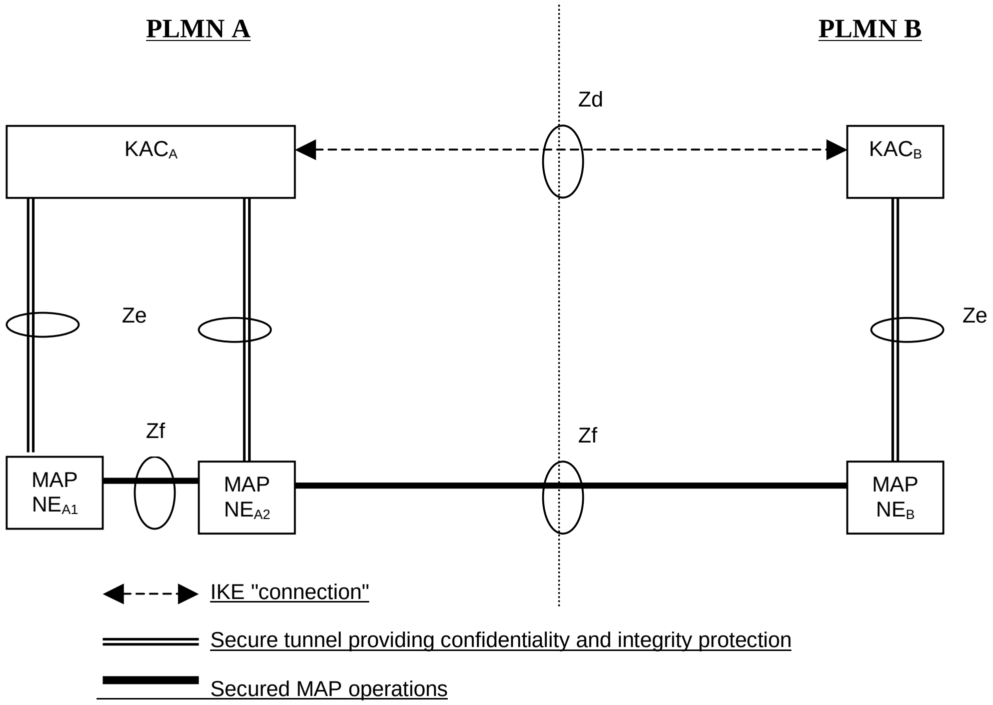

Figure 1: Overview of the Zd, Ze and Zf interfaces

The following interfaces are defined MAPsec.

- **Zd-interface (KAC-KAC)**

  The Zd-interface is used to negotiate MAPsec Security Associations (SAs) between PLMNs. The traffic over Zd consists only of IKE negotiations. The negotiated MAPsec SAs are valid on a PLMN to PLMN basis.

- **Ze-interface (KAC-NE)**

The Ze-interface is located between MAP-NEs and a KAC from the same PLMN. This interface is used for transport of MAPsec SAs and the relevant security policy information from the KAC to the MAP-NE. The KAC and the MAP-NE are able to establish and maintain a secure tunnel between them. Whether the tunnel is established when needed or a priori is for the PLMN operator to decide.

- **The Zf-interface (NE-NE)**

The Zf-interface is located between MAP-NEs. The MAP-NEs may be from the same PLMN or from different PLMNs (as shown in figure 1). The MAP-NEs use MAPsec SAs received from a KAC to protect the MAP operations. The MAP operations within the MAP dialogue are protected selectively as specified in the applied MAPsec protection profile. The interface applies to all MAPsec transactions, intra- or inter-PLMN.

~~Protection at the application layer implies changes to the application protocol itself to allow for the necessary security functionality to be added.~~

~~The MAP application layer security interface between MAP-NEs engaged in security protected signalling is referred to in this specification as the Zf interface. The interface applies to all MAPsec transactions, intra- or inter-PLMN.~~

The security services provided by MAPsec are:

- data integrity;
- data origin authentication;
- anti-replay protection;
- confidentiality (optional).

Annex B includes detailed procedures on how secure MAP signalling is performed between two MAP-NEs.

5 MAP security (MAPsec)

5.1 Properties and tasks of Key Administration Centres (KACs)~~Security services provided by MAPsec~~

~~The security services provided by MAPsec are:~~

- ~~data integrity;~~
- ~~data origin authentication;~~
- ~~anti-replay protection;~~
- ~~confidentiality (optional).~~

Key Administration Centres (KACs) are entities that are used for negotiating MAPsec SAs on behalf of MAP-NEs. The KACs are defined to handle communication over these interfaces:

- the Zd-interface, which is located between KACs from different PLMNs. The IKE protocol with support for MAPsec DoI shall be used over this interface.
- the Ze-interface, which is located between a KAC and a MAP-NE within the same PLMN is used to transfer MAPsec SAs and security policy from KACs to MAP-NEs. The SAs and security policy must be transferred in a secure manner.

When a MAP-NE needs to establish a secure connection towards another MAP-NEs it will request a MAPsec SA from the KAC if it cannot find any appropriate MAPsec SA in its local SAD. The KAC will then either provide an existing MAPsec SA or negotiate a new MAPsec SA, before returning the MAPsec SA to the MAP-NE.

A MAPsec SA is valid for all MAP communications between the two PLMNs for which it is negotiated. That is, the same MAPsec SA shall be provided to all MAP-NEs in PLMN A for communication with MAP-NEs in PLMN B. Each PLMN can have one or more KACs. Each KAC will be responsible to define MAPsec SAs with a well-defined set of reachable PLMNs. The number of KACs in a PLMN will depend on the need to differentiate between the externally reachable destinations, the need to balance the traffic load and to avoid single points of failure.

KACs perform the following operations:

- Negotiate SAs for MAPsec with other KACs belonging to other network operators. This action is triggered either by request for a MAPsec SA by a NE or by policy enforcement when MAPsec SAs should always be available. The negotiation of MAPsec SAs is performed at Zd-interface using IKE protocol with MAPsec DoI.
- Convert specific negotiated SA parameter so that they can be understood by NEs. In particular, the KAC shall convert negotiated SA duration in seconds as negotiated in MAPsec DoI to UTC absolute time format.
- Perform refresh of MAPsec SAs. This could be triggered internally by SA lifetime supervision depending on the policies set by the operator.
- Distribute MAPsec SAs and policy information to NEs belonging to the same PLMN as the KAC.
- KAC may be able to establish secure connections to transmit MAPsec SAs and policy information to the NEs within its PLMN.

KACs are also responsible for the maintenance of the following databases:

- KAC-SPDB-MAP: Defines the scope, the security policy, in which MAP-SAs may be negotiated (e.g. allowed MAP-PPs, Algorithms, SA-lifetimes). This database is updated on operator initiative in the framework of the roaming agreements.
- NE-SPD-MAP: A database in a KAC containing the MAP security policy information that will be used by an NE in protecting MAP messages (e.g. value of “Fallback to unprotected Mode Indicator” and table of protected MAPsec operation components). This is held to update the NEs.
- NE-SADB-MAP: A database in a KAC containing MAP-SA information. This is held to allow the KAC to update the NE.

KACs are responsible for security sensitive operations and shall be physically secured. They shall offer capabilities for the secure storage of long-term keys used for IKE authentication.

5.2 Properties and tasks of MAPsec enabled network elements

MAPsec MAP-NEs shall maintain the following databases:

- NE-SPD-MAP: A database in an NE containing MAP security policy information (see clause 5.3);
- NE-SADB-MAP: A database in an NE containing MAP-SA information. MAP-NEs shall monitor the SA hard expiry time and expired SAs shall be deleted from the database (see clause 5.4).

MAPsec MAP-NEs shall be able to perform the following operations:

- Secure MAP signalling (i.e. send/receive protected or unprotected messages) according to information in NE-SPD-MAP and NE-SADB-MAP. The structure of protected messages is defined in clause 5.5 and the protection algorithms are defined in clause 5.6.
- Communicate with the KAC in the same PLMN in order that the NE-SPD-MAP and NE-SADB-MAP in the NE can be updated.
- NE must be able to establish a secure connection to receive MAPsec SAs and policy information from the KAC within its PLMN.

5.3 Policy requirements for the MAPsec Security Policy Databases (SPD)

Two security policy databases, KAC-MAP-SPD and NE-MAP-SPD, are required to implement MAPsec. KAC-MAP-SPD holds the information needed by the KACs to negotiate MAPsec SAs. NE-SPD-MAP holds the security policy information used by a NE element when applying MAPsec to message over the Zf-interface. A KAC holds a copy of NE-SPD-MAP in order to update the NEs in the same PLMN.

Editor's note: Do the elements in KAC-SPD-MAP need to be defined?

~~The security policies for MAPsec key management are specified in the NE's SPD~~. NE-SPD-MAP entries define which MAP operation components are protected and which MAP SAs (if any) to use to protect MAP signalling based on the PLMN of the peer NE. There can be no local security policy definitions for individual NEs. Instead, SPD entries of different NEs within the same PLMN shall be identical.

Fallback to unprotected mode:

- The "fallback to unprotected mode" (enabled/disabled) shall be available to the MAP-NE before any communication towards other MAP-NEs can take place. For the receiving direction, it is sufficient to have a single parameter indicating whether fallback for incoming messages is allowed or not. For the sending direction, the information should indicate for each destination PLMN whether fallback for outgoing messages is allowed or not;
- The use of the fallback indicators is specified in Annex B;
- The security measures specified in this TS are only fully useful for a particular PLMN if it disallows fallback to unprotected mode for MAP messages received from any other PLMN.

Table of MAPsec operation components:

- The security policy database (SPD) shall contain a table of MAPsec operation components for incoming messages. This table contains operation components which have to be carried in MAPsec messages with Protection Mode 1 or 2. The use of MAPsec operation components is specified in Annex B.

Uniformity of protection profiles:

- In order to ensure full protection, a particular PLMN shall use the same protection profile for incoming MAPsec messages from all other PLMNs. In particular, full protection is not ensured when protection profile A (no protection) is used for some source PLMNs and other profiles are used for other source PLMNs.

Explicit policy configuration:

- The SPD shall contain an entry for each PLMN the MAP-NE is allowed to communicate with.

Editor's note: Some issues need to be investigated: Non-synchronised expiration times issue, mechanism to distinguish inbound/outbound SPDs~~-~~? The exact entries in NE-SPD-MAP may need defining as this information is passed over the Ze-interface.

5.4 MAPsec security association attribute definition

The MAPsec security association shall contain the following data elements:

- **Destination PLMN-Id:**

  PLMN-Id is the ID number of the receiving Public Land Mobile Network (PLMN). The value for the PLMN-Id is a concatenation of the Mobile Country Code (MCC) and Mobile Network Code (MNC) of the receiving network.

- **Security Parameters Index (SPI):**

SPI is a 32-bit value that is used in combination with Destination PLMN-Id to uniquely identify a MAP-SA.

- **Sending PLMN-Id:**

PLMN-Id is the ID number of the sending Public Land Mobile Network (PLMN). The value for the PLMN-Id is a concatenation of the Mobile Country Code (MCC) and Mobile Network Code (MNC) of the sending network.

- **MAP Encryption Algorithm identifier (MEA):**

  Identifies the encryption algorithm. Mode of operation of algorithm is implicitly defined by the algorithm identifier. Mapping of algorithm identifiers is defined in clause 5.6.

- **MAP Encryption Key (MEK):**

  Contains the encryption key. Length is defined according to the algorithm identifier.

- **MAP Integrity Algorithm identifier (MIA):**

  Identifies the integrity algorithm. Mode of operation of algorithm is implicitly defined by the algorithm identifier. Mapping of algorithm identifiers is defined in section 5.6.

- **MAP Integrity Key (MIK):**

  Contains the integrity key. Length is defined according to the algorithm identifier.

- Protection Profile Revision Identifier (PPRI):

  Contains the revision number of the PPI. Length is 8 bits. PPRI-values are defined in section 6.3.

- **Protection Profile Identifier (PPI):**

  Identifies the protection profile. Length is 16 bits. Mapping of profile identifiers is defined in section 6.

- **SA Hard Expiry Time:**

  Defines the actual expiry time of the SA. The hard expiry time shall be given in UTC time.

- **SA Soft Expiry Time:**

  Defines soft expiry time of the SA for outbound traffic. The soft expiry time shall be given in UTC time.

Editor’s Note: The exact format and length to be defined.

After the hard expiry time has been reached the SA shall no longer be used for inbound or outbound traffic. When the soft expiry time is reached, the SA shall not be used any longer for the outbound traffic unless no other valid SA exists.

A MAPsec SA is uniquely identified by a destination PLMN-Id and a Security Parameters Index, SPI. As a consequence, during SA creation, the SPI is always chosen by the receiving side (i.e. the SPI for MAP communications from PLMN A to PLMN B is selected by KAC in PLMN B).

If the SA is to indicate that MAPsec is not to be applied then all the algorithm attributes shall contain a NULL value.

***************** Next Modified Section *****************

5.6.1 Mapping of MAPsec-SA encryption algorithm identifiers

The MEA algorithm indication fields in the MAPsec-SA are used to identify the encryption algorithm and algorithm mode to be used. The mapping of algorithm identifiers is defined below.

Table 1: MAP encryption algorithm identifiers

MAP Encryption Algorithm identifier	Description
0	Null
1	AES in counter mode with 128-bit key length (MANDATORY)
:	-not yet assigned-
15	-not yet assigned-

5.6.1.1 Description of MEA-1

The MEA-1 algorithm is AES used in counter mode with a 128-bit key and 128-bit counter blocks as described is the in clause 5.5 of FIPS 800-XX Recommendation for Block Cipher Modes of Operation [5]. The initial counter block T_1 is initialized with IV. Successive counter blocks T_j (J>1) are derived by applying an incrementing function over the entire block T_{j-1} (J>=2) (see Appendix B.1: The standard incrementing function of [5]).

The MAPsec cleartext shall be cut into P_j blocks of 128 bits . If the last block P_n has less than 128-bits (z bits), then it shall be encrypted by bitwise addition with only the first z bits of output block n (Clause 5.5 of [5]).

5.6.2 Mapping of MAPsec-SA integrity algorithm identifiers

The MIA algorithm indication fields in the MAPsec-SA are used to identify the integrity algorithm and algorithm mode to be used. The mapping of algorithm identifiers is defined below.

Table 2: MAP integrity algorithm identifiers

MAP Integrity Algorithm identifier	Description
0	Null
1	AES in a CBC MAC mode with a 128-bit key (MANDATORY)
:	-not yet assigned-
15	-not yet assigned-

5.6.2.1 Description of MIA-1

The MIA-1 algorithm is the ISO/IEC 9797 Part 1: padding method 2, MAC algorithm 1 (initial transformation=1, output transformation=1). No IV used. The MAC-length m is 32-bits (see clause 5.6.1). See ISO/IEC 9797 [6] for more information.

***************** Next Modified Section *****************

7 Security Association and Key Management Procedures

This clause contains details on the procedures used by the KACs to negotiate new SAs over the Zd interface.

7.1 Inter-PLMN Procedure

KACs from different PLMNs will use MAPsec DoI [7] and IKE to set up MAPsec SAs for use between their PLMNs. The information needed to negotiate the new SAs is held in the KAC-SPD-MAP. The KAC shall assign the MAP Encryption Algorithm Identifier onto the MAPsec DoI TransformID [7] when negotiating a new pair of MAPsec-SA. Similarly the KAC shall assign the MAP Integrity Algorithm Identifier onto the authentication algorithm attribute of the SA [7] for IKE phase 2 when negotiating a new pair of MAPsec-SA. The details of these assignments are ffs. The KAC shall not use the Key Length or Key Rounds Attributes of the SA for IKE phase 2 as this information is implicitly available for the partner KAC via the used TransFormID. The negotiated SA Life Duration shall be transformed into the Hard Expiry Time in the MAPsec SA. The details of this transformation are ffs. It should be noted that although part of an SA, the Soft Expiry Time is not negotiated by the KACs. The value of Soft Expiry Time is set by local policy in the KAC.

Editor's note: The mapping between MAPsec DoI and MAPsec SA algorithms needs specifying. The method of transforming SA Life Duration into Hard Expiry Time needs specifying.

7.2 Intra-PLMN Prodecure

A KAC must create new MAPsec SAs to enable communication between NEs in its PLMN. It does this using the data stored in KAC-MAP-SPD except it must transform the SA Lifetime given in KAC-SPD-MAP into a MAPsec Hard Expiry Time and generate the relevant cryptographic keys. These methods are ffs. The value of Soft Expiry Time is set by local policy in the KAC.

Editor's note: The method of transforming SA Lifetime into Hard Expiry Time needs specifying. The method for generating the cryptographic keys needs specifying.

8 Local Security Association and Policy Distribution

The KAC transmits SAs and security policy information to the NEs in the same PLMN. The method of transmitting this data to the NEs must satisfy the following requirements

- The push mechanism requires the KAC to maintain active SAs with all other networks that MAP signalling is exchanged with. The KAC internally supervises the SA lifetime and performs automatic SA renewal with all other networks that MAP signalling is exchanged with.

- If new SAs are available, the KAC distributes these SAs to the NEs that require them, according to the KAC policy. The KAC must provide fresh SAs to its NEs for a specific network in time before the soft expiry time of the current active SA has been reached.

- The NEs must ensure that only Ze-messages sent by a KAC in the same security domain are accepted. Depending on the security domain policy, the NE may only accept Ze-messages from one or more specific KACs within the domain, out of the set of KACs in that domain.

- The KAC must ensure that only Ze-messages sent by a NE in the same security domain are accepted (Otherwise an attacker could, e.g. by sending false messages, force the KAC to overload the NE's)

- The KAC shall be able to know whether the MAP-NE needs only MAP-SA’s for security domain internal communication or not.
- All messages over Ze should be integrity and confidentiality protected, as they contain sensitive data or trigger messages containing sensitive data, although integrity protection may be sufficient for trigger messages.

8.1 SA Distribution Procedure

Editor's note: The aim of the text below is to provide a functional description of the Ze-interface as a starting point for the stage 3 specification. The full details of the interface are ffs and should be resolved in the stage 3 work.

This clause describes message flows for an extended push mechanism for MAPsec SA distribution over the Ze interface.

Case 1: Initial registration and SA distribution

To initiate communication with the KAC, each NE that requires MAPsec SAs initially registers with the KAC to obtain these SAs. The KAC then, according to the KAC policy, pushes the full set of active SAs that the NE requires (defined by network policy) to the NE, such that the NE receives all required SAs directly after registration. The "action" indicator is set to "REPLACE" to indicate that the NE shall use the SA_List as its new set of SAs, replacing all SAs that are currently in the NE's SADB.

The NE acknowledges the ZE_pushSA message after successfully installing the SAs in its SADB. Otherwise it responds with an error message (tbd).

```
KAC                                                          NE

        <----- ZE_register(NE_ID) -------------------

         ----- ZE_pushSA(action=REPLACE, SA_List) --->

        <----- ZE_ack(NE_ID, [error]) ---------------
```

Case 2: Subsequent SA distribution (normal case)

The KAC sends a ZE_pushSA message when a new SA is negotiated by the KAC to all NEs that require this SA. It is allowed to send several SAs as a list within a single ZE_pushSA message. The "action" indicator is set to "ADD" to indicate that the NE shall add the SA_List to its current SADB.
The NE acknowledges the ZE_pushSA message after successfully installing the SAs in its SADB. Otherwise it responds with an error message (tbd).

```
KAC                                                        NE

         ----- ZE_pushSA(action= ADD, SA_List) --->

        <----- ZE_ack(NE_ID, [error]) ------------
```

Note that the message format for the initial ZE_pushSA and the ZE_pushSA for SA updates is the same for both operations, since the parameter "SA_List" represents a single or a list of SAs.

Case 3: Handling of inconsistent NE states

In case of any event in a NE that leads to an inconsistent SA database in a NE, this NE sends a new register message, and receives the full set of active SAs from the KAC. The KAC, when receiving a new registration of an already registered NE, just updates any old registration with the new one.

Note that this step is considered to be relatively unlikely.

Case 4: SA revocation/removal

In case the revocation of an SA is required, the KAC sends a ZE_pushSA message identifying a list of the SAs to be removed to all NEs currently using them (the parameter SA_ID is tbd., but must uniquely identify the SA to be revoked. For example this can be the combination of SPI and destination PLMN_ID). A list of SAs may be added within the same ZE_pushSA (replacing the removed SAs). The "action" indicator is set to "REMOVE" to indicate that the NE shall delete the list of SAs identified in SA_ID_List from its current SADB, and the action "ADD" may be subsequently used to add the SAs contained in SA_List to its SADB.
The NE acknowledges the ZE_pushSA message after successfully removing the SAs from its SADB. Otherwise it responds with an error message (tbd).

```
KAC                                                           NE

   ---------- ZE_pushSA(action=REMOVE, SA_ID_List,

                        action=ADD, SA_List) ---------------->

   <------------------ ZE_ack(NE_ID) ------------------------
```

Notes:

1) As the NEs in this push model do not have the option to request specific SAs, it is necessary to transmit the complete set of SAs from the KAC to the NE in cases 1 and 3.

2) The replacement of a compromised SA which needs to be revoked, described in case 4, could alternatively be done in two steps: by first sending a revoke operation, followed by an add operation as described in case 2.

3) The extended push mechanism, as described above, may be used to distribute policy information from the KAC to the NEs in a quite similar fashion, by using a SP_List, security policy list parameter instead of the SA_List parameter.

***************** Next Modified Section *****************

Annex A (informative): Guidelines for manual key management

A.1 ~~Inter-domain Security Association and Key Management Procedures~~Void

~~Manual Inter-domain Security Association and Key Management procedures is subject to roaming agreements.~~

~~Some important parts of an inter-domain Security Association and Key Management agreement is:~~

- ~~to define how to carry out the initial exchange of MAPsec SAs;~~
- ~~to define how to renew the MAPsec SAs;~~
- ~~to define how to withdraw MAPsec SAs (including requirements on how fast to execute the withdrawal);~~
- ~~to decide if fallback to unprotected mode is to be allowed;~~
- ~~to decide on key lengths, algorithms, protection profiles, and SA expiry times, etc (MAPsec SAs are expected to be fairly long lived).~~

~~An SA being used by an NE for incoming traffic expires when it reaches its hard expiry time. When this occurs, the NE can no longer use that SA to process incoming MAPsec traffic. If a new additional valid SA is installed into the NE, the "old" one must still be kept by the NE until it reaches its hard expiry time, so as to be able to accept incoming traffic still received under the "old" SA.~~

~~An SA being used by an NE for outgoing traffic expires when it reaches its soft expiry time. When this occurs, the NE must start using another valid SA. If no such valid SA exists, the NE continues to use the "old" SA until it reaches its hard expiry time or another valid SA effectively becomes available.~~

~~In case the current SA gets compromised, a new valid SA should be made immediately available to the NE, which should then stop using the compromised SA and delete it.~~

~~To ease SA renewal, both PLMNs may decide to set up several MAPsec SAs in advance so that NEs can automatically switch from one SA to another SA. In such a situation, the MAPsec SAs would have different soft and hard expiry times.~~

~~When more than one valid SA is available, the NE chooses the one for which the soft expiry time will be reached next.~~

~~A.2 Local Security Association Distribution~~

~~Manual Local Security Association Distribution is executed entirely within one PLMN and is consequently at the discretion of the administrative authority.~~

~~The requirement on the manual distribution procedures can be summarized as follows:~~

- ~~Procedures for transporting the relevant MAPsec SA to the MAP-NEs must be defined. In order to ensure that the MAPsec SA are present when needed, all valid MAPsec SA should be distributed to all MAP-NEs as soon as they are available.~~
- ~~Procedures for revocation of MAPsec SAs must be defined.~~

Annex B (normative): MAPsec message flows

Imagine a network scenario with two MAP-NEs at different PLMNs (NEa and NEb) willing to communicate using MAPsec. Figure 1 presents the message flow.

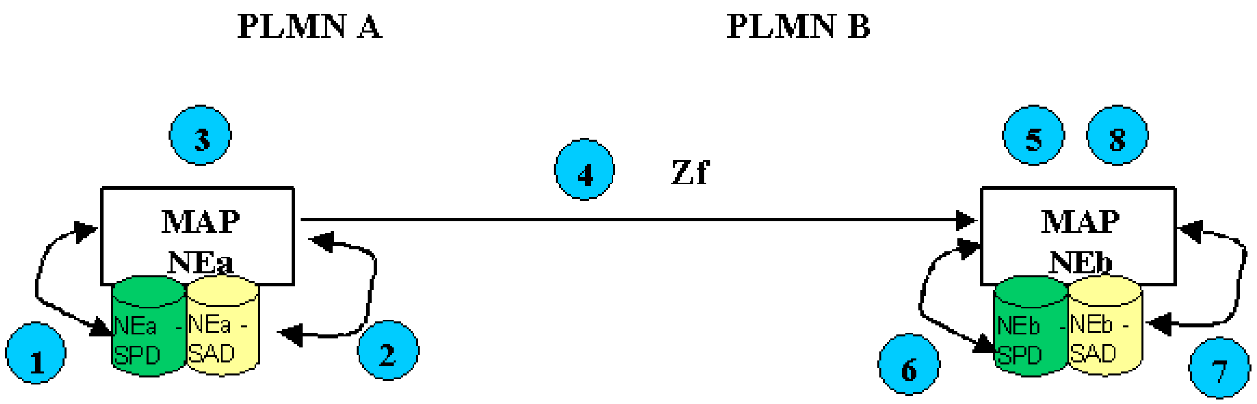

Figure 1. MAPsec Message Flow

Editor's Note: Message flows need to be changed to account for automatic key management

According to Figure 1, when MAP-NEa (NEa) from PLMN A wishes to communicate with a MAP-NEb (NEb) of PLMN B using MAP protocol, the process is the following:

As the Sending Entity, NEa performs the following actions during the outbound processing of every MAP message:

1. NEa checks its Security Policy Database (SPD) to check if MAP security mechanisms shall be applied towards PLMN B:

   a) If the SPD does not mandate the use of MAPsec towards PLMN B, then normal MAP communication procedures will be used and the process continues in step 4.b.

   b) If the SPD mandates the use of MAPsec towards PLMN B, then the process continues at step 2.

   c) If no valid entry in the SPD is found for PLMN B, then the communication is aborted and an error is returned to the MAP user.

2. NEa checks its Security Association Database (SAD) for a valid Security Association (SA) to be used towards PLMN B. In the case where more than one valid SA is available at the SAD, NEa shall choose the one, the soft expiry time of which will be reached next.

   a) In case protection of MAP messages towards PLMN B is not possible (e.g. no SA available, invalid SA…), then the communication is aborted and an error is returned to MAP user.

   b) If a valid SA exists but the MAP dialogue being handled does not require protection (Protection Mode 0 applies to all the components of the dialogue), then either the original MAP message in cleartext is sent in step 4.b, or a MAPsec message with Protection Mode 0 is created in step 3.

   c) If a valid SA exists and the MAP dialogue being handled requires protection, then the process continues at step 3.

3. NEa constructs the MAPsec message towards NEb using the parameters (keys, algorithms and protection profiles) found in the SA.

4. NEa generates either:

   a) MAPsec message towards NEb.

b) An unprotected MAP message in the event that the SPD towards NEb or protection profiles for that specific MAP dialogue so allows it (1.a. or 2.b.).

At the Receiving Entity, NEb performs the following actions during the inbound processing of every MAP message it received:

5. If an unprotected MAP message is received, the process continues with step 6.

   Otherwise, NEb decomposes the received MAPsec message and retrieves SPI and Original component Id from the security header.

6. NEb checks the SPD:

   An unprotected MAP message is received:

   a) If an unprotected MAP message is received and fallback to unprotected mode is allowed, then the unprotected MAP message is simply processed (Process goes to END)

   b) If an unprotected MAP message is received and the ‘MAPsec operation components table’ of the SPD does not mandate the use of MAPsec for the included ‘Original Component Identifier’, then the unprotected MAP message is simply processed (Process goes to END)

   c) If an unprotected MAP message is received, the ‘MAPsec operation components table’ of the SPD mandates the use of MAPsec for the included ‘Original Component Identifier’ and fallback to unprotected mode is NOT allowed, then the message is discarded.

   If the MAP dialogue is still open and it is waiting for an answer, NEb also reports an error back to NEa.

   A MAPsec message is received, NEb checks SPI in the SPD:

   d) If SPI is not in SPD or there is no valid entry for the PLMN associated with SPI in the SPD, then the message is discarded and an error is reported to MAP user.

      If the MAP dialogue is still open and it is waiting for an answer, NEb also reports an error back to NEa.

   e) If a MAPsec message is received, but the SPD indicates that MAPsec is NOT to be used, then the message is discarded and an error is reported to MAP user.

      If the MAP dialogue is still open and it is waiting for an answer, NEb also reports an error back to NEa.

   f) If a MAPsec message is received and the SPD indicates that MAPsec is required, then the process continues at step 7.

7. NEb checks its SAD to retrieve the relevant SA-information for processing of the MAPsec message:

   a) If the received SPI points to a valid SA, then NEb uses the ‘Original Component Identifier’ in the MAPsec header to identify the protection level that has to be applied to the component indicated, according to the protection profile indicated in the SA. If Protection Mode 0 was applied, then the MAP message is simply processed (Process goes to END). Otherwise The process continues at step 8.

   b) If the received SPI does not point to a valid SA, the message is discarded and an error is reported to MAP user. If the MAP dialogue is still open and it is waiting for an answer, NEb also reports an error back to NEa.

8. Freshness of the protected message is checked by ensuring the Time Variant Parameter (TVP) is in an acceptable window. Integrity and encryption mechanisms are applied to the message according to the identified protection level, by using the information in the SA (Keys, algorithms).

   a) If the result after applying such mechanisms is NOT successful then the message is discarded and an error is reported to MAP user. If the MAP dialogue is still open and it is waiting for an answer, NEb also reports an error back to NEa.

   b) If the result after applying such procedures is successful, then NEb has the cleartext MAP message NEa originally wanted to send NEb. The cleartext MAP message can now be processed (Process goes to END)

END: A cleartext MAP message is available at NEb.

In the event the received message at NEb requires an answer to NEa (Return Result/Error), NEb will perform the process in steps 1 to 4 acting as the Sender and NEa will perform the process in steps 5 to 8 acting as the Receiver.

In the event a MAPsec enabled NE initiated a secured MAP communication towards a non-MAPsec enabled NE and the MAPsec enabled NE received an error indication of such circumstance (i.e. "ApplicationContextNotSupported"). The MAPsec enabled NE shall check whether "Fallback to Unprotected Mode" is allowed:

- If NOT allowed, then the communication is aborted.
- If allowed, then the MAPsec enabled NE shall send an unprotected MAP message instead.

The same procedures shall apply to secure MAP communications between MAP-NEs in the same PLMN.

NOTE: Because various error cases may be caused by active attacks, it is highly recommended that the cases are reported to the management system.