3GPP TR 33.828 V11.1.0 (2012-09)

Technical Report

3rd Generation Partnership Project;
Technical Specification Group Services and System Aspects;
IP Multimedia Subsystem (IMS) media plane security
(Release 11)

Keywords

security, IP, Multimedia, SIP

3GPP

Postal address

3GPP support office address

650 Route des Lucioles – Sophia Antipolis
Valbonne – FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

Contents

Foreword

This Technical Report has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

Y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present document studies use cases, requirements and candidate solutions for protecting the IMS media plane against eavesdropping and undetected modification. Currently IMS media protection relies on security provided at the lower layers. With Common IMS, it has become possible to use IMS over a wide variety of access networks which provide varying levels of security and in some cases no security at all. It is therefore desirable to study solutions for securing the IMS media plane in a uniform manner across all access networks. Furthermore, media transport in the core network, although generally less vulnerable than in the access network, may also be realised in varying ways with different levels of security. Therefore, the present document also studies solutions for end-to-end protection of IMS media.

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] 3GPP TR 21.905: Vocabulary for 3GPP Specifications

[2] IETF RFC 5479: Requirements and Analysis of Media Security Management Protocols

[3] 3GPP TS 23.228: IP Multimedia Subsystem (IMS); Stage 2

[4] 3GPP TS 26.234: Transparent end-to-end Packet-switched Streaming Service (PSS)

[5] IETF RFC 4975: The Message Session Relay Protocol (MSRP)

[6] IETF RFC 4976: Relay Extensions for the Message Sessions Relay Protocol (MSRP)

[7] IETF RFC 4346: The Transport Layer Security (TLS) Protocol Version 1.1

[8] IETF RFC 4279: Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)

[9] IETF RFC 3851: Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1

[10] IETF RFC 4347: Datagram Transport Layer Security

[11] IETF Internet-Draft draft-ietf-avt-dtls-srtp-06: Datagram Transport Layer Security (DTLS) Extension to Establish Keys for Secure Real-time Transport Protocol (SRTP) (work in progress)

[12] IETF Internet-Draft draft-ietf-sip-dtls-srtp-framework-05: Framework for Establishing an SRTP Security Context using DTLS (work in progress)

[13] IETF RFC 4474: Enhancements for Authenticated Identity Management in the Session Initiation Protocol

[14] IETF RFC 4916: Connected Identity in the Session Initiation Protocol (SIP)

[15] E.-J. Goh, D. Boneh, P. Golle, B. Pinkas, “The Design and Implementation of Protocol-Based Hidden Key Recovery”, 2003
http://crypto.stanford.edu/~pgolle/papers/escrow.pdf

[16] IETF Internet-Draft draft-wing-sipping-srtp-key-04: Secure Media Recording and Transcoding with the Session Initiation Protocol (work in progress, expired)

[17] IETF Internet-Draft draft-wing-avt-dtls-srtp-key-transport-02: DTLS-SRTP Key Transport (work in progress)

[18] 3GPP TR 23.894: System enhancements for the use of IMS services in local breakout and optimal routing of media

[19] 3GPP TS 33.210: 3G security: Network Domain Security (NDS): IP network layer security

[20] IETF Internet-Draft draft-cakulev-mikey-ibake-00: Identity-Based Mode of Key Distribution in Multimedia Internet KEYing (work in progress)

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in TR 21.905 [1].

Application layer security: In the context of the present document, application layer security is security applied on payload data and it is independent of the transport mechanism used.

Channel security: In the context of the present document, channel security is security applied on data and it is dependent of used transport mechanism or transport identities.

IMS User Equipment: User equipment used for IMS media communications over access networks. The presence of the UICC in this equipment is optional when the equipment does not support any 3GPP access technology. In the case where the user equipment is used for IMS media communications over any 3GPP access network, the IMS User Equipment shall contain a UICC.

3.2 Symbols

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

<symbol>	<Explanation>

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 21.905 [1].

NSPS	National Security and Public Safety
QoE	Quality of Experience

4 Use cases

4.1 Usage models

4.1.1 General

IMS media security may serve different purposes and its relevance for different user groups may vary according to its design and features. The main division of users into groups that may have different requirements on an IMS media security solution is: the general public (private persons), enterprise users, and users from National Security, Public Safety or other governmental organizations (NSPS users).

A first purpose could be to have secure media over all access networks, a second could be to specify an end-to-end media security solution to satisfy the general public, while a third could be to provide high quality end-to-end media security for important user groups like enterprises, National Security and Public Safety (NSPS) organizations and different government authorities, etc.

It should be noted that the protocols for the actual media plane protection are uncontroversial as the working assumption is to use well established protocols like SRTP and PSK-TLS. Thus the open issues are with respect to how the key management solution is designed and where the end-points for the media protection are located. Figure 1 gives an overview of the security endpoints that may be involved.

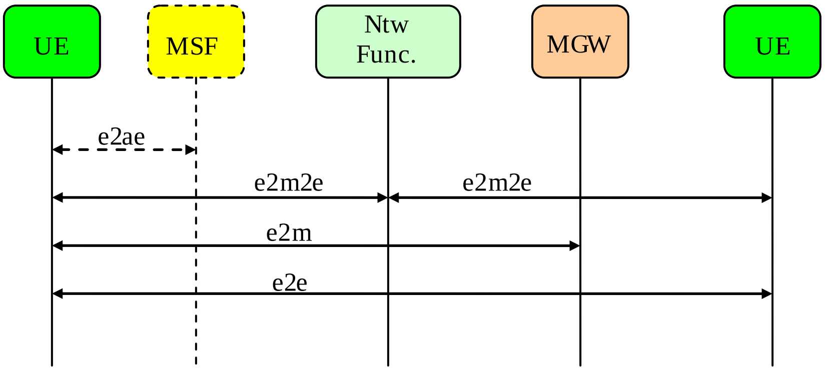

Figure 1: Illustration of the different types of media security endpoints

When the IMS network is trusted, transmitted media may be protected only between the UE and a Media Security Function (MSF) at the edge of the IMS. This case is shown as end-to-access-edge (e2ae). The case when one or more network nodes should be allowed to have access to the plaintext media is denoted end-to-middle-to-end (e2m2e). In cases when media should be forwarded over legacy systems we have an end-to-middle (e2m) use case in which the security is terminated in the network. (e2ae security is a special case of e2m security.) Finally, there is an end-to-end (e2e) security use case.

To handle these use cases the terminals and the network may communicate the security capabilities and the desired/accepted security functionality. Terminals need not necessarily be able to differentiate all the different cases shown in the figure above and the exact functionality is ffs.

4.1.2 Access media protection

The target for access media protection is to establish a security level for IMS media over access networks which would be comparable with the access protection in cellular systems. Such a solution is definitely 3GPP IMS internal and it has no interoperability requirements against other SIP based systems. It is an operator provided and controlled service.

Access media protection would have its main application in IMS systems where the access network does not offer security. It should have the same characteristics as access security in cellular networks, i.e. it should be automatic and in principle invisible to the user. The user experience would be as for an unprotected call.

In the access media protection usage model the user registers his access media protection enabled terminal with the IMS system. If the system supports access media security all media paths will be protected between the terminal and a node in the IMS access edge. This means that all services will operate as in an unprotected system and access media protection will not have impact on their workings and implementation. An indicator in the terminal may inform the user if no access security is provided.

4.1.3 End-to-end protection

For the general public, the peer-to-peer voice call will initially be the most significant use case. While these users do not have specific security policies, it can still be expected that they understand and value the feature that such a voice call can be encrypted in a way that “attackers in the Internet” have no chance to eavesdrop on the communication. (Users may have less concerns about the security of the operator-controlled part of the network, so, among the security requirements from clause 5.4, requirement 6 may be the most relevant for this use case.) Users will understand that it is not sufficient to secure only a part of the connection and that end-to-end protection is needed (potentially protecting all the hops separately) (→ requirements 4 and 5). Note that such a protection feature is already known to the public, e.g. by its usage in Skype.

Users may also understand that encrypted calls are not possible, if the called party does not support encryption. However, they will appreciate it if the protection feature is available not only for a small group of communication peers. This implies that interoperability with communication peers outside the IMS or peers using IMS terminal compliant to Releases prior to the introduction of IMS media security in 3GPP specifications would be beneficial (→ requirement 20).

On the other hand, it is not likely that many users are willing to be charged significantly for the encryption feature (which implies that the solution has to be cost efficient for the operator → requirements 19, 21, 27, 28, 31), and that they would accept degradation of the service performance caused by encryption (→ requirement 29). Also, most users most probably prefer "automatic protection" (→requirement 31 and 39).

Peer-to-peer voice call is as mentioned above initially expected to be the most significant use case. Over time, it is however expected that IMS will offer a rich set of services and that the users will use this services increasingly. For example, instant messaging is a popular service today, and it can be expected that also IMS users will be interested in such a service. It can be expected that users value protection of their instant messaging, as for their voice calls. As another example, users also frequently use voice mailboxes today. For this case, users may desire that the confidentiality of the data is upheld even while stored in the mailbox.

4.1.4 Enhanced end-to-end protection

Many user groups have well established security requirements for protection of their communication, e.g. enterprises, NSPS organizations, and government authorities. The trust model adopted in these cases is based on a need-to-know model. Keys should only be available where needed and only in authorized entities. The same is of course true also for plaintext media. An end-to-end protection should preferably also securely indicate the identities of the caller and the callee.

To serve the different user groups' different requirements, there has to be user control of the application of end-to-end media protection. Some organizations may prefer to have security initiated by specific user request to make sure that the user takes notice that security is turned on or not, some may want to apply security based on the callee identity, and others would like to configure it as an automatic service used without user intervention. The user must have access to full information about the security status of his call and warnings may be required if default security options are not complied with. However, a user should be able to configure the security set-up to give the same user experience as when making an unprotected call.

The user registers his end-to-end media capabilities and preferences with the IMS system. The system may support media security by providing media plane security termination at gateways for interoperability with legacy systems, e2m2e security for the purpose of transcoding, end-to-end protected store and forward of media, etc. All this has of course to be based on a user policy allowing it. A user may then either configure his device to always try to use media security, or he can indicate manually or in his address book when media security should be used.

4.2 Multimedia telephony

4.2.1 General

Clause 4.2 is about use cases where end-to-end media security is a requirement. If access security is the only requirement, there are no use case specific aspects on security.

NOTE: The use of the term "multimedia telephony" in this clause is not limited to the definition in TS 22.172.

4.2.2 Peer-to-peer

The most common use case for multimedia telephony is a call from one peer to another. Usually the call is made directly between the initiating terminal and a terminal used by the designated receiver. However, sometimes the receiver has set up call forwarding to some other user's terminal. In other cases the call may be directed to more than one terminal (forking). A typical scenario which combines both call forwarding and forking is when the phones of a manager and his assistant ring simultaneously and the call is forwarded to a voice mailbox if neither of them answers the call. At call initiation or latest when the connection has been established the originating and the terminating user's identities should be displayed at the other end. Only the party picking up the call should have access to the plaintext media.

The security settings for the call indicate if and how media security should be used. The user may indicate if end-to-end security is requested, or if end-to-middle-to-end security is acceptable. The latter allows trusted network nodes to access the clear text content to be able to perform e.g. transcoding. If a call is started in unprotected mode it is possible to initiate security during the ongoing call. The user may determine if use of security is independent of the terminating user/subscriber identity or if it only should be used if the call is terminated at a defined (set) of user/subscriber(s). Note further that there is a trade-off between these security configuration options and usability requirements.

4.2.3 Non RTP based media

Multimedia telephony includes non-RTP based media such as text communication, file transfers, video clip sharing, picture sharing, audio clip sharing, etc. Such media is normally MIME encoded and transported over MSRP [5][6] with media set-up in SDP. Information carried over MSRP may, according to the standards [5] [6], be protected by (PSK-)TLS [7] [8] on a point-to-point basis or by using S/MIME [9]. The standard solutions available thus either only give a point-to-point protection or rely on public-key cryptography.

4.2.4 Deferred delivery

One use case of particular interest is when a call ends up in a voice or other media mailbox in the network. In this case it may be beneficial if the media payload could be stored by the mailbox in the same encrypted format as it is sent in, i.e. without any decryption of the ciphering protecting it. When the end user later accesses the encrypted media in the mailbox it should be sent without having to perform re-encryption. Whether avoiding decryption and re-encryption at the mailbox for other than security reasons is ffs. In either case, channel security, specifically replay and integrity protection of the communication between the end-point and mailbox is necessary. PSS [4] is an example of how RTP payload media protection can be combined with transport security. Deferred delivery of end-to-end protected media would require an end-to-end security association for application layer security and security association per hop for channel security where the hop-by hop security associations might be derived from the end-to-end security association.

Deferred delivery of end-to-end protected media may require a key management system which does not depend on the identity of transmission end-points but should depend on the identities of the sender and intended receivers. This type of deferred delivery may require new media set-up signalling and new media protection mechanisms or a combination of existing ones. It will however not be a problem for the caller to, if needed, determine the type of the terminating device. As signalling of capabilities already is part of SIP.

4.2.5 Group and conference calls

Another use case is in group communication, e.g. conference calls with end-to-end security. In one realization of this type of service all users have access to the same key, the group key. In another realization separate group keys per sender are used. If support of large groups is out of scope, as it would be for normal size conference calls, group key management could be based on naïve schemes, e.g performing distribution of the group key directly from a key management server to each user in the group. If end-to-end security isn't required, the conference bridge may decrypt and then re-encrypt the media and other solutions will be available, e.g. protecting the communication between a user and the conference bridge using user unique keys. Note that a conference bridge needs access to media in cleartext for a type of group communication where participants expect the bridge to mix media streams, like in a normal conference call. Still group key management could yield simple and efficient solutions also for this case. Note that use of group keys is not the only solution for securing conferences.

According to differences during session setup, conference calls can be classified into two kinds:

1. Passive Join: The chairman of the conference call calls each participant one-by-one or all-at-once to hold the conference call. All participants except the chairman join the conference call passively.

2. Active Join: All eligible participants share the same secret, e.g. "IP address, user name, password, etc.", and join the conference call by themselves.

There is also a scenario where only one participant is allowed to speak in the conference call at a time. In this case the conference bridge is not a mixer but a middle box to co-ordinate the conference call, and end-to-end security can be provided.

4.2.6 Conclusions

The conclusions from the multimedia telephony use cases described above are that it would increase the applicability of the key management system if it, in addition to straightforward point-to-point channel protection also could support group keying, application layer security and deferred delivery of end-to-end protected media. The key management system should also be generic in the sense that it is easy to introduce keying for new services. Media can be RTP-media and/or different types of text, video, and picture streams/files/formats.

It should be noted, however, that existing security protocols (like TLS or SRTP) that may be used to protect multimedia telephony services have considerably different requirements with respect to security context management. It may therefore make sense to reduce complexity by offering optimized solutions for major use cases, e.g. a key management system focussed on RTP based traffic (voice or video calls as well as media streaming).

4.3 Push-to-talk (PoC)

Push-to-talk systems are in principle store and forward systems with message replication for all intended receivers taking place in the PoC server. PoC systems also often support instant messaging. Furthermore, it should be noted that PoC systems may offer automatic functions for recording of all messages a user cannot receive “on-line”. Thus, for true end-to-end security PoC systems exhibit the same requirements on key management and media protection as the multimedia telephony described above, i.e. a group key management system capable of handling deferred delivery of media. A PoC system doesn’t only handle voice but also handles other media types like e.g. video and text.

4.4 Instant messaging

Instant messaging systems have many similarities with PoC systems, the main difference is that they focus on non-speech media even though they may also carry voice and video messages.

For peer-to-peer instant messaging, there might be a direct link between the peers but in most cases, due to charging and delivery of different types of system services, the messages are forwarded via one or more intermediary nodes. For multiuser instant messaging, messages are routed to an instant messaging server where they are replicated and sent to all intended receivers. The messages might be carried in the signalling path in e.g. SIP MESSAGEs or they can be transferred e.g. on MSRP links. To protect messages carried in SIP MESSAGE, application layer security may be used. MSRP links can be protected hop-by hop with TLS or with S/MIME (see clause 4.2.3). Alternatively, SIP MESSAGE messages may rely on the protection mechanisms that are recommended for SIP traffic in general, e.g. TLS or Ipsec in the access, or Za/Zb interfaces in the core [19].

4.5 Chat

Chat differs to a certain extent compared to the use cases described above. Here chat messages usually end up in the chat server where they are handled in plaintext. It is difficult to imagine how an efficient chat service based on true end-to-end security could be developed. Thus here the security requirements are mainly to protect the communication between the user and the chat server. This communication may however be over multiple hops and require the same type of protection of media as used to protect IM to achieve terminal to chat server security.

4.6 Media on demand

Media on demand is a service delivered over IP-based networks managed to provide the required level of QoS/QoE. While there may be many security requirements on media on demand services such as IPTV, the focus of the work in this present document is to study whether an IMS media security solution can be used to protect the confidentiality and authentication of media on demand during transit. It is not the objective of this study to design a solution to protect broadcast services over IP networks, or to provide a content rights management solution.

4.7 Transcoders

Transcoders are devices in the network that need to change the media coding or make other necessary modifications of the media streams. For example RFC 4117 [4117] describes the usage of transcoders in the context of SIP showing examples when media streams are “transcoded” between audio and text as one of the communication endpoints could be deaf or hearing impaired.

As is described in clause 5.4.1 of TS 23.228 [3], the MGW may support transcoding between a codec used by the UE in the IM CN subsystem and the codec being used in the network of the other party. In general a MRFP may perform transcoding and/or other media stream processing.

In order to support this use case media protection needs to be terminated before or at the transcoder.

The current IMS architecture as described in TS 23.228 [3] has specified that transcoding function may be present in the TrGW, the MRFP and the CS/IMS-MGW. The local breakout and optimal media routing work in SA2 may define requirements to provide the transcoding service via these Functional Entities to media flows for roaming UE in the Visited packet switching network [18]

Hence it is necessary that the media security architecture shall not prevent a visited network from providing transcoding service on behalf of flows for roaming UE.

4.8 PSTN-GW

PSTN gateway provides interworking between IMS networks and circuit switched PSTN.

According to clause 5.4.1 of TS 23.228 [3] the IM CN subsystem is also able to interwork with the CS networks (e.g. PSTN, ISDN, CS domain of some PLMN) by supporting, for example, AMR to G.711 transcoding in the IMS MGW element. Furthermore to allow interworking between users of the IM CN subsystem and IP multimedia fixed terminals and other codecs may (this is implementation dependent) be supported by the MGW. I.e. MGW is expected to act as a PSTN-GW.

In order to support this use case media protection needs to be terminated before or at the PSTN-GW.

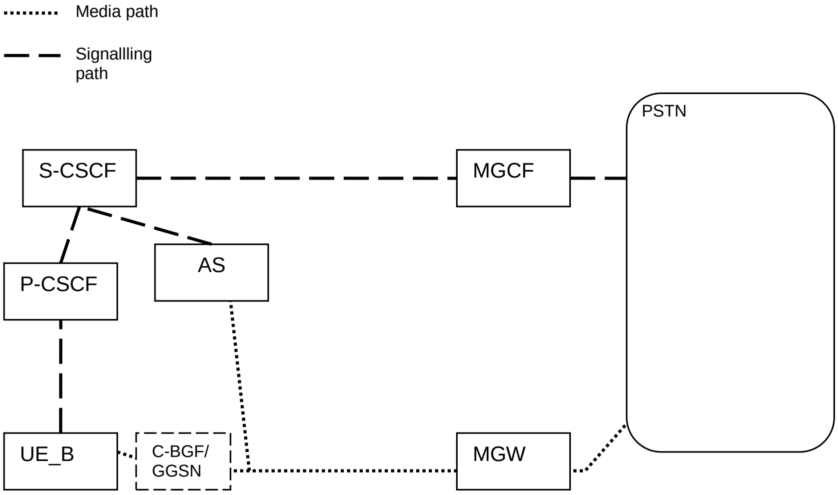

Figure 2: A simplified view of PSTN – IMS interworking

4.9 Termination of media security in an AS

An IMS session is not always setup between two UEs. It may also be terminated in an Application Server (AS).

In order to support this use case media protection needs to be terminated before or at the AS.

5 Requirements

5.1 Overview

The purpose of this clause is to identify 3GPP requirements for IMS media plane security. The requirements are grouped into various categories in order to ease discussion and to check for completeness.

5.2 Summary of requirements

A solution/framework shall preferably provide a level of security that can satisfy the needs of different user groups, including private users, enterprise and NSPS (National Security and Public Safety) related organizations as far as possible. It shall cover well the most frequent use cases. It shall be cost efficient, scale well for a large number of subscribers, shall not adversely affect the performance of IMS services and shall have minimal impact on existing networks. It shall allow interworking with non IMS-capable user equipment. It shall satisfy applicable lawful interception requirements. In case it turns out that there is no single solution satisfying all these requirements, or that such a solution may lead to undue complexity or delay in standardisation and/or deployment, it may be acceptable to standardise more than one solution.

5.3 Lawful interception

1. Lawful interception requirements shall be met.

2. The lawful interception solution shall not require the operator to reveal information to the interception agent that would allow him to intercept user communications that are outside the terms of the intercept warrant.

3. It shall not be possible for users to determine whether their communications are subject to lawful interception.

NOTE: Further study is needed on the exact requirements for lawful interception.

5.4 Security

4. It shall be possible to protect IMS user traffic against eavesdropping, modification, spoofing, and replay on access network interfaces and access network nodes.

5. It should be possible to protect IMS user traffic against eavesdropping, modification, spoofing, and replay on core network interfaces and at core network nodes. Depending on the use case, the degree of protection against these threats provided for IMS user traffic shall be equal to or higher than that provided for IMS signalling traffic.

NOTE 1: It should be considered whether SA3 could relax this requirement so that the decryption key could be revealed to IMS network elements and on some core network interfaces.

6. The level of security provided should satisfy operators and the most important user categories, whilst at the same time satisfying applicable lawful interception requirements. If this level of security is insufficient for high security user groups, an enhanced solution may be additionally provided.

7. A key management solution shall prevent a party engaging in a key exchange with a spoofed user identity (i.e. IMPI/IMPU) without being detected.

8. A 3GPP solution shall provide protection against active attacks on access network interfaces and access network nodes. It should also be possible to protect against active attacks on core network interfaces and at core network nodes. Depending on the use case, the degree of protection against these threats provided for IMS user traffic shall be equal to or higher than that provided for IMS signalling traffic.

NOTE 2: Active attacks at core network nodes may be mitigated by measures, such as e.g. hardening, local access control, provided independently of a media plane security solution. This would allow simple key management solutions to be adopted where the sender generates the end-to-end key and sends it to the receiver in SDP according to e.g. RFC4568.

9. Perfect Forward Secrecy is not considered to be required in a 3GPP network.

5.5 Requirements related to SIP based call features/SIP related problems

5.5.1 Early media/media clipping

10. In a 3GPP architecture media clipping shall be avoided, even at the cost of additional UE signalling.

5.5.2 Secure multiparty communications

11. A key management solution shall support secure multiparty communications (i.e. key management to distribute a group key) where the server relaying multiparty communication (e.g. a conference bridge) does not know the group key.

12. A key management solution shall support secure multiparty communications (i.e. key management to distribute a group key) where the server relaying multiparty communication (e.g. a conference bridge) knows the group key.

NOTE 1: This kind of group key could be used for example for conference call, PoC, etc.

NOTE 2: Shared key conferencing is out of scope of the IETF media security work.

5.6 Architectural

13. Encryption and integrity protection of user media should be applied on an end-to-end basis, where possible, to save on network resources and to avoid restrictions on media plane routing.

14. Where it is not possible to provide protection on an end-to-end basis due to cost or complexity reasons, then solutions should be developed which terminate user plane security in an appropriate network element (e.g. at a conference bridge, a transcoder, an application server or at interworking gateways with non-IMS networks).

15. Since the network based termination is possible in the visited and home packet switching networks and at terminating or originating side, the security architecture should allow a transcoding function residing in the visiting packet switching network to provide transcoding service to media flows for roaming UE, even in cases where the P-CSCF and the transcoding function resides in different operator domains.

16. It should be possible for operators to be able to terminate media plane security in the network in some cases, e.g. if the operator needs access to the media for content control purposes.

17. A solution SHOULD support media recording (ffs).

18. Multiple solutions should be avoided to reduce complexity in the network and to maximise interoperability between user devices. However, in case it turns out that there is no single solution satisfying all these requirements, or that such a solution may lead to undue complexity or delay in standardisation and/or deployment, it may be acceptable to standardise more than one solution. If multiple solutions are standardised, then they shall be defined within a single framework.

NOTE 1: It is ffs whether re-use of IETF developed protocols such as MIKEY, SDES and DTLS-SRTP can be used in 3GPP to satisfy this requirement.

19. The requirement for new functions on the user's smartcard should be avoided unless it would provide significant and cost effective benefits.

20. The solution should support the possibility to protect user traffic on an end-to-end basis between IMS-capable and user equipment which is non IMS-capable or conforming to a 3GPP Release prior to the introduction of IMS media security.

21. The solution shall have minimal impacts on already deployed network entities.

22. A media security solution shall assume that messages cannot be sent over the media path until the media session has been established.

NOTE 2: 3GPP and TISPAN networks will likely block all traffic on media path until the media session has been established (i.e. until the initiator has received the responder's answer in the 200 OK message).

23. A media security solution shall assume that only media traffic can be sent over the media path.

NOTE 3: Media path nodes in 3GPP and TISPAN networks will likely not let anything other than media traffic through, e.g. due to traffic policing.

24. Media security solutions for media protection and key management shall cover both end-to-end and end-to-middle media protection scenarios.

NOTE 4: Whether the solutions (especially for key management) are the same or different for end-to-end and end-to-middle scenarios may depend on environment, cost and complexity reasons.

25. While 3rd-party certificates are not acceptable for a solution for the majority of users, the use of certificates, e.g. from an enterprise PKI, may be acceptable for special user groups.

26. In the 3GPP architecture the preferred solution is to perform the key exchange messages in the signaling path only.

5.7 Scalability, cost and performance

27. The solution should scale well for large numbers of users.

28. The solution should be cost effective.

29. The solution should not adversely affect performance of IMS services. In particular, there should be no significant increase in call set-up delay and no media clipping.

5.8 Requirements regarding the access network type

30. The solution shall support the possibility to provide protection on an end-to-end basis between any IMS-capable UE regardless of what type of access technology they use (fixed DSL, WLAN, cellular, etc.)

31. The key management solution should be based on the existing IMS access security architecture, so that no special user registration or user involvement is required, and so that existing infrastructure can be re-used.

32. Since the IMS client may use different access authentication methods, both smartcard and non smartcard based, the key management solution for end-to-end security shall be able to work independently of any of these authentication methods.

5.9 Backward compatibility and migration

33. Media security shall be mandatory to implement for UEs and networks and optional to use for UEUEs.

34. The media security solution shall allow a UE to negotiate media security settings for each individual call.

35. The negotiation of media security must be protected against downgrading attacks

I36. Void

5.10 Other requirements

37. A solution shall support the possibility to protect RTP-based IMS user plane traffic.

38. A solution shall support the possibility to protect non RTP-based IMS user plane traffic as well as application layer messages, e.g. SIP MESSAGE. In case it turns out that a single solution may lead to undue complexity or delay in standardisation and/or deployment, it may be acceptable to standardise more than one solution. If multiple solutions are standardised, then they shall be defined within a single framework.

NOTE 1: An example use case for this requirement is Message Session Replay Protocol (MSRP) RFC 4975.

NOTE 2: Even though in the example of SIP MESSAGE a signalling message is used for transport, it can still be regarded as being part of the media plane since it carries user content and may need similar protection, e.g. confidentiality, as RTP and MSRP.

39. The media security solution should not require user intervention. It may, however, allow a certain degree of configurability and may support the indication of the security level of a session.

NOTE 3: Some key management solutions require user intervention in the sense of reading aloud an authentication string to the other endpoint. This may be an inconvenient user experience, especially for elderly or disabled persons.

40. A party shall have the possibility to get assurance about the identity of any other party in the session when the party joins a point-to-point session.

NOTE 4: In particular, is necessary to give the calling party assurance about the identity of the responding party (after forking, etc.). It is explained in clause 7.3.2.2 and 7.1.4.4.5 how IMS mechanisms can be used to satisfy this requirement in certain scenarios. The details of the requirement are ffs. The corresponding requirements in the case of a point-to-multipoint session are ffs.

41. A calling party shall have the possibility to stay anonymous towards any called parties in the session.

42. The user should be able to access information about the scope of protection (end to access edge, end-to-middle-to-end or end-to-end), applied security level (if needed). It should also be visible if any non-IMS operators are involved in the session set-up. This should be balanced against the usability of such a feature and complexity of realisation.

43. It should be possible to configure the terminal to give a visible or audible warning when security is not according to a policy defined by the user.

44. A key management solution shall support deferred delivery of media. In case it turns out that a single solution also supporting deferred delivery may lead to undue complexity or delay in standardisation and/or deployment, it may be acceptable to standardise more than one solution. If multiple solutions are standardised, then they shall be defined within a single framework.

6 General aspects of solutions

6.1 Introduction

This report discusses both end-to-middle (e2m) and end-to-end (e2e) media security.

In e2m media security, media protection is applied between an IMS UE and an IMS core network node in the media path without being terminated by any intermediary. Application of e2m security is mainly a network responsibility and may be applied independently on the originating and terminating sides.

In e2e media security, media protection is applied between two IMS UEs without being terminated by any intermediary. Application of e2e media security should be controlled by the user.

Detailed descriptions of specific solution proposals are found in clause 7.

6.2 Architectural aspects of end-to-middle protection

6.2.1 Preferred endpoints for end-to-middle protection

A schematic diagram of an IMS system with controlling and media handling entities is depicted in Figure 3.

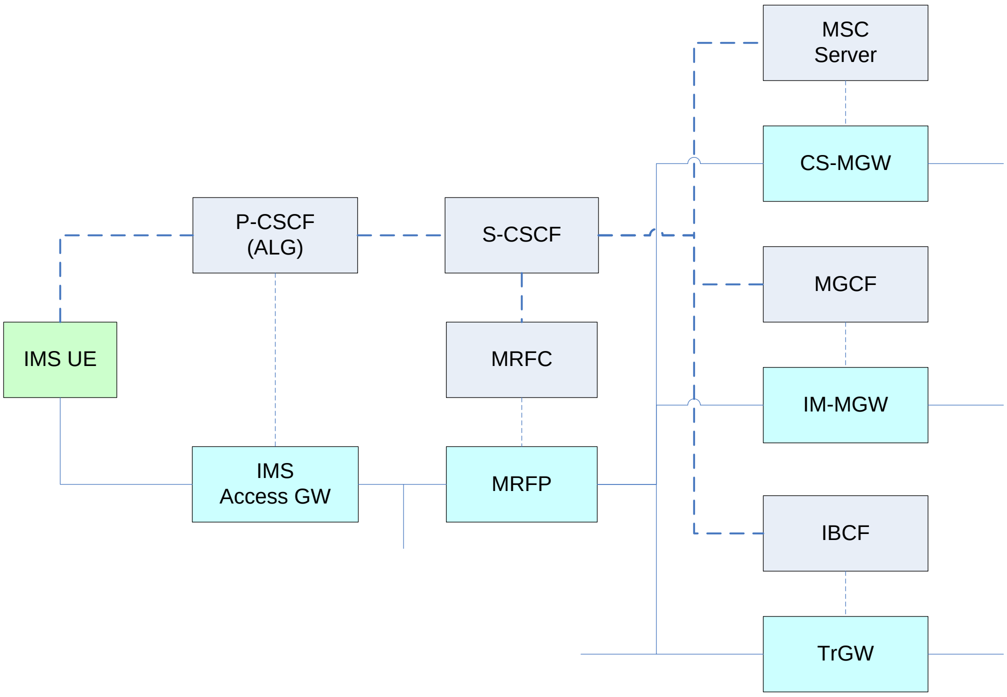

Figure 3: IMS signalling and media plane entities

The media plane traffic may be routed in different ways depending on required network supported functionality and in which type of system the terminating device is present. Traffic between two IMS UEs in the same IMS domain could be routed without involvement of any media node. The media can however be anchored in the access edge via the IMS Access GW (e.g., for NAT traversal or transcoding purposes), while at the same time ensuring that media traffic is only routed locally (which may be a requirement for local breakout). Traffic may also be routed via the home network to a MRFP for conferencing, transcoding or other media handling functions. When the traffic is intended for a subscriber in another domain/network, the media traffic is routed to an interworking gateway (CS-MGW, IM-MGW or TrGW).

From a security point of view it would be preferable to terminate the media plane security as far into the network as possible. This would mean that when traffic is between an IMS domain and e.g. a legacy system, media plane security should be terminated in the IM-MGW and when traffic is between IMS UEs in the same domain the media plane security should be terminated in the IMS Access GW. Such dynamic behaviour may, however, be problematic and may incur a lot of complexities and added new functionality.

The discussion below takes a call initiated by the IMS UE in Figure 3 as a starting point. Similar considerations hold when analysing handling of media coming into the domain via an interworking gateway.

Looking at how a call is set-up, we first note that resource allocation for media handling in SIP-proxies is performed when an INVITE is processed. This means that when e.g. the P-CSCF/ALG, or another SIP proxy in the IMS signalling path, handles an INVITE it has to decide if it should initiate termination of the media plane security or if it should let the secured media pass further into the network. This means that this SIP proxy will have to apply a policy on whether to terminate media plane security or not do this.

If e2m protection is to be potentially performed in an IMS network then, in a particular IMS network deployment, there shall be one dedicated SIP proxy in the IMS signalling path, which is capable of handling e2m protection. It can be determined by network configuration which SIP proxy shall take this role, e.g. the P-CSCF, the S-CSCF or an MRFC. In cases where an IMS Access GW is part of the media path anyhow, it would seem natural to assign this role to the P-CSCF, which controls the IMS Access GW.

6.2.2 Interfaces for end-to-middle protection

Editor's Note: Describes the interfaces needed to pass media encryption keys from the control plane entity to the media plane entity.

6.3 Co-existence of end-to-end and end-to-middle solutions

6.3.1 Introduction

This report discusses 3GPP standardization of e2ae and e2e media security.

Application of e2ae media security is mainly a network responsibility and if the network considers it to be appropriate the network should offer all e2ae capable IMS UEs this protection. Application of e2e media security should be controlled by the user. If a user accepts network access to media for network support functions like transcoding, this is not part of the capability registration but should be handled in the actual media security set-up procedure.

To get simple operational procedures and good usability, it is beneficial to register the IMS UE's media security capabilities when the IMS UE registers. The network can then base its decision on knowledge of the IMS UE's capabilities. This is especially important in the terminating procedures, when the network e.g should select the terminating device from a set of registered devices belonging to the called user.

The set-up of media security is controlled in the signalling plane. Normally, the initiator includes an offer for media security in the originating session set-up. The parameters of such an offer should indicate the preferred media security capability to use.

For e2ae media protection it would be beneficial if the initiator could express a preference that the terminating side also should apply e2ae protection. A notification back to the originator that this has occurred would make this feature even more useful.

The following scenarios described by signalling flows have been written to be independent of the particular security mechanism finally selected for key management and media protection. Where applicable, they generalize the flows in clause 7.2.3.

6.3.2 Registration procedures

Figure 4 shows an IMS registration procedure in which the IMS UE registers its media security capability.

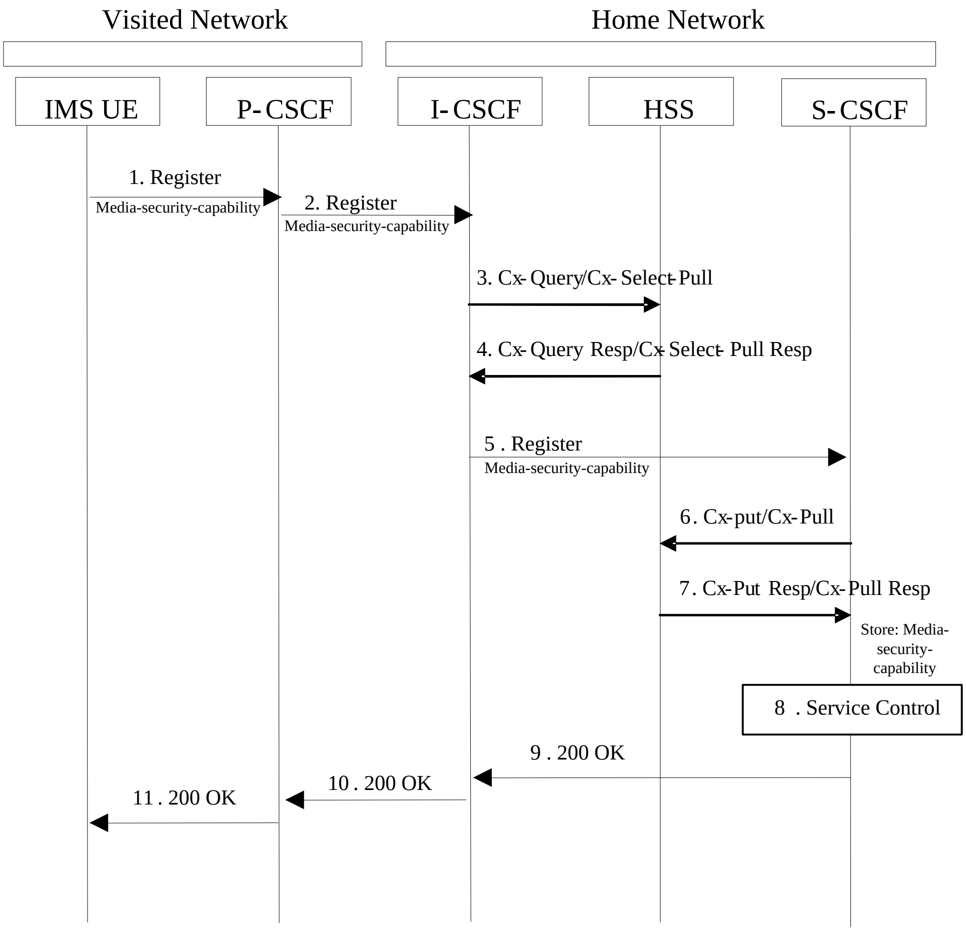
Visited Network
Home Network
IMS UE
P-CSCF
I-CSCF
HSS
S-CSCF
1. Register
Media-security-capability
2. Register
Media-security-capability
3. Cx-Query/Cx-Select-Pull
4. Cx-Query Resp/Cx-Select-Pull Resp
5 . Register
Media-security-capability
6. Cx-put/Cx-Pull
7. Cx-Put Resp/Cx-Pull Resp
Store: Media-security-capability
8 . Service Control
9 . 200 OK
10 . 200 OK
11 . 200 OK

Figure 4: Registering Media security capabilities

The IMS UE performs an IMS registration according to 3GPP TS 23.228. When performing the registration, the IMS UE (in Step 1) includes the supported media security capabilities (end-2-access-edge or end-to-end). This can be added to the registration message as any other capability.

When receiving the media security capability from the I-CSCF (Step 5), the S-CSCF stores the capability of the IMS UE.

6.3.3 Originating procedures

6.3.3.1 End-to-access-edge

Figure 5 shows the originating procedures for session establishment using e2ae security. In this scenario, the P-CSCF / IMS ALG is used to terminate the media security negotiation, and the IMS Access Gateway hosting the MSF is used to terminate the media security from the IMS UE.

NOTE: The procedures shown in the figure apply to users located in their home service area. The same concepts apply to roaming users.

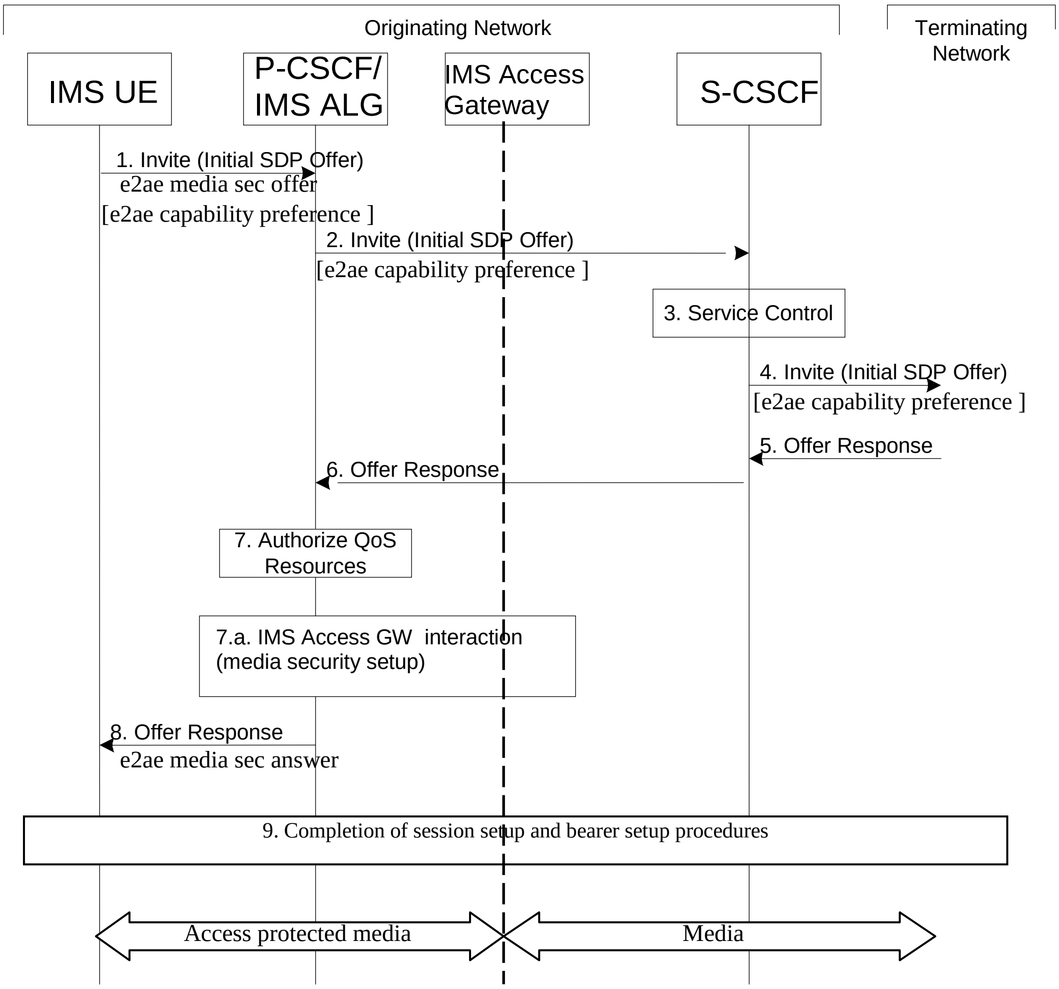
Originating Network
Terminating Network
IMS UE
P-CSCF/ IMS ALG
IMS Access Gateway
S-CSCF
1. Invite (Initial SDP Offer)
e2ae media sec offer
[e2ae capability preference]
2. Invite (Initial SDP Offer)
[e2ae capability preference]
3. Service Control
4. Invite (Initial SDP Offer)
[e2ae capability preference]
5. Offer Response
6. Offer Response
7. Authorize QoS Resources
7.a. IMS Access GW interaction (media security setup)
8. Offer Response
e2ae media sec answer
9. Completion of session setup and bearer setup procedures
Access protected media
Media

Figure 5: Originating call flow for end-to-access-edge case

The IMS UE performs an IMS originating session setup according to 3GPP TS 23.228. When performing the session setup, the IMS UE (in Step 1) includes the e2ae offer, and optionally an indication that it desires this capability to be used for the terminating side as well.

The P-CSCF / IMS ALG is the termination point of the media security negotiation.

When the offer response is received, the P-CSCF / IMS ALG may interact with the IMS Access Gateway to setup the media security. The P-CSCF / IMS ALG includes a media security answer in the offer response sent to the IMS UE. Already at this point in time, both the IMS UE and network will have sufficient credentials for the media security.

When the full session setup has completed, and media can be sent, the protected media is sent between the IMS UE and IMS Access Gateway.

6.3.3.2 End-to-end

Figure 6 shows the originating procedures for session establishment using e2e security.

NOTE: The procedures shown in the figure apply to users located in their home service area. The same concepts apply to roaming users.

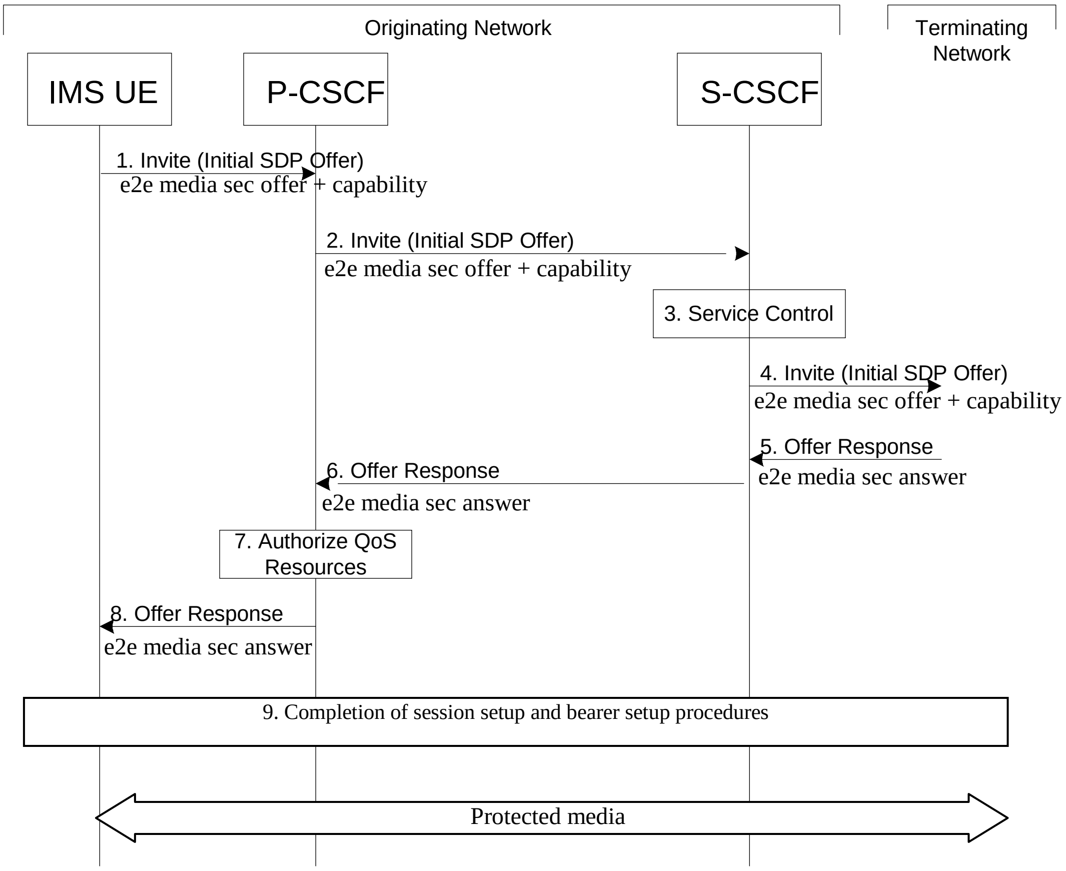
Originating Network
Terminating Network
IMS UE
P-CSCF
S-CSCF
1. Invite (Initial SDP Offer)
e2e media sec offer + capability
2. Invite (Initial SDP Offer)
e2e media sec offer + capability
3. Service Control
4. Invite (Initial SDP Offer)
e2e media sec offer + capability
5. Offer Response
e2e media sec answer
6. Offer Response
e2e media sec answer
7. Authorize QoS Resources
8. Offer Response
e2e media sec answer
9. Completion of session setup and bearer setup procedures
Protected media

Figure 6: Originating call flow for end-to-end case

The IMS UE performs an IMS originating session setup according to 3GPP TS 23.228. When performing the session setup, the IMS UE (in Step 1) includes the e2e media security offer, and an indication that the e2e media security capability is used.

When receiving the offer response, an e2e media security answer is included in the case the terminating end point accepted the offer. Already at this point in time, both the end points will have sufficient credentials for the media security.

When the full session setup has completed, and media can be sent, the protected media is sent on an end-to-end basis.

6.3.4 Terminating Procedures

6.3.4.1 End-to-access-edge

Figure 7 shows the terminating procedures for session establishment using e2ae security. In this scenario, the P-CSCF / IMS ALG is used to originate the media security negotiation towards the IMS UE, and the IMS Access Gateway is used to terminate the media security from the IMS UE.

NOTE: The procedures shown in the figure apply to users located in their home service area. The same concepts apply to roaming users.

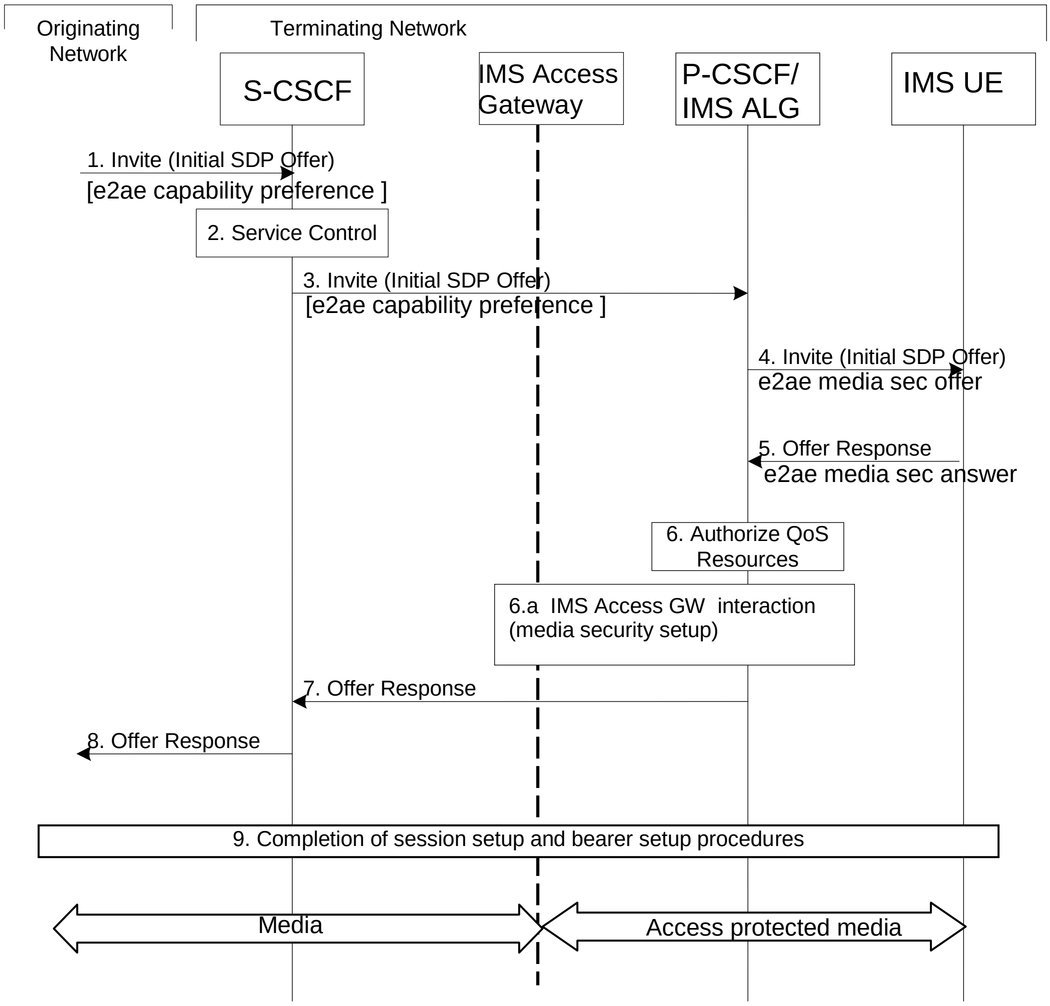
Originating Network
Terminating Network
S-CSCF
IMS Access Gateway
P-CSCF/ IMS ALG
IMS UE
1. Invite (Initial SDP Offer)
[e2ae capability preference]
2. Service Control
3. Invite (Initial SDP Offer)
[e2ae capability preference]
4. Invite (Initial SDP Offer)
e2ae media sec offer
5. Offer Response
e2ae media sec answer
6. Authorize QoS Resources
6.a IMS Access GW interaction (media security setup)
7. Offer Response
8. Offer Response
9. Completion of session setup and bearer setup procedures
Media
Access protected media

Figure 7: Terminating call flow for end-to-access-edge case

A terminating session setup is received at the S-CSCF. In the event that a media security preference is included, the S-CSCF may base the forking based on this capability. The request is forwarded to the P-CSCF / IMS ALG of the selected IMS UE(s).

If e2ae is the default in the terminating network, or if the originator expressed a preference for e2ae and the terminating IMS UE has e2ae capability registered, the P-CSCF / IMS ALG includes an e2ae media security offer in the terminating request (Step 4). The IMS UE includes a media security answer in the offer response sent back.

The P-CSCF / IMS ALG hosting the MSF is the termination point of the media security negotiation.

When the offer response is received, the P-CSCF / IMS ALG may interact with the IMS Access Gateway to setup the media security. Already at this point in time, both the IMS UE and network will have sufficient credentials for the media security.

When the full session setup has completed, and media can be sent, the protected media is sent between the IMS UE and IMS Access Gateway.

6.3.4.2 End-to-end

Figure 8 shows the terminating procedures for session establishment using e2e security.

NOTE: The procedures shown in the figure apply to users located in their home service area. The same concepts apply to roaming users.

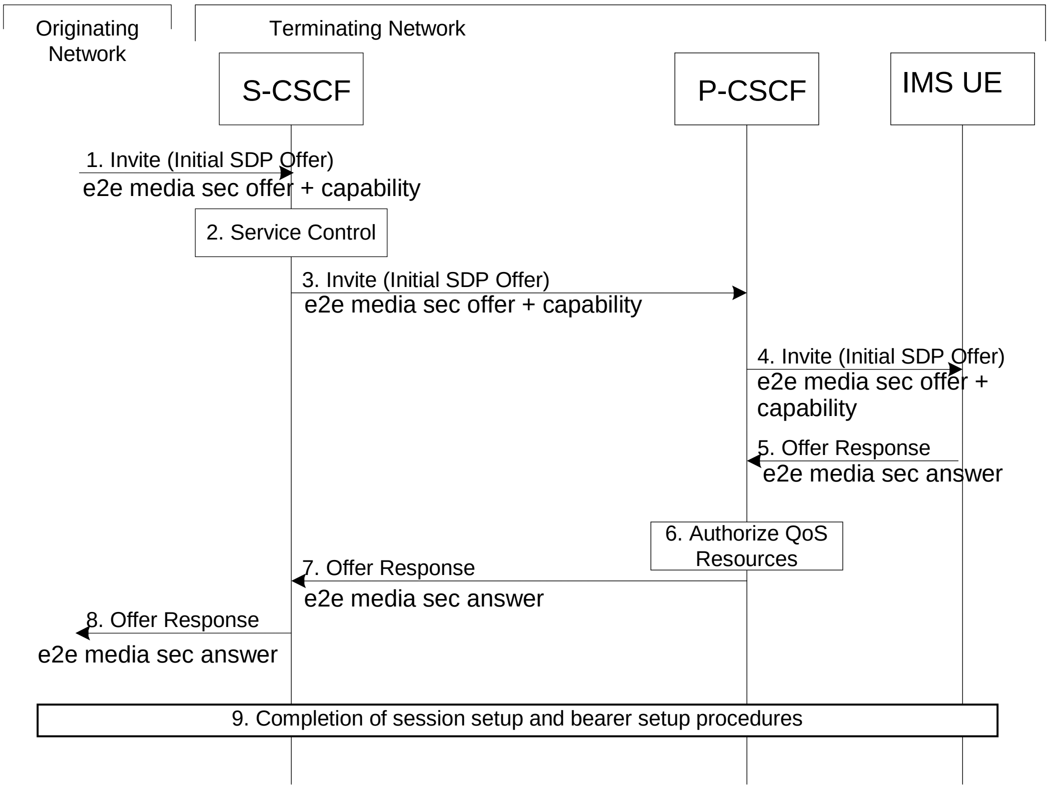
Originating Network
Terminating Network
S-CSCF
P-CSCF
IMS UE
1. Invite (Initial SDP Offer)
e2e media sec offer + capability
2. Service Control
3. Invite (Initial SDP Offer)
e2e media sec offer + capability
4. Invite (Initial SDP Offer)
e2e media sec offer + capability
5. Offer Response
e2e media sec answer
6. Authorize QoS Resources
7. Offer Response
e2e media sec answer
8. Offer Response
e2e media sec answer
9. Completion of session setup and bearer setup procedures

Figure 8: Terminating call flow for end-to-end case

A terminating session setup is received at the S-CSCF. The e2e media security capability is included in the request, which the S-CSCF may use to base the forking decision on. The request is routed to the selected IMS UE(s).

The IMS UE accepts the e2e offer, and includes a media security answer in the Offer response. Already at this point in time, both the end points will have sufficient credentials for the media security.

When the full session setup has completed, and media can be sent, the protected media is sent on an end-to-end basis.

7 Candidate solutions

7.1 Ticket-Based System (TBS)

7.1.1 Introduction

This clause describes a framework solution for IMS media security key management in which requirements from different user groups can be accommodated. A "ticket" concept, similar to Kerberos, is used to identify and deliver keys. The solution is described with MIKEY [19] for key delivery, and as such it is based on protocols already standardized with the IETF. Other key delivery schemes could be used, however. In TBS there are two main categories of tickets: protected and unprotected. Use of unprotected tickets gives security features similar to those offered by the SDES solution described in clause 7.3; their use is based on trust in the security of the complete IMS infrastructure. Protected tickets may be used to achieve higher security and provide security independent of the security of the complete IMS infrastructure; in this case a Kerberos-like Key Management Server (KMS) is the trust anchor. The KMS may also provide copies of keys to authorized network functions and middle-boxes. Use of a ticket based system may also help in the handling of keys for deferred delivery of end-to-end protected media to currently off-line users.

Key management based on a key management service requires a signalling mechanism between parties which allows them to retrieve the common credentials used for the media protection from the key management service. A convenient way to implement such a signalling scheme is to use a ticket based system. The sender requests a ticket from the key management service and sends the ticket containing a reference to the key, or the enveloped key, to the receiver. The receiver then sends the ticket to the key management service which returns the appropriate key.

This solution has a number of advantages. It offers a framework which is flexible enough to satisfy users with a broad range of security needs. Operators can set policies to provide end-to-end security and enable end-to-access-edge and end-to-middle scenarios. By using different MIKEY modes, a KMS can provide protected or unprotected tickets to support different levels of trust in the IMS infrastructure and other system components. Here, and in the following, the term IMS infrastructure denotes the current IMS infrastructure not including a KMS. Examples of user groups and their security expectations include:

- General public user: Trusts IMS infrastructure and is satisfied with unprotected tickets.
- Enterprise users: These users may require protected tickets, depending on their trust assumptions.
- NSPS user: Likely to require additional protection beyond what is provided by IMS, including protected tickets which are bound to the recipient's identity.

Normally key management systems are either based on negotiation between peers (e.g. Diffie-Hellman based schemes), pre-distributed knowledge of user credentials (shared secrets/certificates), or performed with the help of a key management service. In security systems serving large user groups a key management service is often preferred so that there is no need to distribute credentials in advance but to let the user request keys for any other user at time of need.

A KMS also provides a central point to enable LI and other requirements for network access to plaintext media.

In the following, the description of the solution has it main focus on use of protected tickets.

7.1.2 Analysis

According to clause 4.2.4, "it may be beneficial" to store the media without any decryption of the ciphering protecting it. This is manifested in requirement 44 in clause 5.10 which states that the key management solution shall support deferred delivery of protected media. This requirement is the requirement that has the greatest impact on the possible types of key management scheme and it restricts the types of key management that can be used. In particular, it excludes all key management schemes that are based on some type of negotiation between the participating IMS UEs / IMS users and implies that the sender/initiator must have access to media keys before the receiver has been contacted. A consequence is also that the receiver cannot rely on contacting the sender to get access to the keys used. Also note that the requirement on end-to-end protection at deferred delivery is more of a requirement on the media protection protocol(s) used, since deferred delivery of end-to-end protected media would in principle only require that the key management system can establish both an end-to-end security association for application layer security and security associations for channel security.

The best way to design the key management signalling is to have the key information associated with the media, forwarded with the signalling associated with the media set-up in e.g. a ticket. The ticket could be a reference to a key held by the key management system or it could hold the key itself. In the latter case, the ticket of course needs to be confidentiality protected. To have the key itself transported in a ticket is seen as the preferred solution as this would relieve the key management system of the task to keep a record of all keys used for media protection.

There are different alternatives for how the receiver gets access to the key in the ticket. The first is that the confidentiality protection of the ticket is based on a long-term key shared between the receiver and the key management system. With this option, it may be difficult to meet the requirements in clause 5.5.3 on secure multiparty communication. A second alternative is to have the ticket protected by a key known only by the key management system, which seems more favourable. However, this seems to imply that the receiver has to contact the key management system whenever secure media is received, but the KMS could issue base tickets with a certain lifetime from which per call keys are derived by the users. The key management system could implement some authorization functionality for group key management. The details of group key management are ffs.

End-to-end security can be enforced by the key management system by only distributing the media keys to authorised end-users. Note here that it is important to distinguish the end-user from the end-user equipment, and that an authorization function in the key management system should be based on end-user identity (IMPU/IMPI) as in requirement 7 in clause 5.4. This authorization function in the KMS could also be used to help solve the key access problem in forking and retargeting scenarios (as in clause 5.5.1). A discussion on how authorized end-users can be defined can be found in clause 7.1.4.2 and forking is described in clause 7.1.4.4. Some network nodes may require plaintext access to all media to perform various network functions, e.g. transcoding. These nodes should have special authorization to retrieve keys for all users.

The tickets should by preference be generic and their transport should not rely on the type of media they help protect. Thus a signalling plane solution for ticket transport seems to yield the simplest and most general systems solution.

The detailed design of ticket format and the specification of the interface between the IMS UE and the KMS, between different KMSs, and between authorized network nodes and the KMS, are ffs; still, possible approaches are described and discussed in the following. It should be noted that there are different options in the ticket design and that depending on selected features they may influence the statefulness of the KMS.

7.1.3 Solution description

A precondition for a key management scheme as discussed above is that the users can establish secure connections with the key management server and that mutual authentication is provided. In an IMS environment it is natural to base the establishment of such a trusted and protected connection between the user and the KMS on GBA. In Figure 9, a conceptual architecture for the discussed key management system is depicted.

Note that if GBA is unavailable, other types of credentials like username/password, client certificates, onetime passwords and server certificates can be used for establishing mutual authentication between the user and the KMS. Such credentials may, but doesn't have to, be related to the user's credential used for IMS access.

NOTE: 3GPP should only specify very limited number of solutions.

Also note that the KMS does not have to be operated by the IMS operator. It could be run by an enterprise or organization, which wants to have control of the key management for its media security. This is possible as the design of the KMS user SA establishment can, as described above, be based on any type of credentials that the KMS operator find secure enough. Note, however, that this may pose additional difficulties for Lawful Interception in case the enterprise KMS is located in a foreign country.

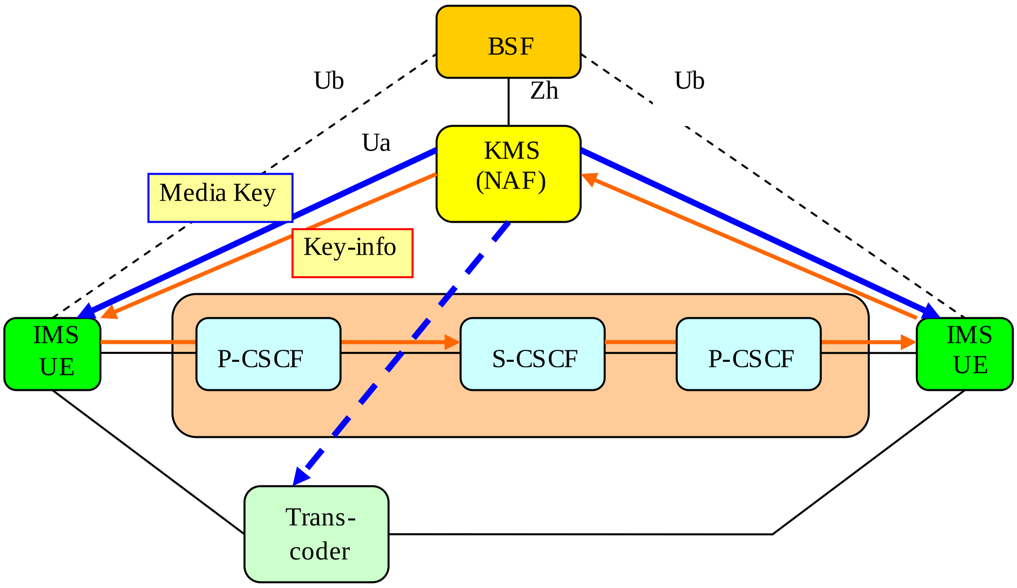

Figure 9: Architecture for key management system

Note that rather than a single KMS, two different KMSs may be involved, one for user A and one for user B. This is discussed in 7.1.4.3 below. Also note that rather than a single S-CSCF, two different S-CSCFs may be involved, one for user A and one for user B.

The key management when user A wants to establish a secure media session with user B follows the following steps:

1. IMS UE belonging to user A bootstraps with the BSF to be able to establish a secure connection with the KMS which acts as a NAF. This allows the BSF to authenticate the user and the user to indirectly authenticate the KMS.

If GBA cannot be used, the IMS UE connects and authenticates to the KMS and establishes a shared key, based on a pre-established security association. The exact procedure for this pre-establishment is ffs.

2. The IMS UE engages in a MIKEY exchange with the KMS and requests a key and a ticket to include in an INVITE to user B. This exchange would likely use the yet-to-be-defined PSK-R mode of MIKEY to allow the KMS to generate the media master protection key. The ticket is confidentiality and integrity protected and includes the media master key and other information needed like receiver's identity. In most cases the user identity should be an IMPU but for group key management a group identity or a list of users could be included.

NOTE: This solution requires extensions to MIKEY in the form of an IETF RFC. Such an Internet Draft is currently progressed in IETF.

3. The KMS generates the media master key and the ticket and sends them to the IMS UE of user A.

4. The IMS UE of user A includes the ticket in the INVITE and sends it to the IMS UE of user B.

5. The IMS core detects the INVITE and handles the ticket in such a way that a network function, if authorized, can get access to the master media key. To get the key the network function sends the ticket to the KMS with a request to receive the plaintext key.

6. The IMS UE of user B receives the INVITE including the ticket.

7. The IMS UE of user B connects to the KMS using GBA based MIKEY. The KMS gets an authenticated user identity this way.

   The comment in step 1 applies here as well.

8. The IMS UE of user B sends the ticket to the KMS and requests the master media key contained in the ticket.

9. The KMS retrieves the master media key and other information from the ticket and checks that user B is an authorized receiver of the master media key.

10. The KMS sends the master media key and the other needed information to the IMS UE of user B.

11. The IMS UE of user B accepts the invitation and use of media security.

7.1.4 System details

7.1.4.1 Ticket information and format

Tickets may carry many different types of information helping to enforce usage policies. Policies may be for all users or they may be per user. In the latter case the KMS has to maintain this information or be able to retrieve it from some other system. An example of basic information forwarded in a ticket is given in Table 1 below.

If tickets are unprotected any entity can generate a valid ticket and thus the only protection that the ticket needs is against random errors during transport. Protected tickets, as the name indicates, need better protection, and only a KMS should be able to generate them. The ticket information must thus be integrity protected and certain fields need confidentiality protection, in particular the media master key. Other types of information may also require confidentiality protection due to privacy reasons.

A KMS bases its ticket protection on an Issuer key. From this Issuer key necessary keys for integrity and confidentiality protection of tickets are derived.

Table 1: Ticket information content

Information field	Plain/ Encr.	Comments
Issuer	P	Identity of the KMS issuing the ticket. The identity should allow entities find the issuers contact details and it would be natural to require it to be a FQDN. Example: (kms.operator1.com)
Issuer key identifier	P	An identifier which makes it possible for the issuing KMS to change base key used in ticket protection. Example: 24 bit sequence number
Originator	P	Public identity of the user requesting the ticket from the KMS Example: user1@operator1.com
Recipient(s)	P	Public identity(ies) of intended recipient. Example: user2@operator2.com
Time of issue	P	Time when the ticket was generated by the KMS Example: 2008-12-11:13:01:23
Time of expiry	P	Time when the ticket expires. The format could be network time format Example: 2008-12-11:14:01:23
Master key	E	128/256 bit
Master salt	E	128/256 bit
Policies	E	Policies for e.g. - which users that are allowed to use the key - which applications are allowed to derive keys - middlebox access allowed or not - use as base key for session key generation – implies generation of a Mod_B - etc
…	E	Other parameters
MAC		Message authentication code

7.1.4.2 Binding between user and ticket recipient identities

The use of the recipient field in tickets may vary according to applied policies per a user group. The recipient field may be used to specify a specific user, a group of users or any user. The naming of users and user groups may follow normal IMS conventions and may be extended with use of wildcards. Example recipient fields would then be

1. SIP: firstname.lastname@operator.com
2. SIP:firstname.lastname@enterprise.com
3. SIP: *.lastname@operator .com
4. SIP: *@enterprise.com
5. SIP:*@*

For the user group denoted as the general public it would be natural to have a policy allowing any recipient (see 5 in list above) to use a given ticket. For enterprise users it may be natural to have as a default that tickets are issued with a recipient field (see 4 in list above) allowing any user within the enterprise to use the ticket. In public safety organizations it is probably stronger requirements on knowing that a call is answered by the intended user so there the default is to insert the identity of a specific user (see 1 in list above).

To ensure that ticket polices are enforced it also necessary that a public user identity can be securely bound to an IMS UE and the ticket resolution request. The solution here is to bind the identity used by the user to authenticate against the KMS to a (set of) public identity. Using GBA as an example the procedure would in principle work as follows on the terminating side:

1. The IMS UE on the terminating side performs a GBA bootstrap.
2. The IMS UE on the terminating side submits a ticket to the KMS together with the BTID obtained in the bootstrap.
3. The KMS, acting as a NAF, requests a NAF-Key for delivering the media master key to the user identified by the BTID. The KMS also requests the public identities associated with this user. This request for public identities may either be served by the BSF or the KMS may obtain the private identity from the BSF and then request the public identifiers from the HSS.
4. The KMS reads the recipient field of the ticket and checks if any of the user's public identities match the identity in the recipients field. If there is a match the KMS performs any key derivations needed and protects the key(s) based on the NAF-Key received from the BSF.
5. The KMS sends the key material to the IMS UE.
6. The IMS UE derives the NAF-Key and reads the key material.

If GBA is not used the mapping between the user identity used for authentication against the KMS has to be bound to the users public identities in some other way.

7.1.4.3 Interoperability between users in different KMS domains

Users in different KMS domains will have their protected tickets generated by different KMSs. The fundamental issue is that a user in general only can be expected to have a trust relation with a single KMS. Thus to ensure that policies are enforced the "home" KMS has at least to be involved in the authentication of the user requesting a new ticket or of a user presenting a ticket for key retrieval. Thus, if one user in one KMS domain shall be able to establish a secure call to a user in another KMS domain the involved KMSs have to cooperate and there has to be a trust relation between KMS A and KMS B. Cooperating KMSs have to be able to exchange secure messages.

Assume that user A in KMS domain A needs a ticket for user B in KMS domain B. Then the procedure can be laid out as follows

1. User A requests a ticket for user B from its home KMS, i.e. KMS A
2. User A sends the ticket to user B
3. User B presents the ticket to his home KMS, i.e. KMS B
4. KMS B sends the ticket to KMS A with an assertion about the identity of the user submitting the ticket.
5. KMS A checks that the identity of the submitter is a valid receiver of the ticket.
6. The KMS generates the keys and other information according to the ticket policies and KMS B request.
7. KMS A sends the keys and the information to KMS B in a protected way.
8. KMS B sends the keys and the information to user B in a protected way.

7.1.4.4 Session and forking keys

7.1.4.4.1 General aspects

One issue with key management systems based on key management services from a central server may be load on the central server. One way to limit the load would be to allow derivation of several independent session keys from one ticket and in this way reduce the number of ticket requests received by the KMS in scenarios where there are different (e.g. consecutive) sessions between one sender and a recipient. Note that the ticket recipient filed may define a group of recipients, see clause 7.1.4.2), which would allow a ticket to be reused for all users in the indicated group. An efficient procedure to derive such session keys is presented below. Another issue is that when a call is forked to several terminating devices, all the terminating devices should use unique keys and not have access to the keys derived/used by other endpoints. A procedure to ensure that this requirement is fulfilled and which in general enforces that all endpoints receiving a given ticket will use different keys is also described.

We use the following notation:

KT_A	The master key in a ticket generated on request from A
KF_AB	Forking Key for user B, i.e. a key unique for user B based on KT_A
KS_AB	Session Key for use between user A and user B
KDF_S (K, Mod)	Key derivation function used to derive a session key from a key K and a modifier Mod
KDF_F (K, Mod)	Key derivation function used to derive a forking key from a key K and a modifier Mod
Mod_A	A (pseudo-)random modifier generated by the initiating party.
Mod B	A (pseudo-)random modifier used at the receiving end.

The methods and procedures described below are mainly aimed for systems using protected tickets. However, the principle behind the procedures for how keys should be generated in forking scenarios is also valid when unprotected tickets are used.

7.1.4.4.2 Session keys

The proposed way to generate session keys is straightforward and only makes use of a key derivation function modifying a base key with help of a random value, Mod_A, generated by the initiating party.

KS_AB = KDF_S (KF_AB, Mod_A)

or if no forking key generation is performed, then

KS_AB = KDF_S (KT_A, Mod_A)

The random value, Mod_A, is transported to the receiving end together with the ticket to allow the session key to be derived there. The key derivation at the receiving end could be performed by either the KMS or the IMS UE. If the key derivation is performed by the receiving IMS UE, then only one access from the IMS UE to the KMS is needed. If the key derivation is performed by the KMS, then the master key in the ticket (KT_A) would never be directly exposed to a network element or an IMS UE. This would improve the security of this key, but of course the receiving IMS UE would have to call the KMS for every session. The preferred solution varies depending on the required security level. Since the whole purpose of this mechanism is to re-use the same base key for several sessions, it obviously requires state about previous sessions to be kept in IMS UEs. Other mechanisms like session resumption in TLS may, when available, be an alternative to this mechanism.

7.1.4.4.3 Forking keys

Ticket policies used will determine rules how and when secure connections can be established in forking scenarios. If the ticket indicates a specific user as recipient, then only that user can resolve the ticket and establish a secure call and thus forking will only work with IMS UEs belonging to that user (see also 7.1.4.2). The other extreme is that the ticket recipient is indicated as “any user in domain D” (or even “any user”). Note that in this case, the initiator is assumed willing to accept any responder from within a group and is assumed to accept that level of granularity in the assurance of the identity of the responder. For this to hold, the KMS must here enforce access control so that only users within the indicated group are permitted to resolve the ticket. Note that the proposed solution below ensures that all users allowed to resolve the ticket will get different keys and they will not be able to derive the key for another user.

Note further that, in a general scenario a sender may not be aware whether forking is performed at the receiving side nor may the sender be capable of determining the set of identities of the allowed recipients of the forked call when requesting a ticket from the KMS. In such a scenario, the sender has no choice but sending a ticket for “any user”, but nevertheless he may want assurance about the recipients identity. For the conclusions in this case cf. clause 7.1.4.4.5.

An approach similar to the one used to generate session keys can be taken to generate different keys for different endpoints in a forking scenario meeting the above level of assurance of responder authentication and key separation. For each endpoint, the master key in the ticket is modified by performing a key derivation function on the media master key and a modifying value, Mod_B. The modifier may depend on the identity of B, preferably in a verifiable way, e.g. hash(indentityB, …..), see also below.

KF_AB = KDF_F (KT_A, Mod_B)

A similar idea is presented in the SDES solution. To have strong assurance that the generated keys are unique per IMS UE, the key derivation function shall be performed by the KMS which also enforces access control relative to the granularity specified in the ticket. E.g. if the ticket specifies *@domain.com, then the KMS is assumed to have means to authenticate B as belonging to that group. If the ticket specifies *@* then no access control is enforced (but the KMS may of course log the identity of B, if desired).

It should be noted that even if an IMS UE is not allowed to perform modifying key derivation by itself, the Mod B parameters used in the modification can be reused for all tickets that the IMS UE receives during the lifetime of its current security association with the KMS, e.g if GBA is used to set up this security association then the Mod_B could be reused for the lifetime of the UE’s GBA base key Ks. This would allow receiving IMS UEs to return the modifying value to the initiator before getting a response from the KMS and hence ensure that no extra delays are introduced.

7.1.4.4.4 Combined session and forking key generation

The combined procedures for session key and forking key generation are depicted in Figure 10 below. Note that the figure does not indicate how security is implemented in the communication between the IMS UEs and the KMS.

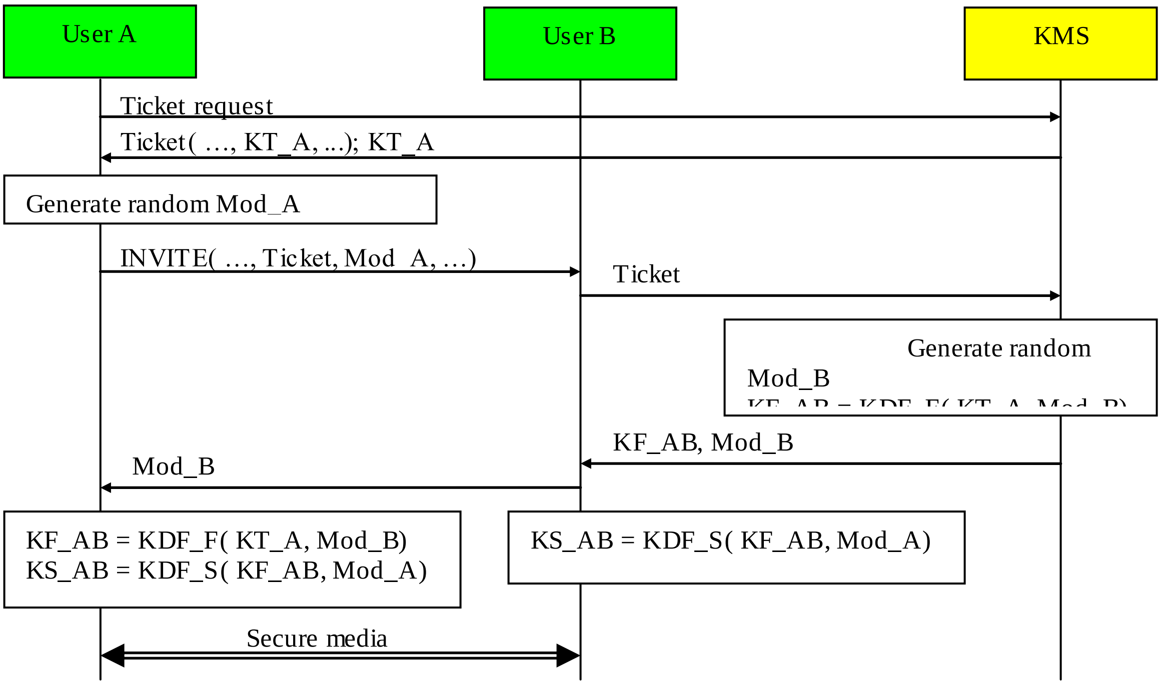
User A
User B
KMS
Ticket request
Ticket(…, KT_A, ...); KT_A
Generate random Mod_A
INVITE(…, Ticket, Mod_A, …)
Ticket
Generate random
Mod_B
KF_AB = KDF_F(KT_A, Mod_B)
KF_AB, Mod_B
Mod_B
KF_AB = KDF_F(KT_A, Mod_B)
KS_AB = KDF_S(KF_AB, Mod_A)
KS_AB = KDF_S(KF_AB, Mod_A)
Secure media

Figure 10: Session and forking key signaling

7.1.4.4.5 Terminating side identity assurance.

Assurance about the terminating side's identity is in many circumstances important. In addition to the standard assurance offered by IMS, TBS can provide additional cryptographically secured identity or group membership assurance. In one approach, the assurance that TBS will provide about the terminating side identity is based on the authorized receivers specified in the ticket, i.e. the set of users in the tickets recipient field. The assurance is obtained as the KMS will only deliver a key associated with a given ticket to a user included in the tickets group of recipients. If a single user is designated as ticket recipient, the initiator will for certain know, when the call is successfully established, that the intended user was reached. If a group of users is designated as recipients, e.g. by *@enterprise.com, then the initiator will know that a user in enterprise is answering the call. Which policy the initiator applies for selecting ticket recipients should depend on the use case and required assurance of terminating side identity

In forking and retargeting use cases a call may terminate in a phone not registered (in IMS and/or KMS) by the intended receiver. In such cases the caller can not get assurance that the intended receiver is the person answering. The only assurance about the terminating side identity that the caller can get is the identity of the user which has registered the phone. It is questionable if this information in general would help the caller to decide if he has reached the person he targeted or not. Thus, if the user needs strong confirmation that he has reached a terminating device belonging to a specific user, he should only have this user as ticket recipient. If the call is forked/retargeted and rejected because the ticket is not valid for the terminating side, the initiator has to request a new ticket and call again. This has the benefit that the caller has full information about what is happening and the drawback that additional SIP signaling is needed. However, if the main concern only is that no-one should be able to intercept a call, a ticket using *@* as recipient can be used and it would allow arbitrary forking and retargeting. Once again this shows that the choice of rules that he initiator should apply for selecting ticket recipients is a policy issue.

In a scenario where the sender sends a ticket for "any user", he may still want to have assurance about the identity of the recipient established in the forking process, cf. clause 7.1.4.4.3. Using another approach than the one described above, it would be possible to let the KMS certify a user identity associated with the device used to resolve the ticket. A simple means to do this would be let the Mod_B be composed of the user identity and a random component (U_Id, Rnd) and have this signed by the forking key or a key derived in a similar way. The details of this solution should be worked out in conjunction with the complete key management procedure for forking and session keys.

7.1.4.5 Unprotected tickets

By using the MIKEY-NULL mode defined in [19], the ticket based solution described here allows a mode similar to SDES, i.e. keys and other set-up information are carried in plaintext in the ticket. This mode is called the unprotected ticket mode. The security properties of this mode are identical to those of SDES: It requires secure transport between all SIP proxies and exposes plaintext keys to all SIP proxies. It also relies on an external mechanism (such as IPSec or TLS) to protect signalling between the IMS UE and the P-CSCF. This mode also provides similar efficiency to SDES, as the communicating IMS UEs do not need to contact the KMS. However, interoperation with current or future IMS UEs implementing SDES according to RFC 4568 will not be possible. This lack of interoperability can be accommodated, however, by a function at the network edge to translate an unprotected ticket setting up SRTP into SDES. Of course, none of the security services offered by a protected ticket will be available, but on the other hand it also cannot be expected that standard SDES equipment supports such services.

7.1.4.6 Ticket replay protection

An attacker of a TBS may try to replay a ticket and in this way be able to find an attack which would give him access to plaintext and/or modify protected media content. Thus a TBS must implement mechanisms rendering such attacks useless. The following mechanisms can be used to prevent replay of tickets:

1. Ticket expiry time. Short life-time tickets can be used to limit the possibilities for replay but will itself not be sufficient to stop attacks. An attacker can e.g. replay a ticket against a device associated with the attacker himself and which has a fast response. Such a replay attack may help in retrieving the target media plaintext if the media protection key generated will be the same as that used for the target media.

2. Tickets could have unique recipients. This would stop an attacker from using other devices than the intended receiver's device as a tool in the replay attack. Still additional means would be needed to stop replay, e.g. a replay counter or some similar mechanism.

3. Keys generated from received tickets can be (pseudo-)randomly modified as suggested above. The way that forking keys are generated is an example of how this can be performed. If the recipient of a ticket always

submits it to the KMS to obtain a forking key, then this modification of the key will be (pseudo-)random and not controllable by the attacker as a new forking key will be generated each time.

4. Receivers could keep a record of received tickets for the ticket's remaining lifetime. This would not stop attacks where replay is initiated on another device.

The conclusion from the above is that having recipient unique tickets and a replay counter is one way of achieving replay protection. Another way is to have the receiving end modify the tickets master key in a (pseudo-)random way before it is used for media protection. The latter method works even if the tickets don't specify a unique recipient.

7.1.4.7 Limiting KMS statefulness

As noted in analysis in clause 7.1.2 the statefulness of the KMS will depend on the choice of the protocol used between IMS UEs and the KMS. To limit the need to store state, this protocol should not build upon a security session like TLS but apply application layer security. In this way the only per IMS UE SA information that the KMS must store between message exchanges is an IMS UE identity and the corresponding key. This SA can be set-up using GBA or other credentials as discussed above. Note also that the interaction between the IMS UE and the KMS always is IMS UE initiated which allows a standard client server design. Furthermore, it can be noted that due to the way keys are generated and retrieved and the possibility to include indications of intended use in the tickets, there is no need for a general replay protection mechanism (see also clause 7.1.4.6).

7.1.4.8 Lawful intercept

First of all we note that use of encryption, LI and related dependences are subjected to national regulations. We also note that possible roaming scenarios and agreements between operators need to be considered. These facts would make a complete analysis of requirements and possible solutions quite extensive and we leave that work to the SA3-LI groups. In the following analysis we limit ourselves to the situation when it can be assumed that user traffic always is routed via the home network.

To be able to provide a clear copy of intercepted communication, the following conditions have to be fulfilled:

1. It must be possible to intercept the traffic (both signalling and media).

2. It must be possible to intercept the ticket and other signalling information (and correlate ticket and traffic).

3. If the ticket is a protected ticket, the keys used for actual traffic protection have to be available. To make the keys available from protected tickets some KMS functions/services would be required.

With media traffic routed via the home network, intercept of the media traffic in the home network will always be possible. So the focus here is on the issues with respect to intercept of tickets and retrieval of key information; if tickets can be intercepted in the signalling plane then so can all other information needed. Intercept of tickets in the home network can be done at SIP server(s). In roaming situations, as the SIP signalling traffic normally is confidentiality protected between the IMS UE and the P-CSCF and considering that in current deployments the P-CSCF is located in the home network, the SIP signalling is only available in encrypted format at bearer level in the visited network.

If an unprotected ticket is intercepted then all key information needed for performing LI can be derived directly from it. When a protected ticket is used, KMS operations on the ticket have to be performed to make the keys needed for LI available and such operations could only be performed by a KMS, i.e. either the ticket issuing KMS or a KMS which interoperates with the ticket issuing KMS.

NOTE: It is essential that the keys to allow decryption in case of LI are provided on per target basis by the KMS in order to ensure confidentiality of the communications that are not to be intercepted..

A summary of the discussion above is that if the SIP signalling is protected and that the P-CSCF always is located in the home network, which seems to be the normal situation in current IMS deployments, intercept of SIP signalling and decrypted content will be possible in the home network. For roaming scenarios, while encrypted SIP signalling and content will always be available, in order to intercept SIP signalling and decrypt the content of communication, one of the following options would be required:

1. SIP signalling is performed in plain between IMS UE and P-CSCF;

2. The P-CSCF is located in the visited network.

3. Keys are provided by alternative mechanisms from the KMS handling entity.

SIP signalling in plain between IMS UE and P-CSCF rules out the use of unprotected tickets while use of protected tickets would allow intercept of the ticket but would require involvement of KMS functionality, i.e. there has to be an interoperation agreement between the visited network and the entity handling KMS. Typically, the KMS will reside in the home network so that, for LI performed by the visited network, cooperation with the home network is needed. With the P-CSCF in the visited network, intercept of tickets would always be possible while also in this case there has to be an interoperation agreement between the visited network and the entity handling KMS functionality/services.

In line with LI standards, when the VPLMN is not involved in the encryption, only encrypted content would be available for LI in the VPLMN.

7.1.4.9 Access to KMS services when roaming

The simplest solution to access KMS services when roaming, would be to go directly to the home KMS. Another solution would be to go to a KMS in the visited network. The KMS in the visited network could then either act as a proxy or it could authenticate the user and perform the KMS services.

Having a KMS in the visited network could possibly help in facilitating autonomous LI in the visited network which would be on the upside of such a solution. On the downside we have the difficulty for the IMS UE to discover the KMS in a visited network. Note that here we talk about a visited IMS network, not a visited access network; if we would consider visited in terms of access then there would be interactions between access and IMS networks and this would make the solution much more complex and out of scope for this standard. On the downside we also note that even if the ticket information becomes available it might be difficult to correlate it to signalling and traffic flows. And the final downside aspect is if the KMS in the visited network must authenticate users. Then all KMS in visited networks have to have access to user authentication functions in the home network / home KMS domain; an example if authentication is based on GBA would be that all KMSs in visited networks would need access the users home BSF.

Thus it seems that the minor benefits from LI point of view does not motivate the complexity of a solution mandating that access to KMS services should always take place via local KMS in a visited network. This may however change in the future.

7.1.4.10 End-to-middle scenarios

In end-to middle scenarios media protection is between an IMS UE and a network entity. The media protection can be set-up either from the IMS UEs or from the network side. Note that to have efficient procedures for network initiated media protection, IMS UE capabilities should be registered with the network. A typical example when end-to–middle may be relevant is when a call is setup between IMS and PSTN networks and in such cases it would be natural to have the MGW to perform security functions on behalf of the PSTN network and we use this scenario to exemplify how media protection is terminated/initiated in the IMS network.

In TBS, end-to-middle traffic cases have to be explicitly allowed by the ticket policies. Assuming this is the case then, in a scenario when the call is initiated from an IMS UE, the set up of the call would follow the same principles as for an end-to-end protected call. The initiating IMS UE requests a ticket from the KMS if protected tickets are used, otherwise it generates a ticket. The ticket is sent together with the INVITE. The MGWC intercepts the ticket. If protected tickets are used, the MGWC contacts the KMS to retrieve the keys used for media protection in the same way as a receiving IMS UE would have done. The MGWC then sets up the MGW to have media security towards the IMS UE. The media traffic is forwarded in plain in the PSTN.

For incoming calls to IMS UEs, the MGWC checks that at least one terminal registered for the intended recipient has registered media security capabilities and preferences. If there is no media protection capable terminal the call is forwarded in plain. Otherwise the MGWC requests a ticket for the recipient from the KMS if protected tickets are to be used, otherwise it generates an unprotected ticket. The MGWC then inserts the ticket in the INVITE and initiates use of media security in the MGW on the media traffic between the MGW and the IMS terminal.

There are other use cases where end-to-middle protection might be required, e.g. when transcoding has to be performed or if only end-to-access_edge protection is allowed. All these cases will follow the procedures described above and they will rely on a controlling entity intercepting and/or injecting tickets in the SIP signalling flow and which also controls the media protection functions in an associated MRF.

7.1.5 Evaluation of solution against requirements.

7.1.5.1 Compliance of TBS with 3GPP Requirements

7.1.5.1.1 LI requirements

The Ticket Based Solution allows compliance with LI requirements in the home network. If unprotected tickets are used the master keys for protecting the communication are known to the P-CSCF and any other SIP proxy processing the INVITE dialogue. When using protected tickets, the LI system must have access to standard user services from a KMS. LI may also be possible in visited networks through the use of unprotected tickets. It should be noted, however, that LI may be difficult in a roaming situation when protected tickets are used and will require cooperation between KMSs in the visited and home networks:

7.1.5.1.2 Security requirements

The TBS is in principle a stand alone key management function which can support keying of any type of media protection protocols including protocols supporting deferred delivery. In the following discussion of the compliance with the security requirements, it is therefore assumed that user plane traffic is properly secured based on the keys established using TBS.

If unprotected tickets are used the SIP traffic has to be secured, i.e. integrity and confidentiality has to be provided for all IMS signalling traffic and especially for the signalling traffic between the IMS UE and the P-CSCF. The security requirements and the needed/offered security for unprotected tickets are exactly as described for SDES in clause 7.3.2. Use of unprotected tickets will exhibit the same security features as SDES in a non-IMS environment. The level of trust may or may not be lowered by the involvement of foreign networks.

The conclusion is that unprotected tickets will comply with the 3GPP security requirements in exactly the same way as SDES does.

A TBS with a KMS will protect the tickets themselves, independent of any SIP signalling protection assumptions, and thus provides security on its own. It will also protect tickets and key information while stored or handled in SIP proxies. TBS may also implement different authorization and group key management schemes, i.e. they support secure group communication using a group key. This ticket protection mechanism is independent of IMS and may be extended to cover also non-IMS environments.

The KMS itself may be a target for attacks. It should be protected in the same way as would a BSF in a GAA/GBA deployment.

7.1.5.1.3 Requirements related to SIP based call features

Concerning forking/retargeting and support of early media, clause 5 doesn't state 3GPP specific requirements – only the IETF requirements apply, cf. further below.

TBS supports secure multiparty communications.

7.1.5.1.4 Architectural requirements

Clause 5.6 lists eleven architectural 3GPP requirements. Compliance of TBS with these requirements is obvious in most cases. Only the most important ones are discussed in this clause.

TBS supports e2e security as well as e2m and e2ae security (see clause 7.1.4.10 for discussion about e2m/e2ae). This is true for both unprotected and protected tickets. When protected tickets are used, a network node needs to have authorization to access the KMS to resolve the key in the ticket.

The requirement to support media recording is supported by TBS independent of if the recording is of plaintext media or if it should be protected. Any issues with recording of protected media are related to the media protection protocol used.

TBS can be implemented in non-IMS UEs from a technical point of view, but from a practical point of view, TBS may or may not be implemented.

The solution comprises two variants, “unprotected tickets, no KMS” versus “protected tickets, usage of a KMS”. Concerning Req 25 (no multiple solutions), it can be stated, that these two variants share the “ticket” as a format for delivering a key and other cryptographic parameters, which allows set up of systems having different functional and security features.

Concerning Req 28 (impact on existing network entities), it should be taken into account that the new function KMS may be deployed in an already existing network equipment which would reduce OPEX and CAPEX, but of course cause an impact on the existing network entity. Network nodes that need to control media protection functionality in e2m scenarios would of course also be impacted.

All different instances of KMS function in different IMSs must be interconnected by security associations, for example in a similar manner how S-CSCFs are connected, when interoperability using protected tickets is required.

7.1.5.1.5 Scalability, cost and performance

Obviously, the unprotected ticket version of TBS complies very well with these requirements, as it is a very simple, straight forward approach without the need of additional network elements, expensive computations, multiple roundtrips etc.

TBS with a KMS offering use of protected tickets will require a KMS supporting all its users. Its size/performance grows proportional to the number of users. However, there is no technical challenge to implement a KMS supporting all IMS users of an operator as can be seen from specifications of and implementations of other nodes in cellular and IMS systems. The only issue might be cost, otherwise, the KMS functionality is simple and likely to have small and efficient implementation, it does not have to store any session state.

Note that TBS offers an opportunity to have some groups rely on protected tickets while other groups rely on unprotected tickets together with trust in the IMS signalling and infrastructure security. Other groups may want to implement the KMS as an external trust anchor independent of IMS. These options allow better operator control of the growth in demand on the KMS.

7.1.5.1.6 Requirements regarding the access network type

TBS complies with these requirements, in the following way:

- it is access network independent;
- it may or may not leverage the IMS security architecture depending on user requirements (for protected tickets, it requires an additional KMS-infrastructure and for unprotected tickets its not required);
- it may or may not work independently of any of the different authentication methods defined for IMS.

7.1.5.1.7 Backward compatibility and migration

TBS complies with these requirements. In particular, keys and other parameters can be negotiated individually for each call. When protected or unprotected tickets are used downgrading attacks cannot be performed if secure SIP signalling is assumed.

With protected tickets, protection against downgrading attacks can be achieved without the secure SIP signalling assumption if IMS UEs implement a policy to only accept calls with “high security” together with making the applied security level visible to the user. It should be noted, however, that, in a general environment with no policy control and automatic acceptance of fallback to unprotected mode which may be a acceptable mode of operation for the general public in a public IMS network, a downgrading attack against the capability negotiation protocol described in clause 6.3, which is based on SIP signalling security, would be still possible for an attacker with access to SIP messages, e.g. by replacing an SDP media description containing SRTP usage by a one that contains only RTP). Downgrading attack could be mitigated by making the user security visible to the user. In such a mode of operation no additional protection against downgrading attacks beyond SIP signalling security would be provided.

7.1.5.1.8 Other requirements

Note that TBS is specified to be independent of the media transport, still allowing tickets to be bound to certain media types and media protection protocols. TBS supports both transport and application layer media protection protocols.

7.1.5.2 Compliance of Ticket Based System with IETF requirements

In this clause, not every IETF requirement is discussed in detail but the important **current** IETF requirements from RFC 5479 [2] are covered.

7.1.5.2.1 Security requirements

The strongest security requirement that currently contain is requirement

R-ACT-ACT: A solution must provide a mode where an attacker, for performing a successful attack, must be active in both the signalling and the media path, and where such an attack would be detectable by the end users.

TBS with protected tickets fulfils this requirement. Unprotected tickets will, in the same way as SDES, of course fail if it is assumed that an attacker can compromise a SIP proxy on the signalling path.

7.1.5.2.2 Forking/retargeting

IETF-requirements

R-FORK-RETARGET: The media security key management protocol MUST securely support forking and retargeting when all endpoints are willing to use SRTP without causing the call setup to fail. This requirement means the endpoints that did not answer the call MUST NOT learn the SRTP keys (in either direction) used by the answering endpoint.

R-DISTINCT: The media security key management protocol MUST be capable of creating distinct, independent cryptographic contexts for each endpoint in a forked session.

How STB can create different keys in a forking scenario is explained in clause 7.1.4.4. If unprotected tickets are used the key modification can be performed by the receiving client in a corresponding way.

In TBS with protected tickets, a sender may authorize the receivers to receive the key from the KMS. For this, he can provide e.g. a list of authorized receivers. How tickets are bound to different receivers or groups of receivers is described in clause 7.1.4.2.

RFC5479 describes that in typical forking/retargeting scenarios, the sender does not know who a call may be forked/retargeted to. This situation is discussed in clause 7.1.4.4 and it is described how tickets can be used and generated to get media security also in forking and retargeting situations. In clause 7.1.4.4 it is further discussed how TBS can offer SIP independent assurance about the terminating side identity. This is done by firstly guaranteeing that it is a legitimate recipient of the ticket that answers the call. If the recipient is defined as a single user, this gives full assurance about the terminating side identity. A second level of assurance can be obtained by having the KMS include such identity information in parameters used in the forking key generation, see clause 7.1.4.4.5.

NOTE: The evaluation was actually not done against RFC5479, but against a version of an Internet Draft, from which RFC5479 evolved.

R-HERFP: The media security key management protocol MUST function securely even in the presence of HERFP signalling.

HERFP behaviour is that in a forked call, rejections of the INVITE sent by different endpoints may be terminated at the forking proxy and never reach the caller. A solution to fulfil this requirement can be accommodated by TBS by not allowing an answerer to send indications about key exchange failures in order to let the offerer “make another try”.

Another IETF-requirement, mentioned under “media considerations”, is also relevant with respect to forking, in case forking leads to a multiparty session:

R-ASSOC: The media security key management protocol SHOULD include a mechanism for associating key management messages with both the signalling traffic that initiated the session and with protected media traffic. Allowing such an association also allows the SDP offerer to avoid performing CPU-consuming operations (e.g., Diffie-Hellman or public key operations) with attackers that have not seen the signalling messages.

With TBS, keys are exchanged in the signalling messages, so association of key management to signalling is clear. Association between key management protocol and media traffic is done implicitly through the context identification used in SRTP. In case of forking, an SIP ACK will only be sent to one of the terminating UEs.

Finally, the following IETF requirement refers to forking/retargeting:

R-BEST-SECURE: Even when some end points of a forked or retargeted call are incapable of using SRTP, a solution MUST be described which allows the establishment of SRTP associations with SRTP-capable endpoints and / or RTP associations with non-SRTP-capable endpoints.

A simple solution to this is that the initiator offers two media streams, one protected and one unprotected. Allowing unencrypted media is of course always a security issue as the user has to be warned if media is not protected.

7.1.5.2.3 Early media

The IETF requirement is

R-AVOID-CLIPPING: The media security key management protocol SHOULD avoid clipping media before SDP answer without requiring Security Preconditions [RFC5027].

TBS will in principle, if the mechanisms to guarantee that different IMS UEs will have different keys in forking scenarios are deployed, need an SDP answer before decryption of media can start which means that encrypted media would be clipped before that. However, 3GPP generally assumes SBCs in the media path that block media before SDP answer anyway and thus TBS does not lead to specific problems here.

As discussed in the corresponding clause for SDES, it may also be considered to allow the usage of unencrypted early media and apply protection only to media after the SDP answer. A straightforward solution here would be for the initiator to offer one plaintext media port and one port for protected content. Such an approach would be in line with the most common way of handling early media in IMS today. It would also obsolete the IETF-requirement:

R-ALLOW-RTP: A solution SHOULD be described which allows RTP media to be received by the calling party until SRTP has been negotiated with the answerer, after which SRTP is preferred over RTP.

Allowing unencrypted media is of course always a security issue as the user has to be warned if media is not protected.

7.1.5.3 Summary requirement compliance

TBS offers a framework which can encompass user groups with differing security requirements. The framework also includes the possibility to allow specific user groups to handle their key management. TBS with a KMS will offer a solution which would comply with all security requirements. Unprotected tickets could be an alternative for environments in which security requirements aren't that strict. Thus TBS offers great flexibility.

The high security provided by TBS solution must be balanced against implementation and performance.

7.2 Using IMS AKA keys for media protection over the access network

7.2.1 Requirements

The following list of requirements are used as a starting point for the solution architecture:

1. The security shall be between the IMS UE and a protection end-point (MSF, Media Security Function) at the edge of the IMS trusted environment.

2. No new credentials shall be needed for the SA (key) establishment between the IMS UE and the MSF.

3. It shall be possible to protect RTP and MSRP traffic.

4. The IMS operator shall be able to control the use of the protection mechanism

5. The control of the protection mechanism shall be realized by SIP signaling.

The requirements are fulfilled as follows:

Requirement 1 is fulfilled by introduction of a new functionality, the MSF, which possibly could be part of e.g. an IMS Access Gateway.

Requirement 2 can be fulfilled by basing the security on a shared secret key obtained from a shared SA used for SIP authentication and/or signaling protection. This is a straightforward solution when user authentication is based on IMS AKA and the associated CK, IK is used in the protection of the SIP signaling between the IMS UE and the P-CSCF. (The CK, IK could be passed through a PDF to generate a media security master key.) When TLS is used to protect the SIP, the shared SA in TLS may be used as the basis for derivation of media protection keys. Finally, the third option is to base the media protection keys on the "password" used in SIP digest authentication. Doing this would give a media protection which has similar strength to the user authentication which might be reasonable, assuming strong and long passwords.

This requirement can also be fulfilled by having the client or the network generate a master key which can be used to derive the needed media protection keys and distribute this master key in SDP by eg. SDES. This would require that the SIP signaling is confidentiality protected.

Requirement 3 is fulfilled by employing SRTP and PSK-TLS. For SRTP the session keys may be e.g. generated with MIKEY. PSK-TLS has its own inbuilt session key generation mechanism. Other SA information is exchanged within SIP/SDP re-using the existing IETF SDESC mechanism.

Requirement 4 and 5 are fulfilled by defining IMS UE security capabilities which the IMS UE includes when registering. The IMS UE may then propose the use of access security or the proposal may come from the network. The network will always be able to decline an invitation / not issue one.

7.2.2 Architecture

The architecture for IMS media access security is depicted in Figure 11. The media security master key may emanate from the CK, IK generated by IMS AKA, the master key used by TLS or from the password used in SIP digest. This media security master key is held by the P-CSCF independently of its origin..

It is indicated that the media security master key is delivered from the P-CSCF to the MSF. This is not the only way to handle the distribution; it could probably also be done via e.g. a MRFC in case the functionality would be part of an MRFP.

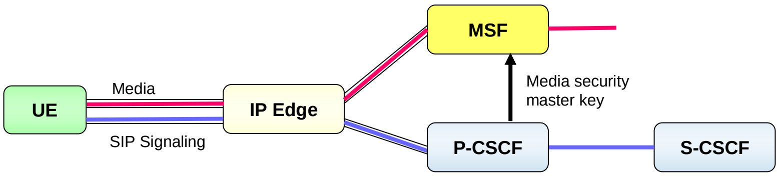

Figure 11: High level architecture for access security

7.2.3 Access security set-up

Figure 12 below shows an example signaling diagram for setting up access security. The first phase, steps 1 to 3 indicates the registration of the IMS UE access security capabilities. The following steps indicate how access security is set up in both access networks. The actual establishment of the media security is not included in the diagram.

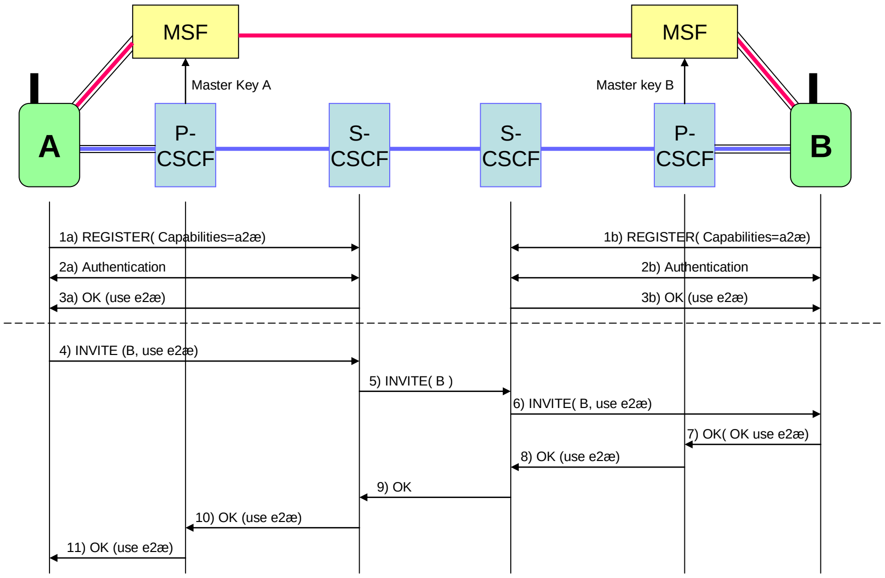

Figure 12: Simplified signalling diagram for access protection

1a/b The IMS UE registers with the IMS system by sending a REGISTER including its capabilities regarding access protection (e2æ).

2a/b The IMS UE is authenticated to make the registration valid.

3a/b The IMS UE gets a 200 OK confirming the registration, and it may acknowledge support of the registered e2æ capability.

4 The IMS UE sends an INVITE containing an offer to use e2æ protection, including parameters for key establishment.

The originating side P-CSCF inspects the INVITE and notices that e2æ protection is proposed. As the network is capable of e2æ protection it tacitly accepts the offer and stores the decision.

5 The originating side S-CSCF performs onwards routing to the terminating side S-CSCF. The originating network may optionally remove the e2æ indicator. If kept, the terminating network will use it as indicator that IMS UEs capable of e2æ should be selected prior other IMS UEs.

The terminating S-CSCF inspects the INVITE and checks if the called party supports e2æ protection.

6 The terminating S-CSCF performs service invocation and onwards routing to the IMS UE. If not present, the terminating network, configured to apply e2æ protection, inserts an e2æ protection offer before the INVITE is forwarded to the IMS UE. The offer includes parameters necessary to establish a shared SA. The SDP must also be changed to route the media via the MSF.

The terminating IMS UE accepts the INVITE including the e2æ offer. It derives the SA to be used and sends it together with a signal to the IMS UE media plane handler instructing it to enable media protection based on the that SA.

7 The terminating IMS UE answers with a 200 OK accepting the e2æ offer. The terminating P-CSCF receives the 200 OK and sees that the access security offer was accepted. It then generates a master key for e2æ protection and pushes it and other information needed to the MSF and requests that it enables media protection.

8. The P-CSCF forwards the 200 OK to the terminating S-CSCF.

9. The terminating S-CSCF forwards the 200 OK to the originating S-CSCF.

10. The originating S-CSCF forwards the 200 OK to the P-CSCF.

The P-CSCF inspects the 200 OK and recalls the decision to use e2æ protection. It generates the master key for e2æ protection. The P-CSCF then push the master key and other information needed by the MSF and a request that the MSF enables media protection.

11. The P-CSCF forwards the 200 OK to the IMS UE. The IMS UE notices that the e2æ protection offer has been accepted and derives the master key to be used. It sends the master key together with a signal to the IMS UE media plane handler, instructing the media plane handler to enable media protection based on the provided SA.

7.2.4 Access security set-up with key mixing

A further enhancement of the methods described in clause 7.2 is the following method of key mixing.

The security of e2m media plane protection is under current assumptions in the TBS (unprotected ticket) and SDES solutions based on the fact that SIP signaling between the IMS UE and the P-CSCF is secure. This means that media plane security cannot be guaranteed if this signaling link is unprotected or only integrity protected; confidentiality protection is thus required for the SIP signalling.

Note that it would be possible to combine the use of end-point generated keys as described for TBS and SDES with a shared secret as describe here in clause 7.2 by mixing the two keys together. If we do this, the requirement on having SIP signaling confidentiality protected over the access link would go away when a shared secret exists and the security would in general be improved. Adopting a solution including such key mixing would mean that the solution would be able to cope with both the situation that a shared secret exists and the situation that there is no shared secret.

The effect of the key mixing would of course be only beneficial when SIP signalling is unencrypted. The key mixing would guarantee that intercept of the plain signalling would not help a wiretapper in obtaining the media key in plaintext. Applying key mixing also when the SIP signalling is confidentiality protected would not give any substantial increase in security, but would neither be detrimental. It would however help converge procedures and avoid having to handle different security set-up cases, given that a shared key exists.

To have a straightforward solution it could be considered to use key-mixing only when user authentication is based on ISIM and AKA. In this case the Ck, Ik will be available in the P-CSCF and a key which could be combined with an end-point generated key could, as indicated in clause 7.2.1, easily be derived. Note that there is no requirement that the initiating end requests that such key mixing takes place as both ends will a priori know when ISIM and AKA is used for authentication and the IMS UE will by signalling know the location of the other media security termination point.

Assume that an initiating IMS UE generates a key K_ep and that the IMS UE and the P-CSCF share Ck and Ik, the key to be used for media protection could in principle be derived as K = PRF(Ck, Ik, K_ep).

NOTE: A replay mechanism needs to be defined.

7.3 Security Descriptions (SDES)

7.3.1 Brief description of SDES

RFC 4568 “Session Description Protocol (SDP) Security Descriptions for Media Streams” defines a Session Description Protocol (SDP) cryptographic attribute for unicast media streams. The attribute describes a cryptographic key and other parameters that serve to configure security for a unicast media stream in either a single message or a roundtrip exchange. The attribute can be used with a variety of SDP media transports, and RFC 4568 defines how to use it for the Secure Real-time Transport Protocol (SRTP) unicast media streams. The SDP crypto attribute requires the services of a data security protocol to secure the SDP message. For the use of SDES in IMS, the SIP signalling security mechanisms defined for IMS shall be used, for more details cf. 7.3.2.2.

SDES basically works as follows: when an offerer A and an answerer B establish a SIP session they exchange cryptographic keys for protection of the ensuing exchange of media with SRTP. A includes the key, by which the media sent from A to B is protected, in a SIP message to B, and B responds with a SIP message including a second key, by which the media sent from B to A is protected.

When used in end-to-end mode SDES has no requirements on the network infrastructure, except for Lawful Interception. When used in end-to-middle mode, the requirements on the network infrastructure can be seen from clause 7.3.5.

7.3.2 Compliance of SDES with 3GPP requirements

7.3.2.1 LI requirements

SDES allows to comply easily with any LI requirements, as the master keys for protecting the communication are known to the P-CSCF and any other SIP proxy processing the INVITE dialogue. LI would also be possible in visited networks.

In more detail:

END-TO-END SCENARIOS:

Non-roaming case: there is no problem as the encryption key can be obtained from a node in the SIP signalling path in the home network.

Roaming case:
LI is always possible in the home network, as the S-CSCF resides in the home network and can provide the master keys.

For LI performed by the visited network, we have to distinguish cases according to the SIP signalling encryption methods defined in TS 33.203:

- for Early IMS = GIBA the encryption is at GPRS level and terminates at the SGSN, which is in the visited network. So there is no problem with LI performed by the visited network.
- for NIBA there is no encryption anyhow, and security is based on the assumption of physical security, so there is no problem with LI performed by the visited network.
- for Ipsec and TLS the encryption terminates at the P-CSCF
    - when the P-CSCF is in the visited network there is no problem with LI performed by the visited network.
    - when the P-CSCF is in the home network and SIP signalling encryption is enabled between IMS UE and P-CSCF then an LI entity in the visited network can obtain the key only with the cooperation of the home network. This is not a problem when home network and visited network are under the same jurisdiction, but may be otherwise.

END-TO-MIDDLE SCENARIOS:

The media is always available in the clear at the encryption termination point in the network. There is no problem with LI in the home network. There is no problem with LI in the visited network in roaming situations if the encryption termination point resides in the visited network. The latter is always the case if the encryption termination point resides at the edge of the access network, For SDES, the end-to-middle scenario is described in clause 7.3.5.

7.3.2.2 Security requirements

SDES is only a key exchange mechanism, while the security requirements refer also to the security of the IMS user traffic, i.e. media. For the discussion of the compliance with the security requirements, it is therefore assumed that user plane traffic is properly secured based on the keys exchanged by SDES (e.g. RTP based media traffic is secured by SRTP).

SDES requires the SIP traffic to be secured between the IMS UE and the P-CSCF. Several alternatives are available for that. In particular, Ipsec (with IMS AKA) and TLS (with SIPS as in RFC3261 or as in TS 33.203 Annex O) are specified. SDES provides the same level of security for IMS media which is also provided by IMS for SIP signalling. So, the user can place the same degree of trust on media security as on signalling security. Within the core network, SDES requires secure transport between all SIP proxies and trust in all SIP proxies. Between the SIP proxies, security can be provided according to the principles of NDS/IP. On the SIP proxies, however, the keys transported with SDES become visible in plaintext. Therefore, the SIP proxies must be trusted. SDES is not compliant with the requirement to protect IMS user traffic against on core network nodes. It is still open if this requirement can be relaxed (see NOTE 1 of clause 5.4).

Against parties that do not control one of the involved SIP proxies, SDES with hop-by-hop protection between all involved SIP agents provides security for the key exchange. Combined with the media plane security protocol SRTP, the security will be higher as for insecured sessions. It should be noted that as media keys will be available/transported in plaintext in all SIP proxies, compromise of these proxies will allow, not only signalling security, but also media security, to be compromised. However, even if media keys are not exposed to the proxies, the proxies need to be protected anyway to secure SIP signalling which is an important requirement for operators and users.

Within the IMS, protection of the SIP traffic can be expected to be available, using the IMS access security mechanisms and NDS/IP. Outside the IMS, at least SIPS (with hop-by-hop TLS) is likely to be supported. It is unclear, how non-IMS SIP providers secure their SIP proxies. This makes SDES appear less secure in a non-IMS environment. On the other hand, from the perspective of an end user, it may not make much difference whether a foreign network, which transports the signalling traffic, is an IMS or not. Typically, the level of trust is lowered by the involvement of foreign networks, be they IMSs or not.

Requirement 7 is satisfied by SDES by using signalling integrity and assertion of identities (P-Asserted-Identity), which prevents spoofing of the user identities IMPI/IMPU. Note that the caller can decide to cancel the call if it is terminated by an undesired callee.

The conclusion is that with the assumptions described in the paragraphs above, SDES complies with the 3GPP security requirements for sessions within an IMS environment, except possibly with Requirement 5. Outside the IMS environment, this may or may not be the case, depending on the availability of SIPS and the trustworthiness of the involved non-IMS-SIP providers.

7.3.2.3 Requirements related to SIP based call features

Concerning forking/retargeting and support of early media, clause 5 doesn't state 3GPP specific requirements – only the IETF requirements apply, cf. further below.

Concerning secure multiparty communications, it must be noted that SDES according to RFC 4568 is currently restricted to point-to-point unicast communication, and multicast is for further study. However, SDES allows each sender to choose a key for the traffic it sends, which may be used as a basis for the support of efficient multicast, where a sender doesn't need to protect the traffic it sends differently for different receivers. A further discussion on needed extensions to SDES to more fully support multicast can be found in clause 7.3.6.4 on Multicast Support.

7.3.2.4 Architectural requirements

Clause 5.6 lists eleven architectural 3GPP requirements. Compliance of SDES with these requirements is obvious in most cases. Only the most important ones are discussed in this clause.

SDES supports security between SIP endpoints, i.e. end-to-end security. A SIP endpoint could also be a network node, e.g. a SIP application server. So the case of end-to-middle security where the node terminating the media plane security within the network is a SIP endpoint is clearly also supported.

End-to-middle security further comprises also cases, where the node within the network that terminates the media plane security is not a SIP endpoint or is not in the signalling path at all. E.g., it could be a PSTN-MGW or an IP-IP-MGW performing transcoding. Such a MGW will be controlled by some node that is aware of the SIP signalling, and thus knows the keys transmitted with SDES. It is assumed that technically, it is rather easy and straightforward to enhance the control protocol between controller and MGW, e.g. H.248, to support sending the key to the MGW. For more details on how SDES can support end-to-middle scenarios, see clause 7.3.5.

The requirement to support media recording is marked as "ffs" in clause 5. The requirement doesn't specify any details on what kind of recording it refers to. One can imagine various scenarios for media recording, in particular recording of encrypted or plaintext media within the network. Recording plaintext media means terminating the media plane security within the network, which is supported by SDES as described in the previous paragraph. In case encrypted media has to be recorded (e.g. the "deferred delivery" as described in the use case in clause 4.1, SDES would allow to store the key together with the encrypted message. The same level of trust can be assumed for a network node recording encrypted messages as for any SIP proxy that handles the keys transmitted with SDES.

Concerning the interoperability with non-IMS-capable UEs, SDES provides a possible basis, as SDES is a standards track RFC of the IETF. SDES is already widely deployed in IMS UEs – currently it is the de facto interoperability standard for "IETF-compliant" equipment that supports SRTP. (Quotation from the summary report of the SIPit22 interoperability test event on April 14-18, 2008 (https://www.sipit.net/SIPit22_Summary): "There was a significant amount of successful SRTP interop at this event.... Most of the tests established the session using sdes."). However, it should be noted that IETF currently promotes a new key management solution called DTLS-SRTP.

7.3.2.5 Scalability, cost and performance

SDES complies with these requirements, as it is a simple, straight forward approach without the need of additional network elements, expensive computations, multiple roundtrips etc. provided that the IMS infrastructure provides the security for the SIP signalling and the SIP proxies required. If such security is not provided, cost and complexity of an update to provide it may be substantial. SIP proxies handling SDES crypto attributes may need enhanced security.

7.3.2.6 Requirements regarding the access network type

SDES complies with these requirements. In particular,

- it is access network independent;
- it leverages the IMS security architecture in that it is totally reliant on it;
- it works independently of any of the different authentication methods defined for IMS.

7.3.2.7 Backward compatibility and migration

SDES complies with these requirements. In particular, keys and other parameters can be negotiated individually for each call, and downgrading attacks cannot be done in the secure signalling environment that is assumed.

7.3.2.8 Other requirements

RFC 4568 currently only describes the usage of SDES for exchanging keys and other crypto parameters for securing RTP based media traffic by SRTP (RFC 3711). However, RFC 4568 indicates that SDES could also be used for exchanging keys for other media plane security protocols, by defining additional forms of “crypto objects”. For example, an enhanced SDES may be used to establish a “shared key” for TLS-PSK (RFC 4279), thus allowing to secure TCP based media traffic. (According to RFC4568, each party currently provides one master key for securing the media traffic it will send. For TLS-PSK, a single shared secret is needed. This could be generated by applying a hash function or pseudo random function to the combined keys provided by the two parties. This will create a single shared secret and at the same time solve any issues with forking and retargeting in this scenario. See also clause 7.3.3.2.)

Note that, according to requirement 38, it may be acceptable to standardise more than one solution if this offers lower complexity. In this case SDES could be used in the case of RTP traffic, and either an enhancement of SDES to support key management for protocols other than RTP or different solutions may be used.

Protection of media transmitted with SIP messages (e.g. using SIP MESSAGE), would not require any additional measures in the SDES approach, as a secure signalling path is assumed.

7.3.3 Compliance of SDES with IETF requirements

The IETF requirements are described in RFC 5479 [2].In this clause, not every IETF requirement is discussed in detail. However, the important IETF requirements from RFC 5479 [2] are covered, including the requirements, where there might be doubts about the compliance of SDES.

NOTE: The evaluation was actually not done against RFC5479, but against a version of an Internet Draft, from which RFC5479 evolved.

7.3.3.1 Security requirements

The strongest security requirement in RFC 5479 [2] is requirement R-ACT-ACT: A solution must provide a mode where an attacker, for performing a successful attack, must be active in both the signalling and the media path, and where such an attack would be detectable by the end users. RFC 5479 [2] states, that compliance of a mechanism with such a requirement cannot be evaluated absolutely, but depends on additional assumptions. For example, the ID evaluates DTLS-SRTP as compliant with R-ACT-ACT, assuming that the SIP-proxies performing the authentication service according to RFC4474 are trusted (i.e. not compromised by attackers). Without this assumption, DTLS-SRTP does not satisfy R-ACT-ACT.

In the IMS environment, as long as none of the nodes is compromised, there is no way to break the security of SDES. So in that environment, SDES (which is evaluated in RFC 5479 [2] to be susceptible even against a passive attack) satisfies the strongest security requirement R-ACT-ACT. (Note that, for DTLS-SRTP, only authenticating SIP nodes need to be trusted, while for SDES all SIP nodes in the signalling path needs to be trusted.)

7.3.3.2 Forking/retargeting

Editor's note: None of the IETF requirements from RFC 5479 [2] in this clause are currently contained in clause 5, so, strictly speaking, a solution proposed to 3GPP need not be compatible with these at all.

RFC 5479 [2] states the IETF-requirement

```
   R-FORK-RETARGET:
        The media security key management protocol MUST securely
        support forking and retargeting when all endpoints are willing
        to use SRTP without causing the call setup to fail.  This
        requirement means the endpoints that did not answer the call
        MUST NOT learn the SRTP keys (in either direction) used by the
        answering endpoint.
```

Without modifications, SDES is not compliant with this requirement. For certain scenarios it can be argued, that forking and retargeting will be performed between endpoints that have a close relationship and possibly also a high degree of trust between each other. In such forking/retargeting scenarios SDES could be considered sufficiently secure. Examples of such forking / retargeting scenarios include: forking to different IMS UEs controlled by a single user, forking to different clerks in a call center that all process the same types of requests, retargeting from a person to its substitute. In other scenarios like when a conversation is privileged between e.g. a counsellor and a client, such trust assumptions do not hold true and other solutions would be required.

In scenarios, where such a level of trust between forked/retargeted endpoints cannot be assumed, an obvious workaround is to rekey the session with only that parties that actually participate in the session. This would require an UPDATE or re-INVITE and therefore some additional signalling. A problem is that the inviting party may not always be aware of the fact that other, non-responding endpoints may have received the SDP offer, and therefore must rekey for every session. This problem might be alleviated by letting the answerer perform the rekeying, assuming that the answerer knows, whether forking/retargeting is configured for the chosen URI, and whether it perhaps acceptable that other forked endpoints have got the key. The above text assumes a setup where parties just know it. This will be the case in many scenarios, e.g., a clerk in a call center will know that incoming calls are forked to all clerks, and a user will know that he has configured his account in a way that incoming calls are forked to his various IMS UEs etc.

Support of SIP forking is also discussed in the SDES RFC (RFC 4568) itself, in its section 7.3.

RFC 5479 [2] further states the IETF-requirement

```
   R-DISTINCT:
        The media security key management protocol MUST be capable of
        creating distinct, independent cryptographic contexts for each
        endpoint in a forked session.
```

For SDES, if an offerer gets two or more answers, there will be two or more keys for received media. Creating different contexts for received media streams is no problem.

For sending media, creating different contexts per receiver (all with the same key) is possible for the caller (e.g. by instantiating the data structure describing the crypto context per receiver). There maybe be minor issues, e.g. if the key lifetime is expressed by the maximum number of packets that can be encrypted with the key, then it has to be taken into account that the same key is used for different contexts. Note that there is no security problem with using the same (master) key for different flows, as the sender can use different SSRC ids for them (synchronization source ids, see RFC 3550), which results in different key streams for the different flows.

It may also be argued that in a forked session, the caller will not send different streams to the forked endpoints, and therefore doesn't really need different crypto contexts. If the caller however decides to start a session with several endpoints that were reached by forking of the original INVITE, the caller can easily re-INVITE these endpoints and specify new, different keys.

RFC 5479 [2] further states the IETF-requirement

```
   R-HERFP:
        The media security key management protocol MUST function
        securely even in the presence of HERFP behavior, i.e., the
           rejection of key information does not reach the sender.
```

HERFP is briefly explained in RFC 5479 [2]: In a forked call, rejections of the INVITE sent by different endpoints may be terminated at the forking proxy and never reach the caller.

SDES does not comprise mechanisms that allow an answerer to send indications about key exchange failures (in order to let the offerer "make another try"). If a sender has included crypto objects for all crypto suites it is willing to use and does not get a response accepting any of these crypto-objects, there is nothing it could do to establish the crypto session, even if it would have received all the (rejecting) answers from the different endpoints the INVITE has been forked to. So SDES complies with R-HERFP.

Another IETF-requirement, mentioned under "media considerations", is also relevant with respect to forking, in case forking leads to a multiparty session:

```
  R-ASSOC:
           The media security key management protocol SHOULD include a
           mechanism for associating key management messages with both
           the signaling traffic that initiated the session and with
           protected media traffic.  It is useful to associate key
           management messages with call signaling messages, as this
           allows the SDP offerer to avoid performing CPU-consuming
           operations (e.g., Diffie-Hellman or public key operations)
           with attackers that have not seen the signaling messages.

           For example, if using a Diffie-Hellman keying technique
           with security preconditions that forks to 20 endpoints, the
           call initiator would get 20 provisional responses
           containing 20 signed Diffie-Hellman key pairs.  Calculating
           20 Diffie-Hellman secrets and validating signatures can be
           a difficult task for some devices.  Hence, in the case of
           forking, it is not desirable to perform a Diffie-Hellman
           operation with every party, but rather only with the party
           that answers the call (and incur some media clipping).  To
           do this, the signaling and media need to be associated so
           the calling party knows which key management exchange needs
           to be completed.  This might be done by using the transport
           address indicated in the SDP, although NATs can complicate
           this association.
```

With SDES, keys are exchanged in the signalling messages, so association of key management to signalling is clear. SDES has an issue concerning the association of incoming media to the keys transported with SIP signalling, if several endpoints answer on a single INVITE and start sending media. E.g., in an RTP session where A receives on one of its transport addresses (IP address + UDP port) media streams from two parties B and C, B and C will use individual keys, and must also use different SSRC ids (synchronization source ids, see RFC 3550). As SDES doesn't define the transport of SSRCs within the crypto object (but uses the "late binding" approach, see RFC 4568), at the beginning, A will not know, which key to use for which SSRC. In case of an authenticated packet, A can find this out deterministically by trying all received keys. (Note that while authentication is mandatory only for RTCP, but not for RTP, for general security reasons it is highly recommended to authenticate also RTP.)

These additional computations are only needed when a new SSRC id appears. Moreover, they can be avoided completely by using different receive ports for the streams received from different senders. RFC 4568, in its section 7.3, suggests to take this approach. It also states that alternative approaches are possible.

Note further, that SDES doesn't require expensive computations (like DH exchanges), which alleviates the problem of DoS attacks as mentioned in R-ASSOC.

Finally, the following IETF requirement refers to forking/retargeting:

```
  R-BEST-SECURE:
        Even when some end points of a forked or retargeted call are
        incapable of using SRTP, a solution MUST be described which
        allows the establishment of SRTP associations with SRTP-capable
        endpoints and / or RTP associations with non-SRTP-capable
        endpoints.
```

Concerning the multi-party aspects of this, see clause 7.3.6.4. Concerning the usage of RTP instead of SRTP, see the discussion of R-ALLOW-RTP in clause 7.3.3.3.

7.3.3.3 Early media

The respective IETF requirement is

```
   R-AVOID-CLIPPING:
        The media security key management protocol SHOULD avoid
        clipping media before SDP answer without requiring Security
        Preconditions [RFC5027].
```

SDES allows decryption only after successful transmission of the SDP answer, so encrypted media would be clipped before that. However, 3GPP generally assumes SBCs in the media path that block media before SDP answer anyway. So SDES doesn’t lead to a specific problem here.

It may also be considered to allow the usage of unencrypted early media and apply protection only to media after the SDP answer. This may be reasonable for non-sensitive announcements, ring tones, advertisement etc. Usage of RTP for early media and then changing to SRTP after the SDP answer could be specified outside SDES, just by suitable definition of the semantics of an SDP offer specifying an encrypted session. This approach would also be in line with another IETF-requirement:

```
   R-ALLOW-RTP:  A solution SHOULD be described that allows RTP media
        to be received by the calling party until SRTP has been
        negotiated with the answerer, after which SRTP is preferred
        over RTP.
```

7.3.4 Summary requirement compliance

Within an IMS environment assuming trusted SIP proxies and usage of the recommended security mechanisms (e.g. TLS or Ipsec in the access, or Za/Zb interfaces in the core) SDES provides a security level corresponding to the access protection of cellular systems. Outside the IMS, support of SIP over TLS has to be assumed and if applied would protects SIP messages between the proxies. A remaining security risk is that one of the involved operators is malicious or fails to protect its proxies against attackers. It has to be evaluated if this is acceptable for operators as well as the most relevant user groups. Still, it would be an improvement compared to the unencrypted media streams in a “legacy” IMS and the PSTN.

SDES is a lean approach, without needing any involvement of the network and without the need to modify existing networks given that the networks provide the required security mechanisms to have secure SIP signalling and trusted and protected SIP-proxies. Under these conditions it is cost efficient and it scales well. It does not require expensive computations or additional roundtrips, so it does not cause any significant overhead and does not adversely affect any IMS services.

SDES is a mechanism that is already widely deployed in non-IMS UEs – currently it is the de facto interoperability standard for “IETF-compliant” equipment that supports SRTP. However, it should be noted that IETF currently promotes a new key management solution called DTLS-SRTP. SDES allows to comply with any requirements where LI is performed in the home network, and, in many scenarios, also with requirements where LI is performed in the visited network, as the operator has access to the keys exchanged in the signalling messages. Access to the keys is always possible for the home operator, and, in many scenarios, also for the visited operator.

7.3.5 SDES in end-to-middle scenarios

While SDES is suitable as an end-to-end solution, where only the endpoints encrypt/decrypt the media streams, it is usable in a straightforward way also as an end-to-middle solution.

The classical end-to-middle scenario is a call between an IMS endpoint and a PSTN endpoint. Here, the media gateway (plus its controller) can perform the media plane security procedures on behalf of the PSTN endpoint. In this approach, the network chooses a key for protecting the media sent by the PSTN endpoint on the IP based call leg and inserts the key into the SDP body sent to the IMS endpoint. This key, as well as the key provided by the IMS endpoint, are passed to the media gateway which performs encryption/decryption between the PSTN call leg and the IP based call leg.

Figure 13 illustrates the principle of this procedure, for a voice call from an IMS endpoint to a PSTN endpoint. Note that the picture is an abstraction focussing on the SDP offer and answer, not on the SIP messages. It does not show the different SIP roundtrips required for the call setup within the IMS. In addition, it does not show any details of the signalling towards the PSTN.

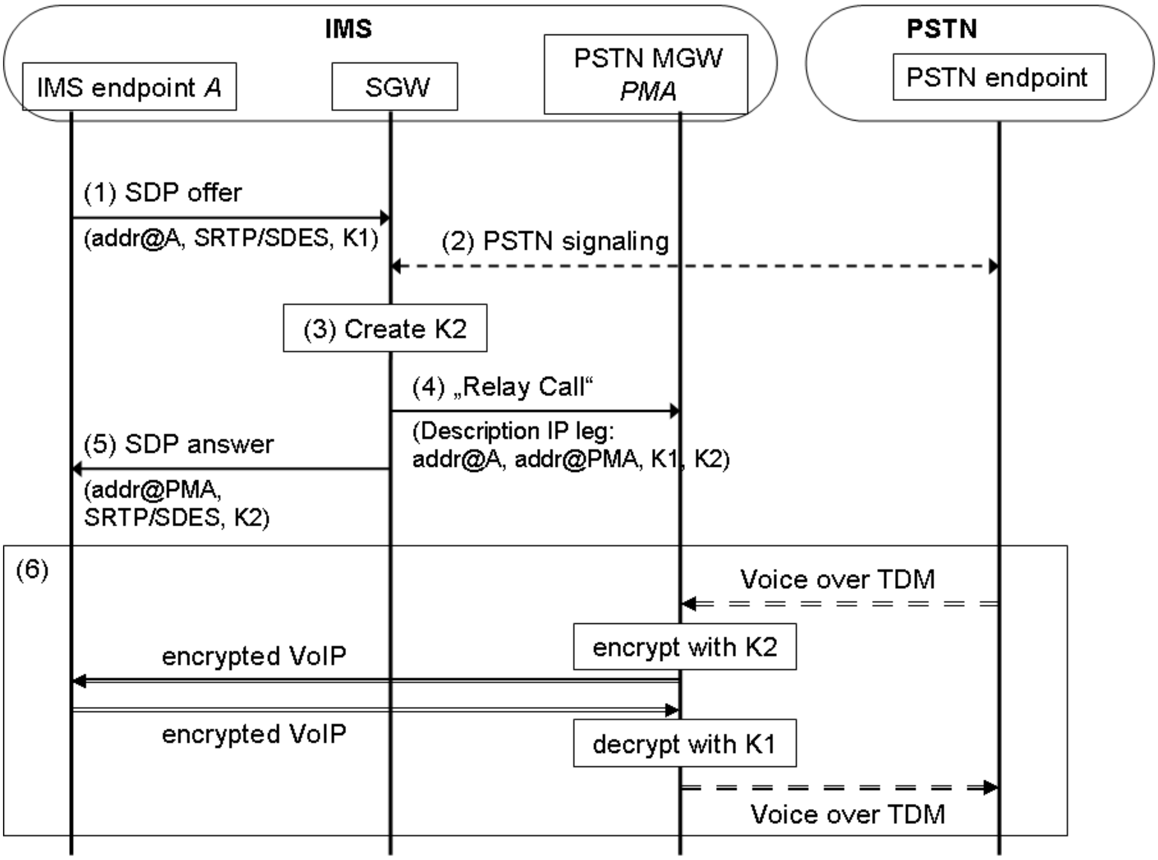

Figure 13: SDES flow with a PSTN end-point

Description:

(1) An SDP offer for an SRTP stream and a crypto object containing a key K1 from an IMS endpoint arrives at the signalling gateway (SGW) controlling the PSTN MGW.

(2) The SGW performs the TDM based signalling towards the PSTN.

(3) At some time, the call setup reaches a state where an SDP answer can be sent to the IMS endpoint. The SGW creates a key K2 for protecting the media stream from the PSTN MGW to the IMS endpoint.

(4) The SGW commands the PSTN MGW to relay the voice of the call. For the IP based leg, the command comprises the relevant transport addresses and the keys for both directions.

(5) The SGW sends an SDP answer to the IMS endpoint, comprising the transport address at the MGW and K2.

(6) Media can be passed between the IMS endpoint and the PSTN endpoint. SRTP is used on the IP based leg. The PSTN MGW performs decryption for data arriving from the IMS endpoint and encryption for the data sent to the IMS endpoint.

Another scenario, where end-to-middle security may be applied, is in calls where one endpoint does not support media plane security. Such an endpoint will not use SDES in INVITE requests and in answers to such requests. Like in the PSTN gateway scenario, the operator could act on behalf of such an IMS UE and by this provide media plane security at least in the call leg that lies within the operator's network. The operator may also decide to apply media plane security not for the complete call leg, but only between the edge of the IMS core network and the security-capable endpoint. This may be reasonable if the core network already provides media transport that is sufficiently secured. In both cases, a signalling proxy in the operator's network inserts the key management info into the SDP sent towards the security-capable endpoint, and a media proxy in the operator's network performs encryption/decryption based on the keys provided to it from the proxy in the signalling path that controls the operation.

Media plane security applied between the user equipment and the edge of the IMS core network, as mentioned in the previous paragraph, provides in particular protection of IMS media over the access network, cf. also clause 7.2. By the very nature of any such solution, media keys must be available in the SIP proxy controlling the encryption termination point (media proxy) at the edge of the network (cf. SPA and MPA in Figure 14 below). This SIP proxy typically is the first hop SIP proxy, i.e. the P-CSCF (similar to what is shown also in Figure 11). Assuming network domain security between SIP proxy and media proxy, the level of security of the media key management solution is then as high as the level of security of the protocol used for IMS SIP signalling protection between IMS UE and P-CSCF. A high degree of protection can be obtained in this scenario by the use of IMS AKA with IPsec or Digest with TLS (each with encryption enabled), as defined in TS 33.203, as then the protection extends in an uninterrupted fashion between IMS UE and P-CSCF.

NOTE: The use of SDES in this scenario is also possible with other signalling security methods defined in TS 33.203 in an identical way (as the use SDES does not depend on any of these methods), but it should be noted that all these other methods in TS 33.203 make assumptions on the security of the underlying access networks for providing SIP signalling security. These assumptions would then also apply to media security and may even make end media protection unnecessary, e.g. due to the strong link layer security on which GIBA is based.

Figure 14 illustrates the principle of this procedure, for a call from endpoint A supporting SRTP/SDES to an endpoint B that supports only RTP (no SRTP). Note that the picture is an abstraction focussing on the SDP offer and answer, not on the SIP messages. In addition, it does not show the different SIP roundtrips required for the call setup within the IMS.

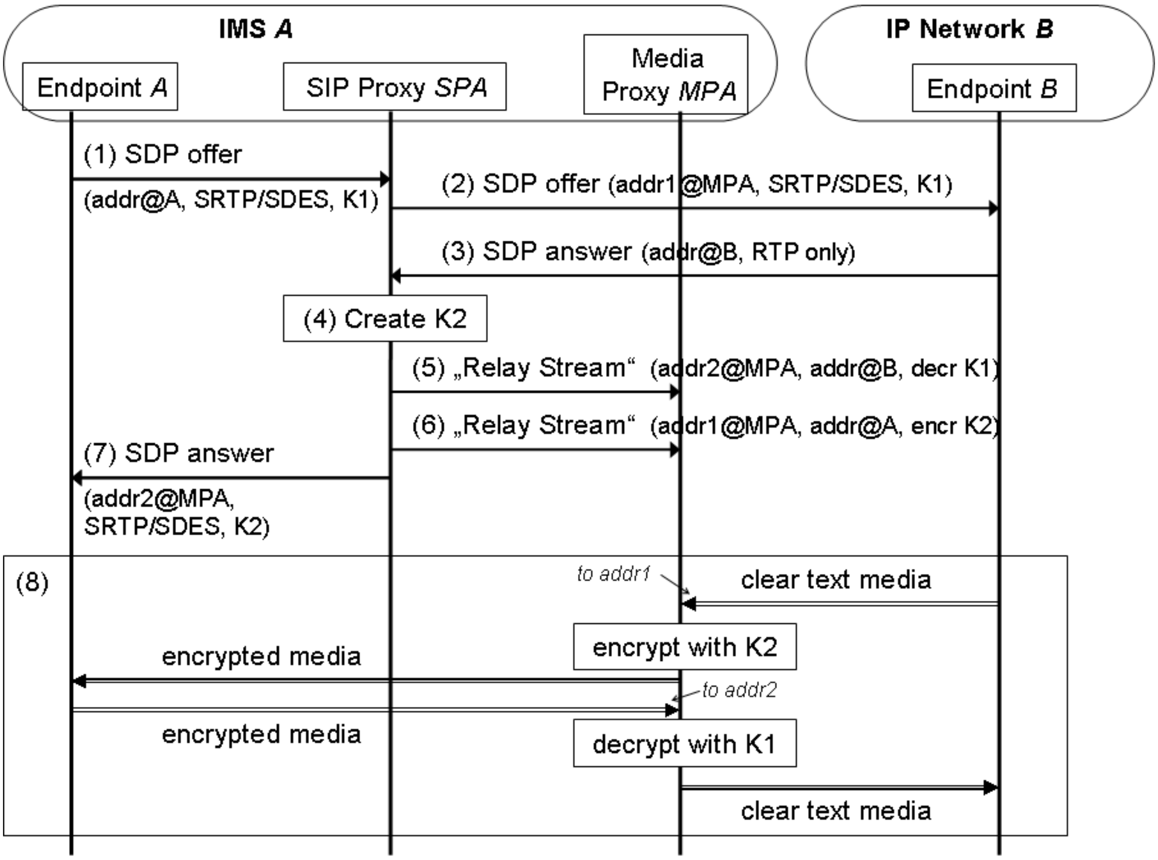

Figure 14: SDES flow where one end-point does not support SRTP/SDES

Description:

(1) An SDP offer for an SRTP stream and a crypto object containing a key K1 arrives from endpoint *A* at the SIP proxy *SPA*. *A* prefers SRTP, but he cannot assume that this is supported by *B* and is willing to communicate even without security, if necessary. So he has included an offer for unencrypted communication, i.e. RTP, too.

(2) To route the offered media stream towards *A* via the media proxy *MPA*, the SIP proxy changes the transport address in the SDP offer from the address at *A* (addr@A) to an address at the media proxy (addr1@MPA) and sends the changed SDP offer towards *B*.

(3) An SDP answer from *B* arrives at the SIP proxy, which analyses it and finds out, that *B* does not support SRTP but can use RTP only.

(4) The SIP proxy creates a key K2 for protecting the media stream towards *A* between the media proxy and *A*.

(5) The SIP proxy commands the media proxy to relay the stream from *A* to *B* and to decrypt the media arriving from *A* using K1. Another address at the media proxy (addr2@MPA) is used for routing the media stream to the media proxy.

(6) The SIP proxy commands the media proxy to relay the stream from *B* to *A* and to encrypt the media arriving from *B* using K2.

(7) The SGW sends an SDP answer to the endpoint, comprising a transport address at the media proxy *MPA* (addr2@MPA), indication of support of SRTP/SDES and the key K2.

(8) Media is exchanged between *A* and *B* via the media proxy, which decrypts media arriving from *A* and encrypts media arriving from B before passing the media on.

Note that in these e2m scenarios endpoint A has only a security association with a proxy or gateway, not an e2e security association as with normal SDES usage according to the RFC 4568. If the subscription only promises to provide e2m protection rather than e2e protection, this is obviously not an issue. Clearly, if the security state of a session is indicated to the user, an indication associated with “e2e security” should not be sent or shown by the IMS UE.

A reasonable approach minimizing user intervention for e2m security while maximizing the usage of protection could be that the IMS UE supporting media security with SDES is configured to, when making calls, always send offers of protected communication. For the purpose of receiving calls, there should be an appropriate node in the terminating network always insert an offer of protected communication, if this is not already part of the original offer from the calling party. The IMS UE may or may not be configured to indicate whether a session is protected (meaning e2m protection).

7.3.6 Possible enhancements to an SDES based solution

As discussed in the previous clauses, there are some issues with plain SDES as described in RFC 4568, in particular related to SIP based call features, i.e. forking/retargeting and early media. Workarounds have been described how to cope with these issues without changing SDES itself. These workarounds may provide acceptable solutions to the discussed issues.

Nevertheless, alternatives are considered how a solution based on SDES might be enhanced in order to provide improved support for SIP based call features. These alternatives may provide more efficient solutions. We provide an outline of possible approaches in the following.

With respect to other requirements, in particular the security requirements, these new approaches are very similar to regular SDES.

7.3.6.1 The SDES crypto object

For the convenience of the reader, this clause shortly describes the usage of the crypto object according to RFC 4568.

SDES introduces a crypto object

a=crypto:<tag> <crypto-suite> <key-params> [<session-params>]

with

key-params = <key-method> “:” <key-info>

and only a single key-method (value “inline”), where a key is contained directly in the crypto object.

If A sends an INVITE to B, the crypto object in the INVITE contains the key for the traffic to be sent from A to B. A can include several crypto objects. B must select one of them (identified by <tag>), but must insert a different key. This key is used for traffic sent from B to A.

7.3.6.2 Advanced support for forking/retargeting

The security issue with forking could be addressed by interpreting the <key-info> of SDES in the SDP offer not as the key, but as a nonce used to derive the key. Together with the <key-info> of SDES in the SDP answer, it could serve as an input to a hash function or pseudo random function that generates the key. By this mechanism, forked endpoints that do not engage in the session will not get the key used for the traffic sent by the SDP offerer. As a trade-off, the offerer could no longer use a common key for different receivers in a multiparty communication.

According to the previous paragraph, the <key-info> of SDES in the SDP answer can be used as an input for a key derivation function to derive the key for securing the traffic sent by the offerer. Obviously, this key derivation function could also generate a (different) key for the traffic in the opposite direction. However, it might be advantageous to use the <key-info> of SDES in the SDP answer directly as the key to protect traffic from the answerer to the offerer, because this supports efficient multicast by the answerer (i.e. using a common key for a stream that is sent to multiple receivers).

Note that the same support for forking/retargeting and multiparty communications can be achieved by deriving both keys solely from the <key-info> provided by the answerer.

7.3.6.3 Support for encrypted early media

It is unclear whether this is a 3GPP requirement at all. If it is, the subsequent considerations may be taken into account.

Support of encrypted early media requires the SDP offerer to somehow provide the key to be used to protect the media sent to it. This would be in accordance to the session description concept applied in SIP, where each endpoint describes the streams it is willing to receive rather than the streams it is going to send. This approach is discussed in an appendix of RFC 4568. A number of problems exist, as well as workarounds for these problems.

A now expired internet draft (draft-wing-mmusic-sdes-early-media-00, available from http://tools.ietf.org/html/), proposes that the offerer provides keys for both directions. Obviously, this makes the forking/retargeting issues even worse, because all endpoints seeing the INVITE get keys for both directions (not only for the traffic sent by the offerer as in regular SDES). This could be mitigated again by specifying that such a SDP answerer provide a new key for the direction answerer to offerer. This new key replaces the key provided by the offerer. With this modification, the approach would not solve the forking/retargeting issue of the regular SDES, but would provide some support for early media without sacrificing the multicast properties of SDES (each sender can choose a common key for the streams it sends to different recipients).

An alternative approach is to let the offerer provide only a (preliminary) key for receiving encrypted early media, and the answerer provide the final keys for both directions. This would be similar to the approach mentioned in the last paragraph of clause 7.3.6.1, with additional support for encrypted early media.

7.3.6.4 Multicast support

SDES according to RFC 4568 currently is restricted to point-to-point unicast communication. As discussed in clause 7.3.6.2, multiparty communication is not excluded by this. If however multicast communication is considered to be a strong requirement, SDES may need to be enhanced to make it more flexible and capable.

For example, the crypto-object may be enhanced to be able to transmit more input values to the SRTP crypto context, e.g. rollover counter and sequence number. Also more session info might be helpful, e.g. SSRC ids.

7.3.6.5 How to indicate new SDES key exchange semantics

An obvious way to indicate usage of one of the proposed new key exchange mechanism is to introduce new key words for <key-method>.

Alternatively, one could replace the current semantics of the <key-info> in SDES even without change of the protocol syntax of SDES, only by using the transmitted <key-info> not necessarily as the sending key, but in the new “modes” described in the previous clauses.

When using the unchanged syntax of SDES with new semantics, this might be indicated by a new field in the SIP header or within the session description. This field must be used by the caller to indicate, which of the modes the caller wants to use.

If a called endpoint supports this mode, it returns the same value, and the semantics of the specified mode are used. If the called endpoint does not support the mode, it does not return the value, which means fallback to regular SDES.

If the called endpoint does not recognize the new field, it will ignore it and will not return it. Again, this results in the usage of regular SDES.

A called endpoint must not use the new field if the calling endpoint didn't use it, and it must not select a different mode.

In this way, interoperation between endpoints that can use the new semantics and endpoints that do no know the new semantics is possible.

7.4 Otway-Rees based key management protocol

7.4.1 Definitions

KMS: Key Management Server.

ID-A: Identity of User A.

ID-B: Identity of User B.

Ka: Shared key between IMS UE-A and KMS.

Kb: Shared key between IMS UE-B and KMS.

Ea (X): X is encrypted with key Ka.

Eb (X): X is encrypted with key Kb.

7.4.2 Solution description

Figure 15 shows a basic idea for IMS media security solution, which is based on the "Otway-Rees" key management protocol.

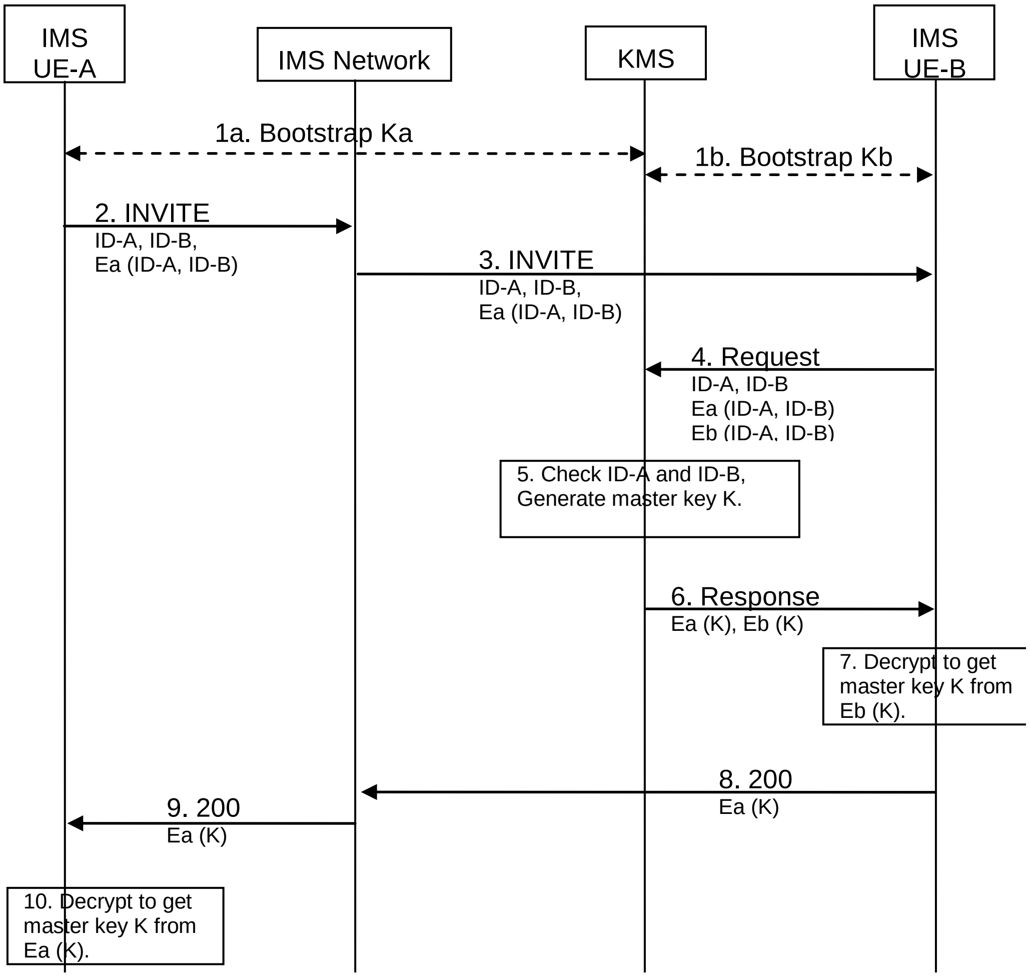

Figure 15: Otway-Rees key management system

1a. IMS UE-A bootstraps with KMS to establish a shared key Ka.

1b. IMS UE-B bootstraps with KMS to establish a shared key Kb.

NOTE: Ka/Kb may be established through GBA mechanism where KMF is a NAF, or other methods.

2. IMS UE-A sends an INVITE message which includes the following parameters: plain identity of user A ID-A, plain identity of user B ID-B and Ea (ID-A, I, ID-B) (encrypted ID-A and ID-B with key Ka,).

3. The INVITE message is sent to IMS UE-B.

4. IMS UE-B sends a request message, which includes the following parameters: plain ID-A and plain ID-B, Ea (ID-A, ID-B) and Eb (ID-A, ID-B) (encrypted ID-A and ID-B with key Kb,), to the KMS to request the master key K for media protection.

5. The KMS uses the plain ID-A and plain ID-B respectively to retrieve the shared key Ka and Kb, then use these keys respectively to get decrypted ID-A and ID-B from Ea (ID-A, I, ID-B) and Eb (ID-A, I, ID-B), and compare the decrypted ID-A and ID-B with plain ID-A and ID-B to make sure they are same. KMS then generates the master key K for media protection.

6. The KMS encrypts the master key K using Ka to get Ea (K), and encrypts the master key K using Kb to get Eb (K), then sends the Ea (K) and Eb (K) to IMS UE-B in the response message.

7. IMS UE-B gets K by decrypt Eb (K) using Kb.

8. IMS UE-B sends the 200 response message which includes the Ea (K).

9. The 200 response message is sent to IMS UE-A.

10. IMS UE-A gets K by decrypt Ea (K) using Ka.

Now IMS UE-A and IMS UE-B share the master K which can be used further to protect the media between them.

7.4.3 Analysis

7.4.3.1 Peer-to-peer

For peer to peer communication, the solution could support end to end media protection.

7.4.3.2 Forking

Forking scenarios include two situations:

1. A call is forked to different terminals belonging to a single user.

2. A call is forked to different users within a group based on the user-specific policy registered in advance.

In forking scenarios, all terminating terminals should use unique keys, and only the terminal to which the call is finally established must get hold of the media encryption key, while other terminals must not be able to obtain the key. The proposed way to generate the media keys in forking scenarios is straightforward. It makes use of a key derivation function modifying a base key generated by the KMS with the help of a random value, which is generated by each terminal on the terminating side.

Editor's note: The case of several terminals using the same USIM is for further study. The same issue should also be studied for TBS.

Figure 16 depicts the procedures for forking key generation. IMS UE-B is assumed to be the terminal that finally answers the call, while IMS UE-C is any other forked terminal.

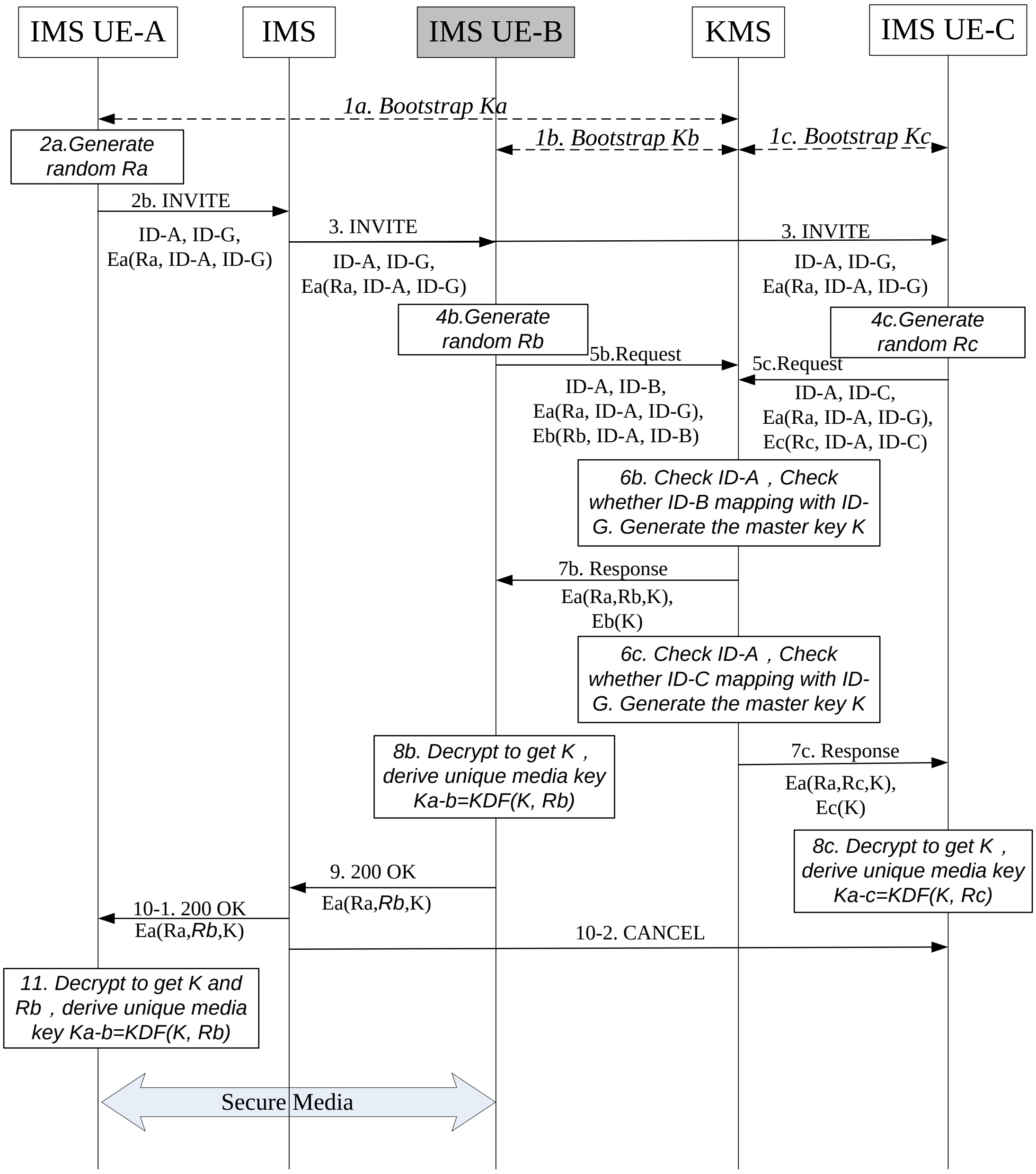
IMS UE-A
IMS
IMS UE-B
KMS
IMS UE-C
1a. Bootstrap Ka
1b. Bootstrap Kb
1c. Bootstrap Kc
2a.Generate random Ra
2b. INVITE
ID-A, ID-G, Ea(Ra, ID-A, ID-G)
3. INVITE
ID-A, ID-G, Ea(Ra, ID-A, ID-G)
3. INVITE
ID-A, ID-G, Ea(Ra, ID-A, ID-G)
4b.Generate random Rb
4c.Generate random Rc
5b.Request
ID-A, ID-B, Ea(Ra, ID-A, ID-G), Eb(Rb, ID-A, ID-B)
5c.Request
ID-A, ID-C, Ea(Ra, ID-A, ID-G), Ec(Rc, ID-A, ID-C)
6b. Check ID-A，Check whether ID-B mapping with ID-G. Generate the master key K
7b. Response
Ea(Ra,Rb,K), Eb(K)
6c. Check ID-A，Check whether ID-C mapping with ID-G. Generate the master key K
8b. Decrypt to get K，derive unique media key Ka-b=KDF(K, Rb)
7c. Response
Ea(Ra,Rc,K), Ec(K)
8c. Decrypt to get K，derive unique media key Ka-c=KDF(K, Rc)
9. 200 OK
Ea(Ra,Rb,K)
10-1. 200 OK
Ea(Ra,Rb,K)
10-2. CANCEL
11. Decrypt to get K and Rb，derive unique media key Ka-b=KDF(K, Rb)
Secure Media

Figure 16: Signalling diagram for parallel forking

ID-G is a collective identity for all forked terminals, IMS UE-B and IMS UE-C. IMS UE-B is the terminal which finally answers the call.

1a. IMS UE-A bootstraps with KMS to establish a shared key Ka.

1b. IMS UE-B bootstraps with KMS to establish a shared key Kb.

1c. IMS UE-C bootstraps with KMS to establish a shared key Kc.

NOTE: Ka/Kb may be established through GBA mechanism where KMS is a NAF or through other methods.

2a. IMS UE-A generates a random Ra.

2b. IMS UE-A sends an INVITE message which includes the following parameters: plain identity of IMS UE-A ID-A, plain identity of ID-G and Ea(Ra, ID-A, ID-G)(encrypted Ra, ID-A and ID-G with key Ka).

3. IMS network forwards INVITE message to corresponding IMS UEs, i.e. IMS UE-B and IMS UE-C in parallel or in sequence based on the user-specific forking policy. Note that in Figure 16, parallel forking is showed. For sequential forking, only the step sequence is changed, but no main difference occurs.

After receiving the INVITE message, each IMS UE on the terminating side will carry out the following procedure. Each IMS UE, which receives an INVITE, generates a random number and sends a request message to the KMS to ask for the media master key. After receiving the media master key from the KMS, the IMS UE derives a unique media key by performing a key derivation function on the master key K and the random number generated by the IMS UE, i.e.

For IMS UE-B shown in Figure 16:

4b. IMS UE-B generates Rb.

5b. IMS UE-B sends a request message including plain ID-A, ID-B, Ea(Ra, ID-A, ID-G) and Eb(Rb, ID-A, ID-B)(encrypted Rb, ID-A and ID-B with key Kb).

6b. KMS uses the plain ID-A and plain ID-B respectively to retrieve the shared key Ka and Kb, then use these keys respectively to get decrypted ID-A and ID-G from Ea(Ra, ID-A, ID-G), ID-A and ID-B from Eb(Rb, ID-A, ID-B), and KMS compares the decrypted ID-A and ID-G with plain ID-A, ID-B to ensure that they have the same ID-A, and ID-B matches the ID-G. Assuming that, IMS UE-B is the first one that contacts KMS to ask for a master key. Then KMS generates the master key K.

Note that, for the same originating IMS UE, KMS will generate only one master key for all receiving IMS UEs.

7b. KMS encrypts random numbers Ra, Rb and the master key K using Ka to get Ea(Ra, Rb, K), and encrypts the master key K using Kb to get Eb(K). Then KMS sends the Ea(Ra, Rb, K) and Eb(K) to IMS UE-B in the response messages.

8b. IMS UE-B gets K by respectively decrypting Eb(K) using Kb, then derives the unique media key by performing a key derivation function on the master key K and the random number that is generated by each terminal, i.e. Ka-b= KDF(K, Rb).

For IMS UE-C shown in Figure 16:

4c. IMS UE-C generates Rc.

5c. IMS UE-C sends a request message including plain ID-A, ID-C, Ea(Ra, ID-A, ID-G) and Ec(Rc, ID-A, ID-C)(encrypted Rc, ID-A and ID-C with key Kc).

6c. KMS uses the plain ID-A and plain ID-C respectively to retrieve the shared key Ka and Kc, then use these keys respectively to get decrypted ID-A and ID-G from Ea(Ra, ID-A, ID-G) and ID-A and ID-C from Ec(Rc, ID-A, ID-C), and KMS compares the decrypted ID-A and ID-G with plain ID-A, ID-C to ensure that they have the same ID-A, and ID-C matches the ID-G. If KMS has generated a master key K for the same initiator, it retrieves the master key K; otherwise, KMS generates the master key K.

7c. KMS encrypts random number Ra, Rc and the master key K using Ka to get Ea(Ra, Rc, K), and encrypts the master key K using Kc to get Ec(K). Then KMS sends Ea(Ra, Rc, K) and Ec(K) to the IMS UE-C in the response messages.

8c. IMS UE-C get K by respectively decrypting Ec(K) using Kc then derives the unique media key by performing a key derivation function on the master key K and the random number that is generated by each terminal, i.e. Ka-c=KDF(K, Rc).

NOTE: The messages from each IMS UE towards the KMS are independent of other IMS UEs, which means the messages towards the KMS from different IMS UEs are not correlated.

Editor's note: It has to be clarified whether the master key is the same in 6b and 6c and if this is the case how it is ensured that user C and user B do not find out each others session key

9. IMS UE-B sends the 200OK response message which includes the Ea(Ra, Rb, K) to IMS network.

10-1. IMS network forwards the 200OK response message to IMS UE-A.

10-1. As the call is established between IMS UE-A and IMS UE-B, IMS network sends CANCEL message to other terminals, i.e. IMS UE-C in Figure 1.

11. IMS UE-A gets K and Rb by decrypting Ea(Ra, Rb, K) using Ka, it then derives the media key using KDF from the master key K and the random number Rb, i.e. Ka-b=KDF(K,Rb).

Now IMS UE-A and IMS UE-B share the same unique media key Ka-b that is used to protect the media between them.

7.4.3.3 Deferred delivery

Figure 17 shows the procedure for the deferred delivery solution. For the sake of simplicity the random number used in each message to prevent replay attack is omitted in the message diagram.

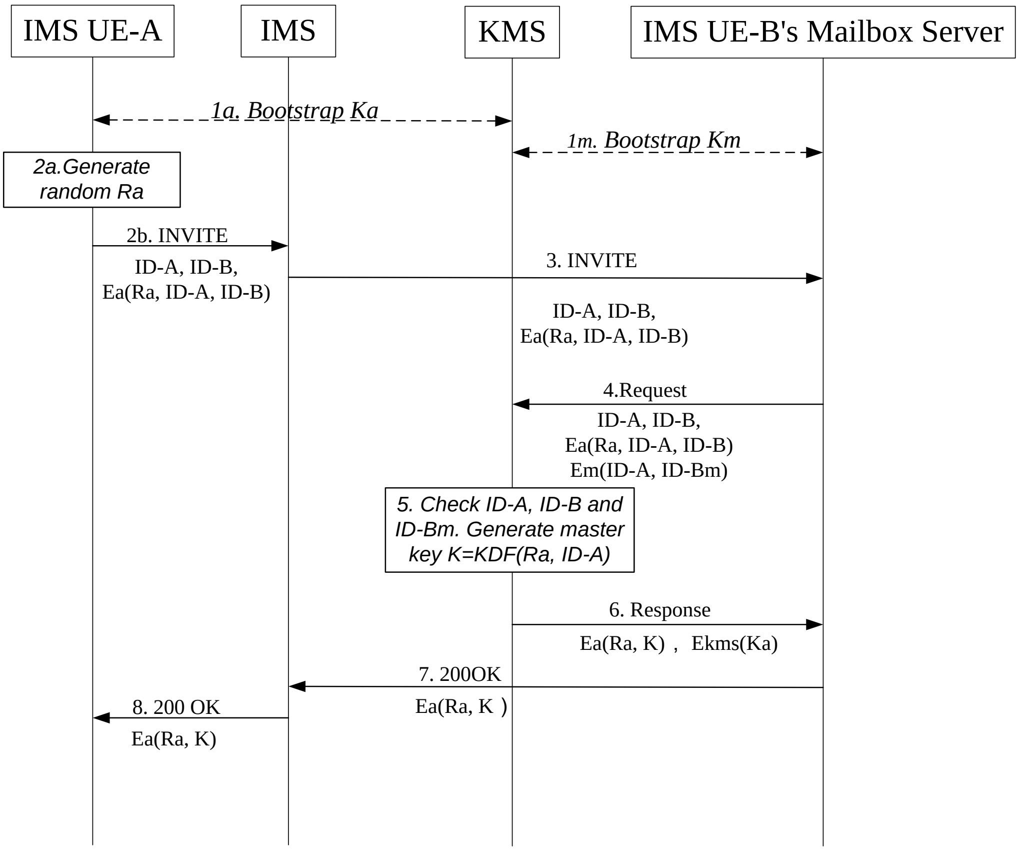
IMS UE-A
IMS
KMS
IMS UE-B's Mailbox Server
1a. Bootstrap Ka
1m. Bootstrap Km
2a.Generate random Ra
2b. INVITE
ID-A, ID-B, Ea(Ra, ID-A, ID-B)
3. INVITE
ID-A, ID-B, Ea(Ra, ID-A, ID-B)
4.Request
ID-A, ID-B, Ea(Ra, ID-A, ID-B) Em(ID-A, ID-Bm)
5. Check ID-A, ID-B and ID-Bm. Generate master key K=KDF(Ra, ID-A)
6. Response
Ea(Ra, K) , Ekms(Ka)
7. 200OK
Ea(Ra, K)
8. 200 OK
Ea(Ra, K)

Figure 17: Signalling diagram for deferred delivery

1a. User A bootstraps with KMS to establish a shared key Ka. If GBA is not support, User A can use other authentication method to get shared key Ka

1m. Mailbox AS can establish a secure connection with KMS by Ipsec, TLS, GBA, etc by which KMS and mailbox AS can have a shared secret Km.

2a. IMS UE-A generates a random number Ra.2b. IMS UE A sends an INVITE message which includes the following parameters: plain identity of user A, ID-A, plain identity of user B, ID-B and Ea (Ra, ID-A, ID-B) (encrypted Ra, ID-A, ID-B with key Ka) to IMS network.

3. IMS network forward INVITE message to User B's mailboxserver.

4. Once receive the INVITE message from user A, User B's mailbox server sends a request message, which includes the following parameters: plain ID-A, plain ID-B, Ea (Ra, ID-A, ID-B) and Em(ID-A,ID-Bm) to the KMS to request media key K, where ID-Bm is the identity of user B's mailbox.

5. KMS uses Ka, Km to get decrypted Ra, ID-A, ID-B from Ea (Ra, ID-A, ID-B) and ID-A, ID-Bm from Em(ID-A,ID-Bm), compares the decrypted ID-A with plain ID-A to make sure they are same, and check whether ID-Bm is IMS UE-B's mailbox's identity. Then KMS generates the media key K by performing a key derivation function based on the random number Ra and ID-A. KMS encrypts the random Ra and the media key K using Ka to get the Ea(Ra, K), and as ID-Bm indicates IMS UE-B's mailbox, KMS encrypts the shared key Ka using its own key Kkms to get Ekms(Ka).

6. KMS sends Ea(Ra, K) and Ekms(Ka) to IMS UE-B's mailbox server in the response message.

7. IMS UE-B's mailbox server sends the Ea(Ra, K) to IMS network in the response message to IMS UE-A.

8. IMS network forward the response message to IMS UE-A.

After the above-mentioned procedure, IMS UE-A gets the media key K while IMS UE-B's mailbox server has no way to get the media key K. So the IMS UE-B's mailbox server cannot access the plaintext of the media.

Once the IMS UE-B is online, it first gets security parameters including ID-A, ID-B, Ea(Ra, ID-A, ID-B) from its mailbox server. From those security parameters, IMS UE-B gets to know there is a deferred media for it from user A, it will send a request including the parameters that it gets from the mailbox to KMS to ask for the media key. The message diagram in Figure 18 shows the procedure where IMS UE-B fetches the media key from KMS.

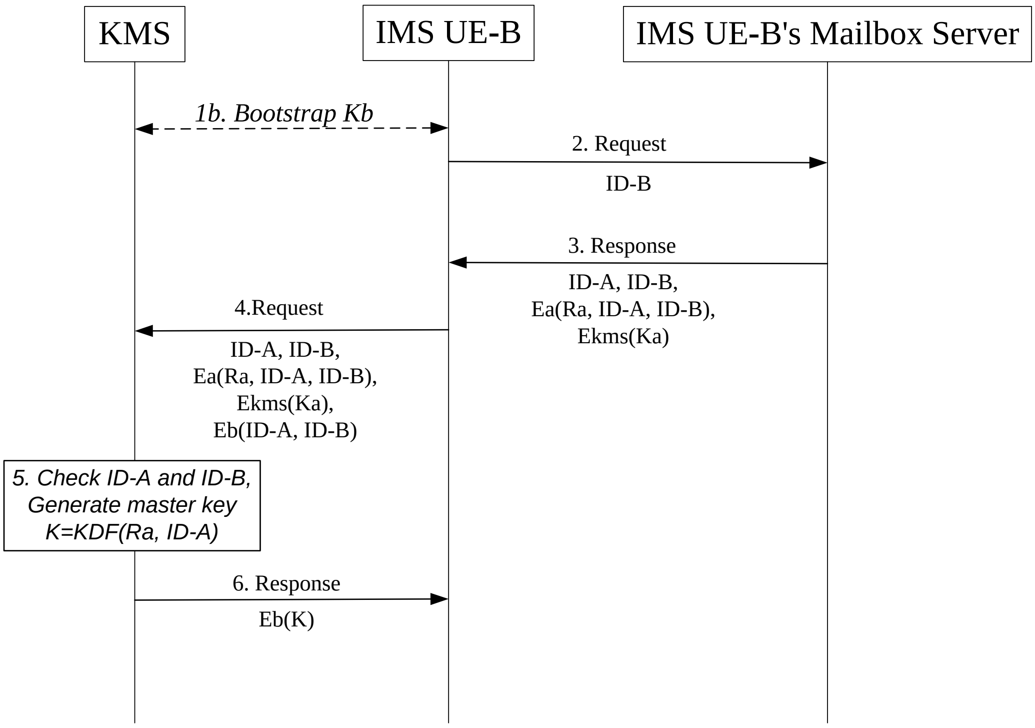

Figure 18: Message flow when IMS UE-B is online

1b. IMS UE-B bootstraps with KMS to establish a shared key Kb. If GBA is not support, IMS UE-B can use other authentication method to get shared key Kb

2 IMS UE-B sends a request including its identity ID-B to its mailbox server.

3 IMS UE-B's mailbox server response to user B with the messing including following security parameters: ID-A, ID-B, Ea(Ra, ID-A,ID-B) , Ekms(Ka).

4 IMS UE-B sends a request message to KMS including ID-A, ID-B, Ea(Ra, ID-A,ID-B), Ekms(Ka), Eb(ID-A,ID-B).

5 KMS using Ka by decrypting Ekms(Ka) using Kkms. Then KMS uses Ka, Kb to decrypt the Ea(ID-A,ID-B) and Eb(ID-A,ID-B) and compares the decrypted ID-A, ID-B with plain ID-A, ID-B to make sure they are the same. Then KMS derives the media key K based on Ra and ID-A.

6 KMS encrypts the K using Kb to get the Eb(K), and then sends the Eb(K) to IMS UE-B in the response message.

After the procedure IMS UE-B also gets the media key K, then IMS UE-B can retrieve the deferred media from mailbox server.

NOTE: To prevent mailbox from DoS or MitM attack, the procedure illustrated in figure 17 can also be improved in the way that a shared key (Ke2m) between IMS UE-A and the mailbox server can be derived to ensure the mutual authentication and integration protection between IMS UE-A and the mailbox server in case there is no existing mutual authentication and integration protection mechanism in place. See figure 19.

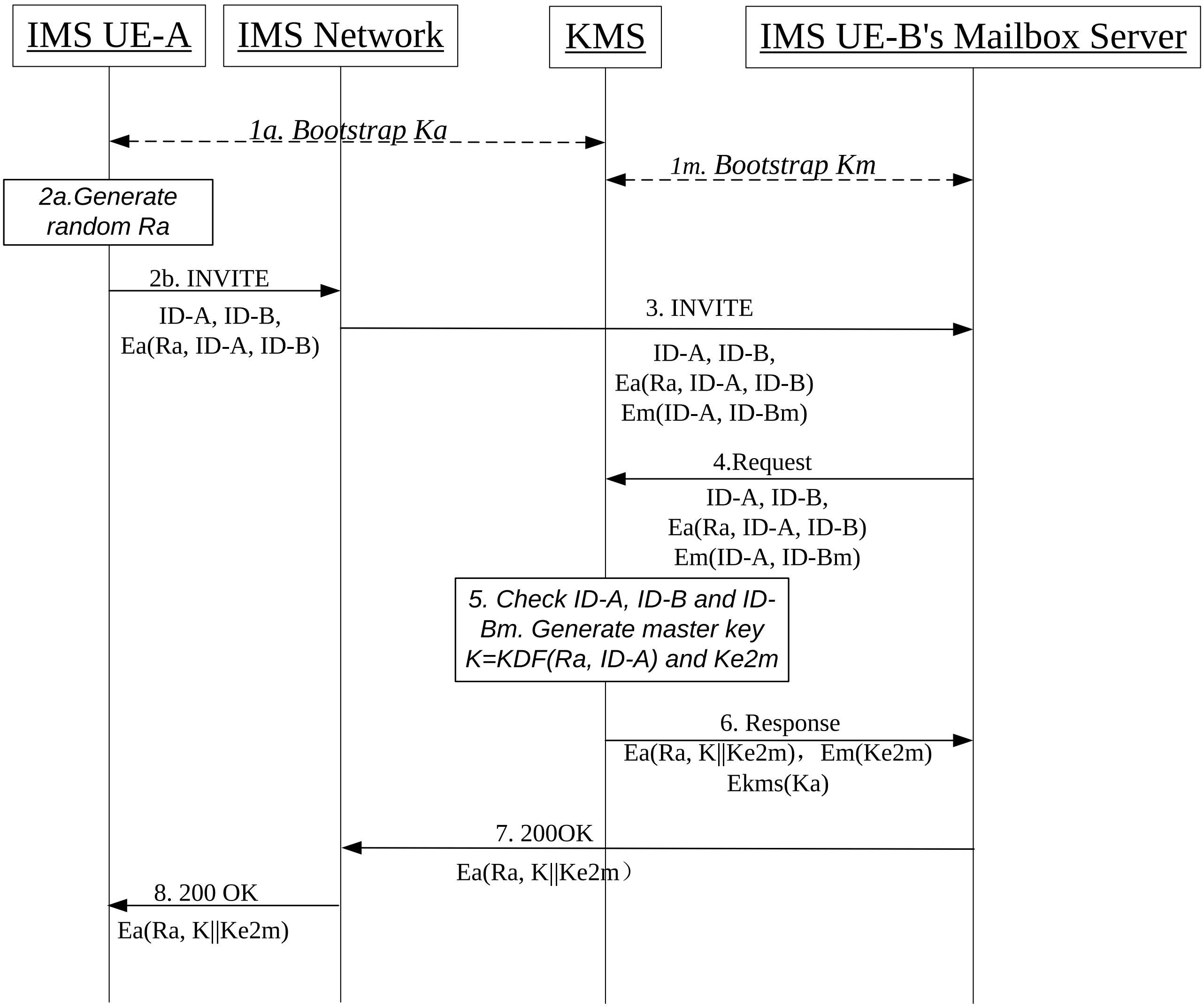
IMS UE-A
IMS Network
KMS
IMS UE-B's Mailbox Server
1a. Bootstrap Ka
1m. Bootstrap Km
2a.Generate random Ra
2b. INVITE
ID-A, ID-B, Ea(Ra, ID-A, ID-B)
3. INVITE
ID-A, ID-B, Ea(Ra, ID-A, ID-B) Em(ID-A, ID-Bm)
4.Request
ID-A, ID-B, Ea(Ra, ID-A, ID-B) Em(ID-A, ID-Bm)
5. Check ID-A, ID-B and ID-Bm. Generate master key K=KDF(Ra, ID-A) and Ke2m
6. Response
Ea(Ra, K||Ke2m)，Em(Ke2m) Ekms(Ka)
7. 200OK
Ea(Ra, K||Ke2m）
8. 200 OK
Ea(Ra, K||Ke2m)

Figure 19: Signalling diagram for deferred delivery with e2m protection

As shown in Figure 19, KMS generates Ke2m and K at the same time. Then KMS deliveries the Ke2m protected by Ka and Km respectively to ensure only user B’s mailbox server and IMS UE-A can get the Ke2m. With the shared key Ke2m, mailbox server and IMS UE-A can use Ke2m to do mutually authentication and transport protection.

7.4.3.4 Transcoders

For the network functions operating on plaintext media, e.g. transcoders, the KMS could deliver the master key K to the network functions after successful authorization. So this solution could support this use case. The detail is FFS.

7.4.3.5 Group and conference calls

Figure 20 shows the procedure for the conference calling. For the sake of simplicity the random number used in each message to prevent replay attack is omitted in the message diagram.

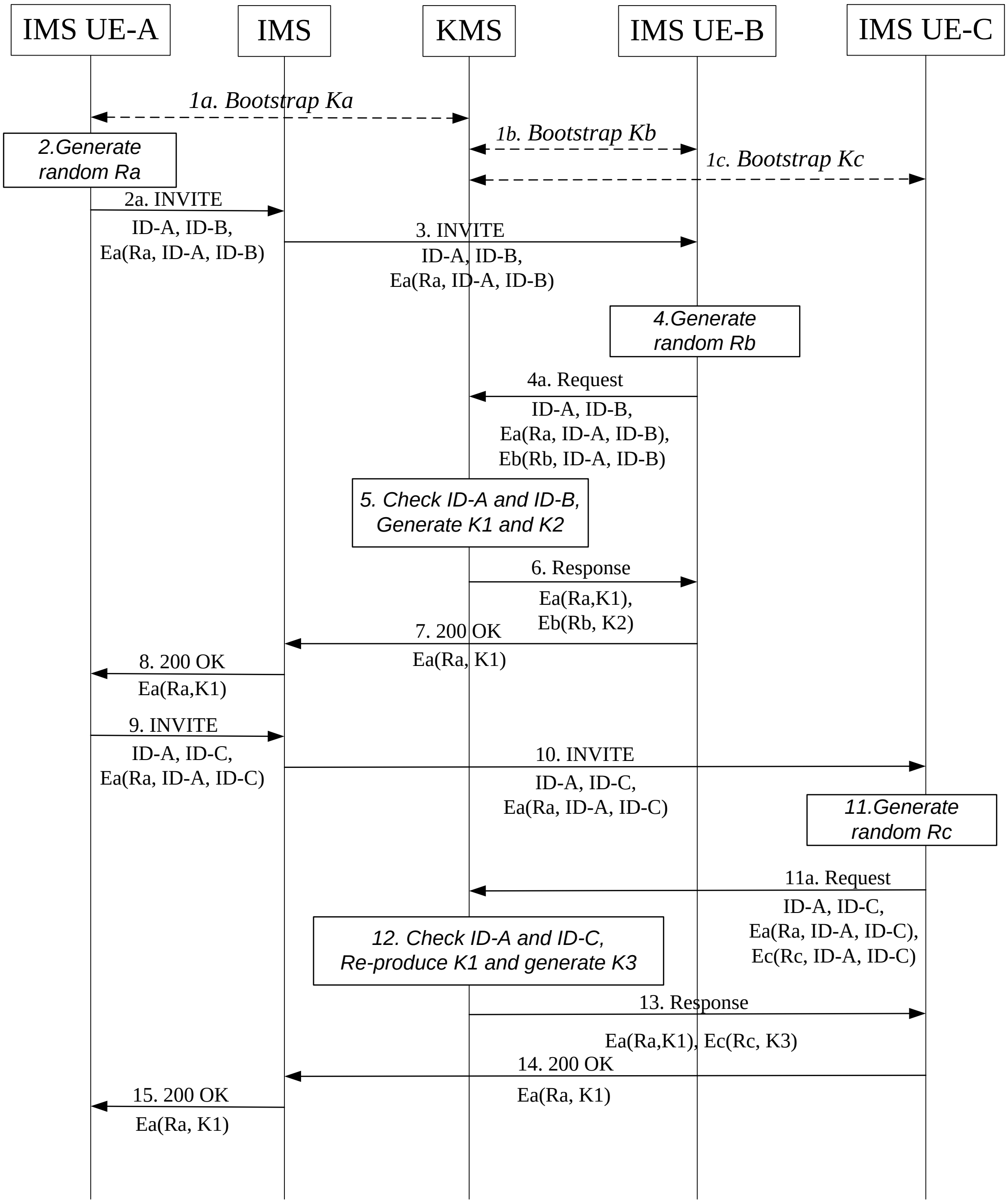

Figure 20: Message flow for group and conference calls

1a. IMS UE-A bootstraps with KMS to establish a shared key Ka.

1b. IMS UE-B bootstraps with KMS to establish a shared key Kb.

1c. IMS UE-C bootstraps with KMS to establish a shared key Kc.

Note: Ka/Kb/Kc may be established through GBA mechanism where KMS is a NAF, or other methods.

2. IMS UE-A generates a random Ra.

2a. IMS UE-A sends an INVITE message which includes the following parameters: plain identity of user A ID-A, plain identity of user B ID-B and Ea (Ra, ID-A, ID-B) (encrypted Ra, ID-A, ID-B with key Ka).

3. IMS Network forwards the invite message to IMS UE-B.

4. IMS UE-B generates a random Rb.

4a. IMS UE-B sends a request message, which includes the following parameters: plain ID-A，plain ID-B and Ea (Ra, ID-A, ID-B) and Eb(Rb, ID-A,ID-B) to the KMS to request the media key.

5. The KMS use plain ID-A, plain ID-B to get Ka, Kb, and use Ka, Kb to get decrypted ID-A, ID-B from Ea (Ra, ID-A, ID-B) and Eb(Rb, ID-A,ID-B), and compare the decrypted ID-A, ID-B with plain ID-A , ID-B to make sure they are same. KMS then generates the key K1 for media protection between UE-A and Conference Bridge and the key K2 for media protection between IMS UE-B and Conference Bridge.

6. KMS use Ka to encrypt the Ra and K1 to get Ea(Ra, K1) and use Kb to encrypt the Rb and K2 to get Eb(Rb, K2), then send Ea(Ra, K1) and Eb(Rb, K2) to IMS UE-B in response message.

7. IMS UE-B get K2 by decrypting Eb(Rb, K2) using Kb, then send Ea(Ra, K1) to IMS network in response message.

8. IMS network forward the Ea (Ra, K1) to IMS UE-A in response message. Thus IMS UE-A can get the K1 by decrypting Ea(Ra, K1).

9. IMS UE-A sends an INVITE message which includes the following parameters: plain identity of user A ID-A, plain identity of user C ID-C and Ea(Ra, ID-A,ID-C). (Encrypted Ra, ID-A, ID-C with key Ka).

NOTE: Step 9 can be implemented concurrently with step 2.

Editor's Note: In the case that step 9 and step 2 happen concurrently, it needs to be described how the KMC can handle this situation.

Technically step 2 to 8 can be implemented concurrently or in sequence with step 9 to 15. It depends on different user practice. The chairman of conference call may call each participant one by one or call each participant all at once. The step 2 to 8 is independent of the step 9 to 15 with only one correlation that the Ra used in step 2 to 8 and step 9 to 15 is same. So KMS can use Ra as input parameter to generate the K1 to ensure IMS UE-A always gets the same K1 after calling a new percipient.

10. IMS Network forwards the INVITE message to IMS UE-C.

11. IMS UE-C generates random number Rc.

11a. IMS UE-C sends a request message, which includes the following parameters: plain ID-A，plain ID-C and Ea (Ra, ID-A, ID-C) and Ec(Rc, ID-A,ID-C) to the KMS to request the media key.

12. The KMS use plain ID-A, plain ID-C to get Ka, Kc and use Ka, Kc to get decrypted ID-A, ID-C from Ea (Ra, ID-A, ID-C) and Ec(Rc, ID-A,ID-C), and compare the decrypted ID-A, ID-C with plain ID-A , ID-C to make sure that they are same. KMS then re-produces the key K1 for media protection between UE-A and Conference Bridge and generates the key K3 for media protection between UE-C and Conference Bridge.

13. KMS use Ka to encrypte the Ra, K1 to get Ea(Ra, K1) and use Kc to encrypt the Rc, K3 to get Ec(Rc, K3), then send Ea(Ra, K1) and Ec(Rc, K3) to UE-C in response message.

14. IMS UE-C get K3 by decrypting the Ec(Rc, K3) using Kc, and then send Ea(Ra, K1) to IMS network in response message.

15. MS network forward the Ea(Ra, K1) to IMS UE-A.

After the above-mentioned steps, the IMS UE-A, IMS UE-B and IMS UE-C get the media key K1, K2 and K3 with KMS respectively.

Figure 21 shows that the Conference Bridge communicates with KMS in a secure way to obtain K1, K2 and K3 from the KMS. The conference bridge then has the shared secrets it needs to communicate securely with IMS UE-A, IMS UE-B and IMS UE-C respectively.

Editor's Note: It needs to be clarified how the Conference Bridge can communicate securely with the KMS, e.g. using NDS/IP.

Editor's Note: It needs to be clarified how keys K1, K2 and K3 are generated.

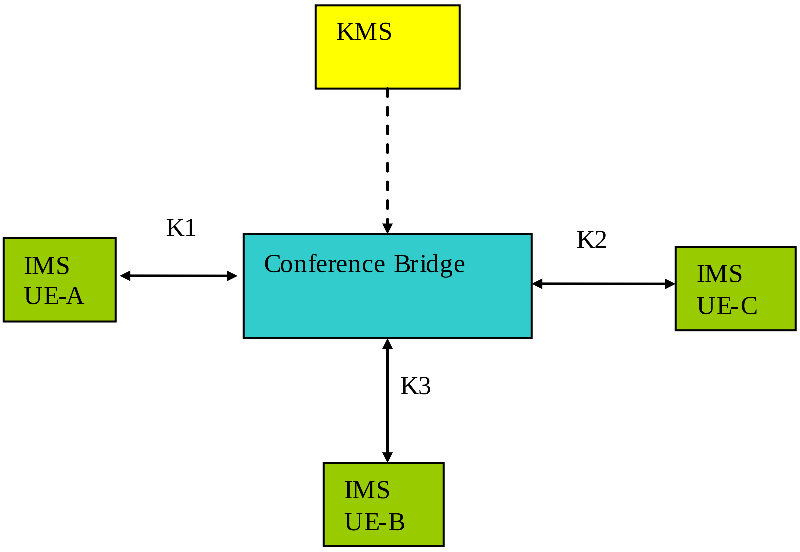

Figure 21: Key distribution for conference calling

7.4.3.6 End-to-middle

Figure 22 shows a typical end-to-middle scenario where a call takes place between an IMS end user and a PSTN end user. This solution does not apply to the case where the protection is required between the end user and the access edge. For the sake of simplicity the random number used in each message to prevent replay attack is omitted in the message diagram.

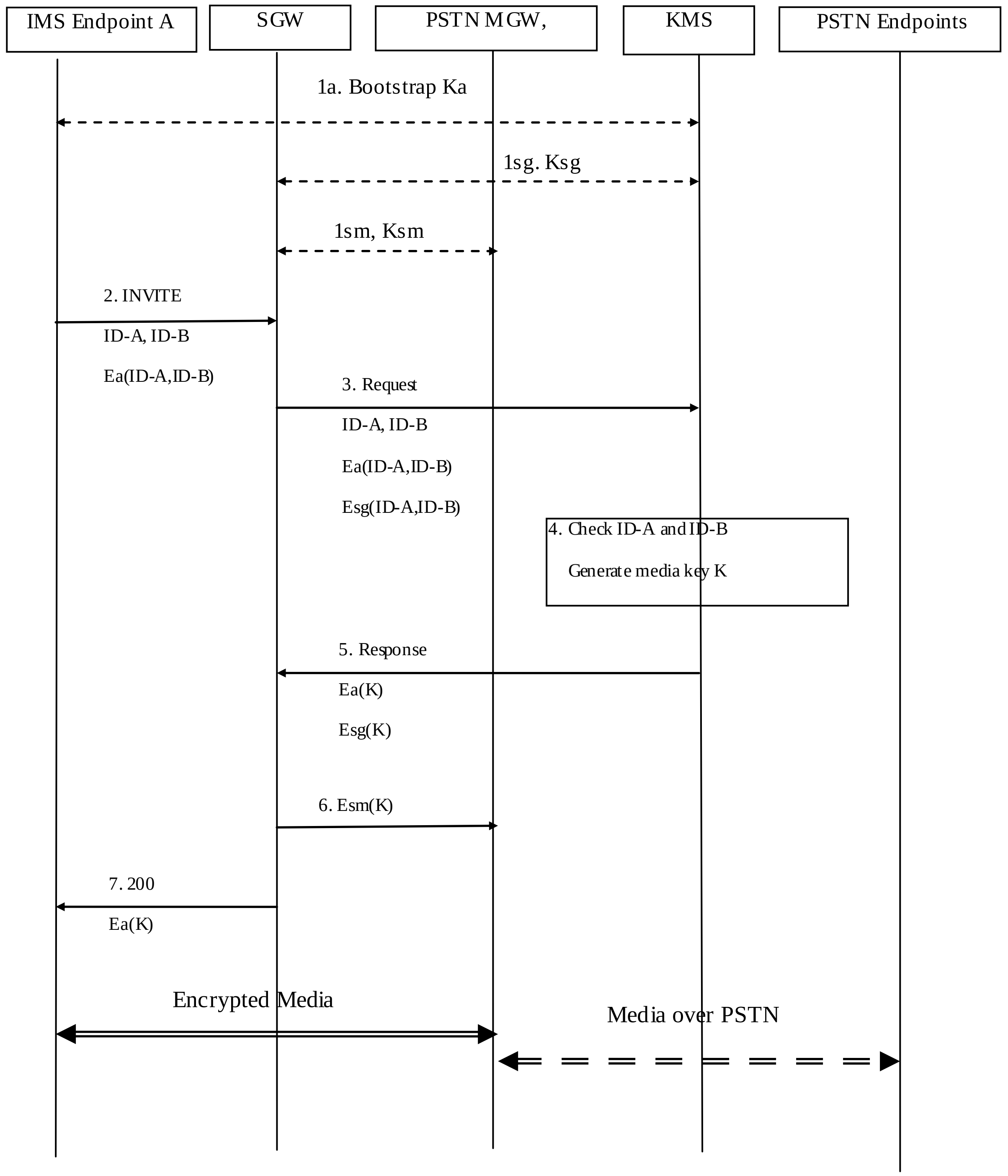

Figure 22: Message flow for end-to-middle scenario

1a. User A bootstraps with KMS to establish a shared key Ka. If GBA is not support, User A can use other authentication method to get shared key Ka

1sg. Secure Gateway(SGW) establish a secure connection with KMS by Ipsec, TLS or any other authentication method by which KMS and SGW can have a shared secret Ksg.

1sm. SGW establish a secure connection with PSTN media gateway (MGW) by Ipsec, TLS or any other authentication method to get shared key Ksm

2. User A sends an INVITE message which includes the following parameters: plain identity of user A ID-A, plain identity of user B ID-B and Ea (ID-A, ID-B) (encrypted ID-A, ID-B with key Ka) to SGW

3. Once receive the INVITE message from user A, SGW sends a request message, which includes the following parameters: plain ID-A, plain ID-B, Ea (ID-A, ID-B) and Esg(ID-A,ID-B) to the KMS to request media key K.

4. KMS use Ka, Ksg to get decrypted ID-A, ID-B from Ea (ID-A, ID-B) and Esg(ID-A,ID-B), and compare the decrypted ID-A, ID-B with plain ID-A , ID-B to make sure they are same. Then KMS generate the media key K.

5. KMS encrypts the K using Ka, Ksg to get the Ea(K), Esg(K) respectively, and then sends the Ea(K), Esg(K) to SGW in the response message.

6. SGW get media key K by decrypting the Esg(K). Then SGW send media key K to PSTN MGW using the security link between SGW and PSTN MGW protected by Ksm.

7. SGW send media Ea(K) to user A.

NOTE: Step 6 and step 7 can be executed concurrently.

After the above-mentioned procedure, the IMS endpoint user has a security association with PSTN MGW.

7.4.4 Lawful intercept

As a KMS-based solution, Otway-Rees has a lot of similarities with the TBS solution with respect to lawful interception. But in the Otway-Rees solution, the KMS generates the master media key and distributes the master media key to the users. The master media key for an on-going call should be stored in the KMS. Once the call is completed, the master media key can be purged out of the KMS.

Editor's Note: It is ffs on how KMS is notified when a call is completed.

To be able to provide a clear copy of intercepted communication, the following conditions have to be fulfilled:

1. It must be possible to intercept the traffic (both signalling and media).

2. It must be possible to intercept the identities of calling parties from signalling.

3. Based on the intercepted identities, KMS can provide the master media key.

If signalling messages can be intercepted in the signalling plane to determjne the identities of calling parties, LI can be done by retrieving the master media key from the KMS based on the identities of calling parties. In roaming situations, if the P-CSCF is located in the home network rather than visit network, the SIP message is always in encrypted format at bearer level in the visited network, in which case the SIP message may be transferred in plaintext. With the P-CSCF in the visited network, intercept of signalling would always be possible while also in this case there has to be an interoperation agreement between the visited network and the entity handling KMS functionality/services.

The master media key is produced in the KMS. So the master media key should be retrieved from KMS as part of LI functions. If the terminal is roaming, LI may involve contacting a KMS in the visited network or a KMS in the home network.

From an LI point of view, the visited network should have the capability to interception all calls within its own network. But in this case, the visited KMS needs to communicate with user's home BSF to authenticate the user, which can cause a lot of complexity. Thus the simplest way is that the roaming terminal still uses its home KMS. In case the visited network wants to intercept the call, it should have an agreement with the home network to transfer the master media key.

The LI should intercept the signaling messages between two parties to know that a call happens between two particular persons based on the SIP URI or TEL URI. It is assumed that SIP URI or TEL URI won't be changed for bypassing LI purpose because the SIP message otherwise won't be routed correctly. If the SIP URI matches the target identify, LI can use two methods to know which KMS it should go to in order to retrieve the master media key. The first method is that the user terminal always uses its home KMS and the URI of the user binds with the URI of the KMS by a natural way, such as if the SIP URI is username@abc.com, then its KMS URI is: kms@abc.com. The second method is more dynamic; that either terminal transfers its current KMS identity to the core network, or core network allocate a KMS for a user and then send the KMS identity to the user. This method is more flexible in that one user can use multiple KMSs. In this solution, the LI function can grab the user identity from signalling. All necessary key material or credentials for LI comes from KMS rather than signalling because the LI may otherwise grab the corrupted key material or credentials from signalling since the KMS-based solution has no security requirement on signalling. But with minor modification, this solution also supports the scenario where the LI picks the key material or credentials from signalling and then submits it to the KMS for resolving to get the master key.

7.5 DTLS-SRTP

7.5.1 Brief Description of DTLS-SRTP

DTLS-SRTP, described currently in two IETF Internet-Drafts ([11] and [12]), uses the handshake protocol of DTLS (RFC 4347, [10]) to establish keying material, algorithms, and parameters for SRTP. The handshake is performed in the media path, using UDP between those transport addresses (transport address = IP address + port) that are also used by the RTP media streams to be secured. DTLS-SRTP is specified for point-to-point sessions with two participants.

DTLS ([10]) requires that peers can be mutually authenticated, preferably by presenting certificates signed by a certificate authority (CA) that is trusted by both peers. (Other peer authentication methods like relying on a pre-shared key are also specified.) The goal of DTLS-SRTP is however to allow secure communication between parties that do not know each other before and that do not share a common trusted CA. To achieve this, DTLS-SRTP uses peer authentication methods where each peer is authenticated via a certificate that is not signed by a CA, but only by the peer itself. The identity of the peers cannot be asserted by such certificates, but is asserted via the SIP signalling used to establish the media session, e.g. by the usage of the P-asserted-identity header field or by SIP identity and SIP connected identity (RFC 4474 and RFC 4916, [13] and [14]).

To ensure that an attacker in the media path cannot perform a man-in-the-middle attack on the certificates, certificate fingerprints are transmitted in the SIP messages (inside the SDP bodies) that allow verifying the validity of a certificate received over the media path. The integrity of the fingerprint must be protected, e.g. by general measures to protect the signalling traffic, or by the usage of SIP identity and SIP connected identity. (Additional variants have been proposed in different (personal) Internet Drafts.)

The following sections discuss the most important issues for DTLS-SRTP as a solution candidate for the IMS media plane security.

7.5.2 Usage of the media path

According to requirement 22, a solution must not rely on the media path being available before session establishment, because it is assumed that session border controllers may be present that block the media path until the session has been established.

While DTLS-SRTP allows that the called party uses the media path to perform the DTLS-SRTP handshake immediately after it has received the SDP offer, it is also possible for the called party to stay passive and let the caller start the handshake.

Depending on the policies in the network, the media path may be available after the SDP answer has been transmitted, or – at the latest – after the 200 OK message has been passed. In both cases, the calling party is the receiver of the message that opens up the media path and can therefore start the DTLS-SRTP handshake immediately after receiving the message.

According to requirement 23, a media security solution shall assume that only media traffic can be sent over the media path. In case of DTLS-SRTP, the handshake packets, which are not media packets, must be transmitted in the media path. These packets use exactly the same IP addresses and UDP ports as the subsequent media packets. There seems not much point in blocking such traffic. Moreover, there is reason to assume that deployed SBC-products feature a considerable degree of flexibility and are not limited in a way that they cannot be configured to let the handshake traffic pass.

Concerning requirement 29: Obviously, a slight delay arises because the handshake cannot be done before the media path is available. However, the problem of clipping is rather caused by the policy enforced by the SBCs (block the path until session establishment) than by the fact that the DTLS-SRTP handshake is done in the media path.

7.5.3 Lawful interception

Requirements 1-3 require the support of LI. Three approaches to perform LI for DTLS-SRTP are outlined in the following sections. None of them is as easy and straightforward as it would be e.g. for SDES. Of the three approaches below, only “Key disclosure” seems to be feasible.

Note that on the other hand, it is currently not fully clear, to what degree an operator will be obliged to provide cleartext communication content, if the operator does not contribute to the encryption and does not know the keys (as it is the case for DTLS-SRTP).

7.5.3.1 Lawful MitM attack

At its current state, LI for DTLS-SRTP would require a man-in-the-middle "attack" (it would be a "lawful attack") in both the media and the signalling path to allow interception. This "attack" could not be detected by the end user by applying the means available through the DTLS-SRTP mechanism, e.g. by comparing the certificate fingerprint from the signalling messages with the certificate used during the DTLS handshake. (End users could however agree on additional means allowing them to find out that there is a man-in-the-middle, e.g. transmitting the certificate fingerprints again by spoken voice and comparing them with the ones received during the DTLS-SRTP handshake. It is assumed that it would not be feasible for the operator to prevent such methods.)

This method obviously requires considerable effort for LI, and it is doubtful whether it is feasible.

7.5.3.2 Protocol-based hidden key recovery

The principles of such an approach are described in [15]. The idea is to use protocol fields that carry a random or an unspecified value to transport secret information (like e.g. a session key) to a party (the Law Enforcement Agency) that eavesdrops the communication and is informed about this kind of secret information disclosure. A prerequisite is, that the protocol implementation (on the user equipment) must include this „disclosure function", i.e. it must be compromised (from the point of view of its unknowing user).

An example would be the following: A client TLS implementation that performs RSA key exchange uses the 28 Byte nonce in the client hello to transport a value that can be used by the eavesdropping LEA to compute the pre-master secret (and by this the session keys).

One problem with this approach is, that suitable protocol fields are not always available – e.g. in TLS, the available fields are too short. Workarounds for this are available, but they require that secret information is disclosed during several consecutive sessions. The LEA must not miss one of these, and can only decrypt the sessions that are established after all necessary information has been disclosed (i.e. it cannot decrypt the first few sessions).

There are more problems, e.g. it seems hard to ensure that users do not use other, non-compromised protocol implementations. When protocols change (e.g. improved, new versions), the method may have to be adapted or may even become unfeasible.

Because of these weaknesses, protocol-based hidden key recovery is not considered to be a sound basis for LI.

7.5.3.3 Key disclosure

The Internet-Draft draft-wing-sipping-srtp-key-04 (formerly entitled "SRTP Key Disclosure") ([16]) proposes that after the key exchange, user agents send SRTP keys to trusted nodes in the network. This is proposed in order to support scenarios, where the network has to decrypt the media, e.g. for recording or because of the need for transcoding. While this is expected to be done with knowledge and agreement of the end users, one could imagine that an operator mandates such a procedure for all calls and discards all call attempts that do not comply. (The operator will have to make this part of the subscription contract, and can justify this by legal obligations.) The operator will then get all SRTP keys, and can use them in case a call has to be intercepted.

There are some issues with this approach. One of it is that one or two additional messages would have to be passed and processed per call. (Whether one or two messages depends on the method used for key disclosure – different options have been described.) Moreover, the solution currently does not cover roaming scenarios that require that traffic is decrypted in a visited network.

Another issue is the question of how to prevent "cheating", i.e. "disclosing" a wrong key. Note that this issue also arises for all other key management procedures: Two users could agree on performing a secret, additional transformation of the keys as known to a network element that supports LI before using them for encrypting media. There is however a difference: While typically, both end users must agree on a "cheating mechanism", with key disclosure, the intercepted end user can sabotage interception without cooperation of the other end user in the call.

7.5.4 Support of multiparty communication

DTLS-SRTP is specified for point-to-point communications. DTLS-SRTP "inherits" the key exchange methods of TLS ([7]). In these key exchange methods, both peers contribute random values to the key material, so a peer cannot dictate the SRTP master key. Multiple point-to-point sessions (of one peer with multiple other peers) will use different keys typically, so multiparty communications are not supported efficiently.

The Internet-Draft draft-wing-avt-dtls-srtp-key-transport-02 ([17]) defines an extension to DTLS-SRTP that allows a peer to dictate the SRTP master key. E.g., a conference bridge could dictate a single master key to all listeners for the traffic it sends, and it could dictate a new master key to the listeners each time a participant joins or leaves the conference.

7.6 MIKEY-IBAKE Solution

Editor's Note: This solution is to be considered for possible inclusion in the Rel-10 version of TS 33.328.

7.6.1 Introduction

This clause describes a framework solution for IMS media security key management in which additional focus is placed on preserving forward and backwards secrecy, as well as removing a necessity for the high availability key escrow server. In this solution an identity-based encryption concept similar to RFC 5091, RFC 5408 and RFC 5409, is used to generate the session keys, while MIKEY [20] is used for key delivery. Therefore, the framework described is based on protocols already standardized in IETF.

As described in clause 7.1, the TBS solution relies on Key Management Servers (KMS) in the network that create, distribute, and manage keys.

Traditionally, the KMS will have to be online with guaranteed high availability, and have to be networked across operator boundaries. In deployments which can not guarantee such real time high availability KMS, solution specified in this section is preferred.

Moreover, since the keys are created and distributed by the KMS, these servers are de-facto escrow points leading to increased vulnerability and discomfort on the part of end-users. The latter scenario is particularly applicable to Enterprise environments, where the operator offers managed services to the enterprise, but the enterprise requires end-to-end security without the operator knowing what keys were used.

A solution described in this section allows the KMS's to communicate with end-user clients periodically (e.g., once a month) to create a secure identity based encryption framework, while the on-line transactions between the end-user clients (for media plane security) are based on an Identity Based Authenticated Key Exchange framework which allows the participating clients to exchange 'key components' in an 'asymmetric identity based encryption' framework. Observe that the KMS to client exchange is used sparingly (e.g., once a month) – hence the KMS is no longer required to be a high availability server, and in particular different KMS's don't have to communicate with each other (across operator boundaries).

In addition, this framework provides for perfect forwards and backwards secrecy.

Given asymmetric identity based encryption framework is used, the need for Public Key Infrastructure (PKI) and all the operational complexities of certificate management and revocation are eliminated.

Additionally, various IMS media plane features are securely supported – this includes secure forking, retargeting, deferred delivery, pre-encoded content, media clipping, and anonymity.

Extensions of the solution allow for secure conferencing applications, where an IMS conference application server authenticates users into a call but all participants of the call decide on a group key (with contributions from everybody) while the conference server itself does not learn the key (as outlined in requirement 11 in section 5.5.2). Moreover, the group key can be modified to account for new participants and participants who exit a call.

7.6.2 Solution description

7.6.2.1 General

A precondition for a key management scheme as discussed in clause 7.1 is that the users can establish secure connections with the key management server and that mutual authentication is provided. As stated in clause 7.1 it is natural to base the establishment of such trusted and protected connection between the user and the KMS on GBA. Note that if GBA is unavailable, other types of credentials like IKEv2 can be used for establishing this mutual authentication between the user and the KMS. During this transaction, the UE presents its subscription credentials following which the KMS generates a set of private keys (used in IBAKE). As an example, if this transaction is performed once a month, the KMS may choose to generate one key for each day. The number of keys, and the frequency of this exchange is a matter of policy and it may be tied to the subscription. This flexibility is especially useful for prepay customers.

In Figure 23, a conceptual architecture for the discussed key management system is depicted.

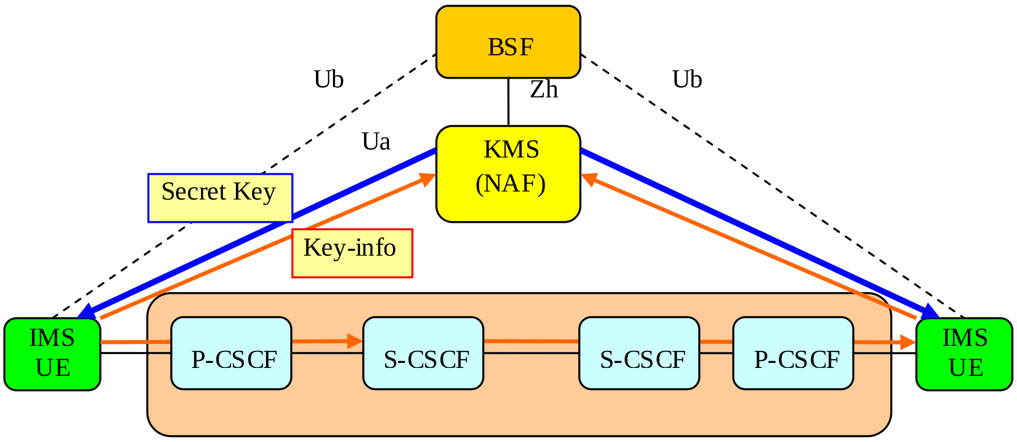

Figure 23: Architecture for key management system

Note that rather than a single KMS, two different KMSs may be involved, one for user A, KMS_A, and one for user B, KMS_B. However, KMS_A and KMS_B do not have to communicate with each other. This scenario is especially applicable in inter-operator scenarios.

Below, a short summary of exchanges involved in MIKEY-IBAKE is provided.

In the example depicted in Figure 24 below, suppose A, B are the two users that are attempting to authenticate and agree on a session key. At the same time, A and B represent their corresponding identities, which by definition also represent their public keys. Let $H_1(A)=Q_A$ and $H_1(B)=Q_B$ be the respective points on the elliptic curve corresponding to the public keys. In effect one could refer to Q_A and Q_B as the public keys as well, since there is a one-to-one correspondence between the identities and the points on the curve obtained by applying H_1. Let x be a random number chosen by A, and let y be a random number chosen by B. Also, let P be a well known point on the elliptic curve E. Encryption below refers to identity based encryption described in Annex A.

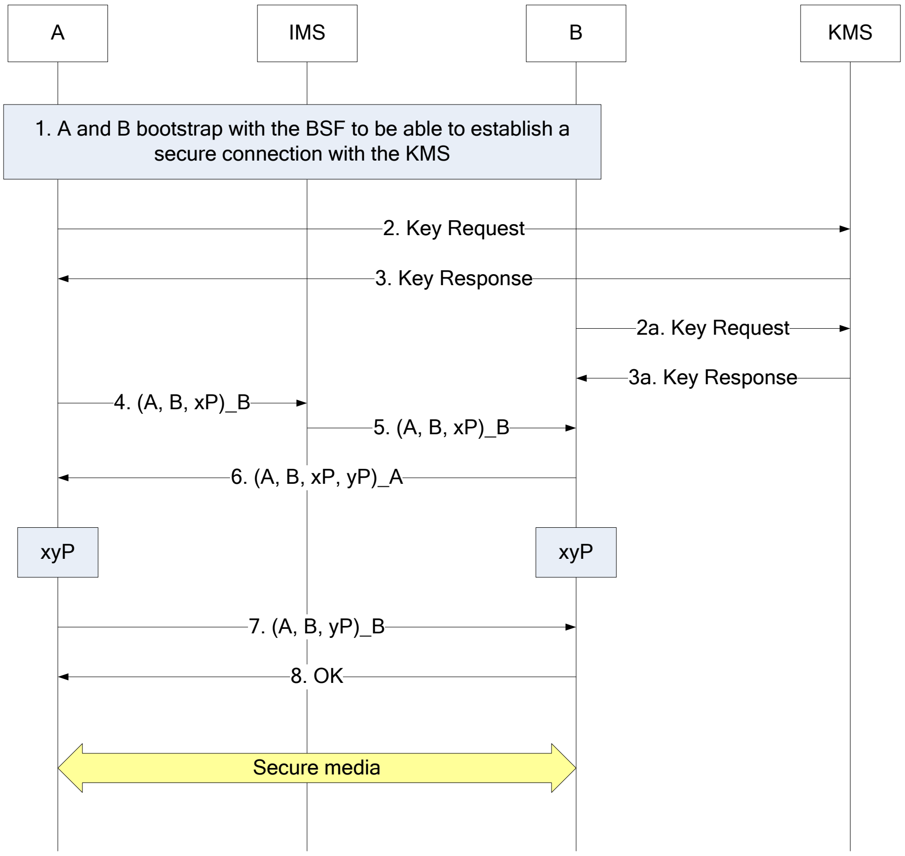

Figure 24: MIKEY-IBAKE Basic Operation

The protocol exchanges consist of the following steps:

1. IMS UE belonging to user A bootstraps with the BSF to be able to establish a secure connection with the KMS which acts as a NAF. This allows the BSF to authenticate the user and the user to indirectly authenticate the KMS.

   If GBA cannot be used, the IMS UE connects and authenticates to the KMS and establishes a shared key, based on a pre-established security association. The exact procedures for this pre-establishment are not described here.

2. The IMS UEs engage in MIKEY exchanges with the KMS and requests private keys (or multiple private keys, e.g., one for each day). These exchanges use a new mode of MIKEY, MIKEY-IBAKE to allow the KMSs to generate user's private key(s).

3. The KMS generates the private key(s) for IMS UEs of user A and B and sends it to the users.

4. The IMS UE of user A computes xP (i.e., P as a point on E added to itself x times, using the addition law on E) encrypts it using B's public key, and transmits it to IMS UE of user B.

5. The IMS core detects the INVITE and handles it in such a way (e.g., forwards it to the LI entity) that a network function, if authorized, can get access to the session key. This step in particular is applicable only to support the active escrow feature needed to satisfy any Lawful Intercept requirement.

6. The IMS UE of user B receives the INVITE including encrypted xP. IMS UE of user B decrypts the message and obtains xP. Subsequently B computes yP, and encrypts the pair {xP, yP} together with A and B's identities using the public key of IMS UE of user A and then transmits it in a response message to A.

7. Upon receipt of this message, IMS UE of user A decrypts the message and obtains yP. At this point, both A and B can compute the session key as xyP. Subsequently IMS UE of user A encrypts yP using B's public key and sends it back in response conformation message to B.

8. The IMS UE of user B accepts the invitation and use of media security.

Observe that A chose x randomly, and received yP in the step 6 of the protocol exchange. This allows A to compute session key as xyP by adding yP to itself x times. Conversely B chose y randomly, and received xP in step 5 of the protocol exchange. This allows B to compute the same session key as xyP by adding xP to itself y times.

7.6.2.2 Discussion

Mutual authentication

As stated above, A chose x randomly, and received yP in the step 6 of the protocol exchange, allowing A to compute session key as xyP. Also, B chose y randomly, and received xP in step 5 of the protocol exchange, allowing B to compute the same session key as xyP. Further observe that the contents of the payload in steps 4, 5 and 7 are encrypted using B's public key. Hence B, and only B, can decrypt these messages. Similarly, the contents of the message in step 6 can be decrypted by A and only A. Also note that, steps 4 through 7 allow B and A to authenticate with each other (by proving that the message was decrypted correctly). This novel feature, allows for A and B to mutually authenticate each other without the aid of any on-line server or certificate authority.

Identity Management

As described above, to encrypt a message a sender uses recipient's public key, generated using the identity (or one of the identities) of the recipient. The identity of the recipient may be in format that specifies a specific user, a group of users or any user. The naming of users and user groups may follow normal IMS conventions and may be extended with use of wildcards.

For a user group it would be natural to have a policy allowing all recipients in the group to use the private key corresponding to the identity of that particular user group. For example, for enterprise users it may be natural to have as a default that private keys corresponding to identity of enterprise are distributed to all enterprise users. Note that due to the properties of identity based encryption, although all the users belonging to a group possibly posses the private key of that group, nevertheless cannot obtain the session key established between a sender and some other user belonging to that same group. This is further explained in clause 7.6.2.2.

To ensure that polices are enforced it is also necessary that a public user identity can be securely bound to an IMS UE. In other words, it is important to tie the identity used by the user to authenticate against the KMS to a (set of) public identity. How it is done using GBA is described in clause 7.1.

Lawful intercept

Editor's Note: Lawful interception issues are for further study.

To be able to provide a clear copy of intercepted communication, the following conditions have to be fulfilled:

1. It must be possible to intercept the traffic (both signalling and media).

2. The session keys used for actual traffic protection have to be available. To make the session keys available KMS functions/services are required.

As stated before, the actual session keys used for traffic protection are generated between the sender and the recipient, thus not known by the KMS. Therefore, for KMS to obtain a session key between users A and B it needs to establish a session key between itself and user A and itself and user B. This approach is also referred as lowful man-in-the-middle-attack. This "attack" could not be detected by the end user by applying the means available through the MIKEY-IBAKE mechanisms,

With signalling traffic routed via the home network, intercept of the signalling traffic in the home network can be done at SIP server(s). This signalling traffic then needs to be routed towards the appropriate KMS in order for this KMS to establish the needed session keys with the corresponding users. In roaming situations, as the SIP signalling traffic normally is confidentiality protected between the IMS UE and the P-CSCF and considering that in current deployments the P-CSCF is located in the home network, the SIP signalling is only available in encrypted format at bearer level in the visited network.

For roaming scenarios, while encrypted SIP signalling and content will always be available, in order to intercept SIP signalling and decrypt the content of communication there has to be an interoperation agreement between the visited network and the entity handling KMS. Typically, the KMS will reside in the home network so that, for LI performed by the visited network, cooperation with the home network is needed.

In line with LI standards, when the VPLMN is not involved in the encryption, only encrypted content would be available for LI in the VPLMN

Users in different KMS domains

Users in different KMS domains will have their private keys generated by different KMSs. As a result, a different set of public parameters (e.g., cryptographic material) can be used to generate public and private keys for users in different KMS domains. To ensure proper encryption/decryption, a sender and recipient need to know exact public parameters used by each side. Nevertheless, if a user in one KMS domain needs to establish a secure call to a user in another KMS domain the involved KMSs do not need to cooperate.

End-to-middle scenarios

In end-to-middle scenarios media protection is between an IMS UE and a network entity. In a scenario when the call is initiated from an IMS UE, the set up of the call would follow the same principles as for an end-to-end protected call. The initiating IMS UE uses the identity of the network entity (e.g. MGWC) to encrypt xP as described above and sends it together with the INVITE. The MGWC intercepts the message, and generates yP in the same way as a receiving IMS UE would have done. The MGWC then sets up the MGW to have media security towards the IMS UE. The media traffic is forwarded in plain in the PSTN.

For incoming calls to IMS UEs, the MGWC checks that at least one terminal registered for the intended recipient has registered media security capabilities and preferences. If there is no media protection capable terminal the call is forwarded in plain. Otherwise the MGWC chooses y and generates yP. The MGWC then inserts the encrypted yP (using the IMS UEs identity) in the INVITE and initiates use of media security in the MGW on the media traffic between the MGW and the IMS terminal.

Perfect secrecy

Observe that x and y are random. Hence the session key xyP is fresh and bears no relation to past or future transactions.

Elimination of Passive Escrow

Observe that, while the KMS (or a pair of KMS's) can decrypt the messages in the exchanges, it is hard to determine xyP given xP and yP. The hardness assumption relies on the Diffie-Hellman problem over Elliptic curves. Also note that, the curves used for IBE are KMS specific, and moreover need not be the same as the curve used to generate the session key. This flexibility offers a wide number of choices, and also eliminates any coordination needed between KMS's.

7.6.2.3 Key forking

In this section, forking is discussed for the case of MIKEY-IBAKE. Forking is the delivery of a request (e.g., INVITE message) to multiple locations. This happens when a single IMS user is registered more than once. An example of forking is when a user has a desk phone, PC client, and mobile handset all registered with the same public identity.

In the example depicted below, assume that IMS UE of user B has multiple contact addresses registered with a single public user identity B. In other words, both B1 and B2 obtain a private key corresponding to a public identity B. In this case, if IMS UE of user A wants to contact the IMS UE of user B, the request will be delivered to both B1 and B2. Assuming that B2 responds to a call, B2 first decrypts the message received using private key associated with the identity B. B2 then chooses random y and sends to A a message including yP and its identity B2 encrypted using A's public identity. Upon receiving this message user A decrypts it, realizes that it is communicating to user B2, and sends a response conformation message including received yP encrypted using B2's public identity.

Observe that B1 is able to decrypt the first message received from user A encrypted using B's public identity, therefore is able to obtain xP. However, it is not able to decrypt the message sent from B2 as it is encrypted using A's identity. Thus, user B1 is not able to obtain yP, and therefore is not able to obtain the session key.

In Figure 25, (M)_X denotes that the message M is encrypted using the public key of X.

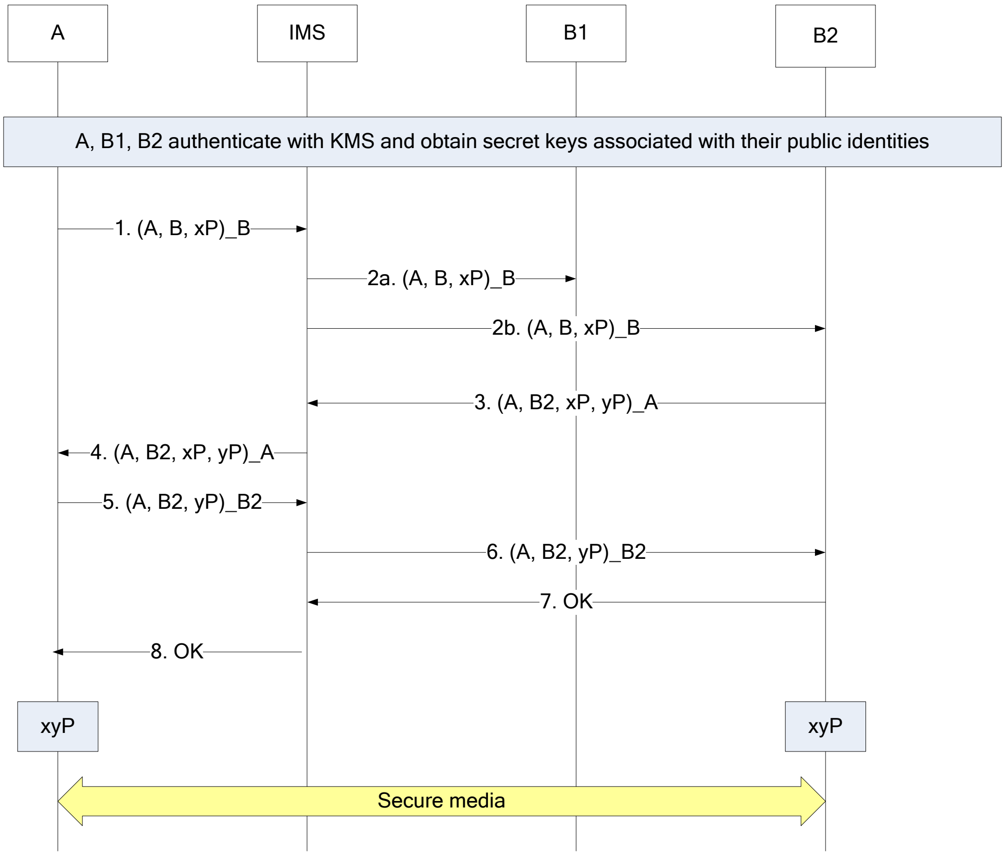

Figure 25: Key Forking

7.6.2.4 Redirection

In this section, session redirection is discussed for the case of MIKEY-IBAKE. Session redirection is a scenario in which a functional element decides to redirect the call to a different destination. This decision to redirect a session may be made for different reasons by a number of different functional elements, and at different points in the establishment of the session.

Session redirection enables the typical services of “Session Forward Unconditional”, “Session Forward Busy”, “Session Forward Variable”, “Selective Session Forwarding”, and “Session Forward No Answer”.

There are two basic scenarios of session redirection. In scenario one, a functional element (e.g., S-CSCF) decides to redirect the session using SIP REDIRECT method. In other words, the functional element passes the new destination information to the originator. As a result the originator initiates a new session to the redirected destination provided by the functional element. For the case of MIKEY-IBAKE this means that the originator will initiate a new session with the identity of the redirected destination.

In the second scenario, a functional element decides to redirect the session without informing the originator. A common scenario is one in which the S-CSCF of the destination user determines that the session is to be redirected. The user profile information obtained from the HSS by the ‘Cx-pull’ during registration may contain complex logic and triggers causing session redirection.

In the example depicted in the figure below, without loss of generality it is assumed that the user B sets up session forwarding to the user C. In this case, user B includes in its user profile its private key SK_B encrypted using C’s identity. Therefore, once the S-CSCF receives the message from user A and decides that the message needs to be redirected, it includes B’s encrypted key in the message redirected to the user C. Upon receiving the message, the user C encrypts the private key, and in turn, the message from A. User C then chooses random y and sends to A a message including yP and its identity C encrypted using A’s public identity. Upon receiving this message user A decrypts it, realizes that it is communicating to user C, and sends a response conformation message including received yP encrypted using C’s public identity.

In Figure 26, (M)_X denotes that the message M is encrypted using the public key of X.

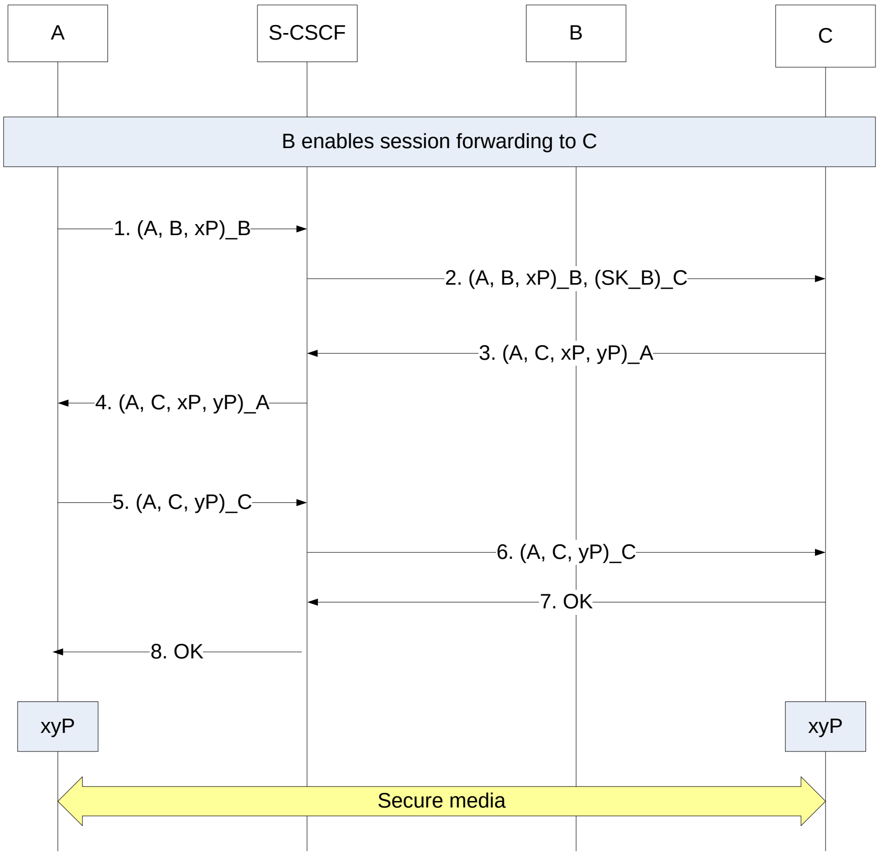

Figure 26: Session Redirection

7.6.2.5 Deferred delivery

In this section, deferred delivery is discussed for the case of MIKEY-IBAKE. Deferred delivery is type of service such that the session content cannot be delivered to the destination at the time that it is being sent (e.g., the destination user is not currently online). Nevertheless, the sender expects the network to deliver the message as soon as the recipient becomes available. A typical example of deferred delivery is voicemail.

Below, the basic scenario of deferred delivery for the case of MIKEY-IBAKE is presented. In the scenario (Figure 27) presented, user A and B's mailbox perform mutual authentication before they agree on the key to be used for decrypting the content of the message intended for deferred delivery.

In the scenario depicted in Figure 27, it is assumed that the user A is trying to reach the user B, who is currently not available, therefore the call is forwarded to the B's 'voicemail' (more generally deferred delivery server). Following the MIKEY-IBAKE protocol, the message received in step 2 by the B's mailbox is encrypted using B's identity, therefore B's mailbox will not be able to decrypt it. B's mailbox chooses random y and computes yP and send its identity and yP IBE-encrypted to the user A. The user A recognizes that the B did not receive the message and that the actual recipient was not able to decrypt the message sent in step 1 by the lack of its identity and xP in the message received in step 4. Therefore, the user sends a new message containing A's identity, B's mailbox identity, xP and yP all IBE-encrypted using B's mailbox identity. The user A also chooses a random a and includes its identity and aP encrypted using B's public key. Upon reception of this message, the B's mailbox accepts aP as the session key for the message intended for B and returns A's identity and xP to the user A to complete the authentication. Subsequently, when B is online and checks 'voicemail' (checks with the deferred delivery server), B can obtain the encrypted value of aP from the mailbox server.

Note that B may have to authenticate with the mailbox to obtain the key (this is not shown in the figure below) – this could be based on existing authentication mechanisms already in place.

Note also that the mailbox needs to establish a secure communication path to B and communicate the original authenticated user's identity with every message.

In addition, the mailbox needs to implement a MIKEY-IBAKE client to mutually authenticate with A and B in order to protect against A spoofing the source of deposited messages towards B.

Note further, that in order to prevent alterations of deposited multimedia messages by the man-in-the-middle attacker, the integrity protection of IMS transport is expected. In order to achieve full end-to-end integrity protection at the IBAKE layer for the deferred delivery solution is required, which means that further modification of the protocol may be needed.

The practicality of addressing these additional requirements on mailbox security and potentially needed modifications to the IBAKE protocol need to be taken into account when deciding the suitability of MIKEY-IBAKE for deferred delivery scenarios.

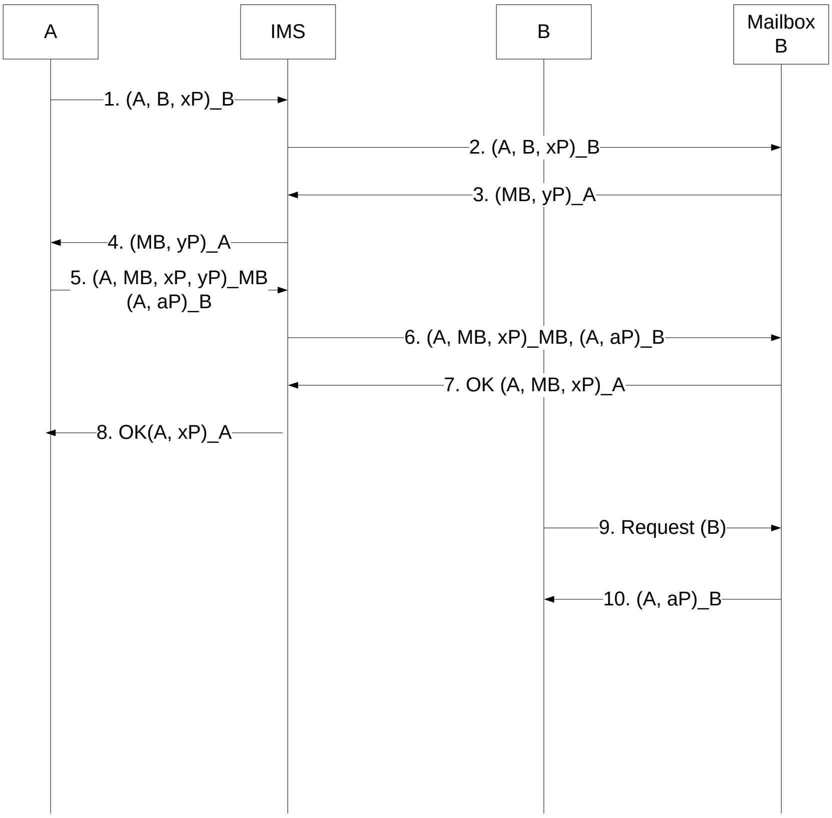

Figure 27: Deferred Delivery

7.6.2.6 Group and conference calls

7.6.2.6.1 General

In this section key management scheme based on MIKEY-IBAKE is discussed. To satisfy requirement 11 specified in clause 5.5.2, the assumption is that the server relaying multiparty communication (e.g. a conference bridge) does not know the group key, while all the users have access to the same group key.

In Figure 28, it is assumed that there is a conference server (AS/MRFC) that invites users to the conference call. This could be a result of for example previously received REFER request from another user. An alternative approach would be to delegate this function to one of the users (e.g., conference chair). Although this alternative is not shown below the approach would be similar and the computation of the group key would be the same.

In the description below, all messages are IBE encrypted (e.g., if a user Y is sending a message M to a user X, then the massage M is encrypted using X's identity) using the appropriate identity. In the figure this is denoted as (M)_X meaning the message M is IBE encrypted using the public key of X. In the first set of exchanges with the conference server users A_1, A_2, and A_3 choose random a_1, a_2, and a_3 respectively and each user A_i sends $w_i = a_iP$ to the conference server. In the second set of exchanges the conference server sends all received a_iP's to every user, while each user sends $z_i = a_i(a_{i+1}P - a_{i-1}P)$. In the finial exchange, the conference server sends all received z_i's to each user. Upon this, all conference participants are able to compute the group key as follows:

$K_i = 3a_iw_{i-1} + 2z_i + z_{i+1}$.

Note that $K_1=K_2=K_3$. In other words, all users generate the same session key. Also note, while users A_1, A_2, and A_3 are able to generate the group key, conference server is not since while it knows the z_i's and w_i's, only individual users know their randomly chosen a_i.

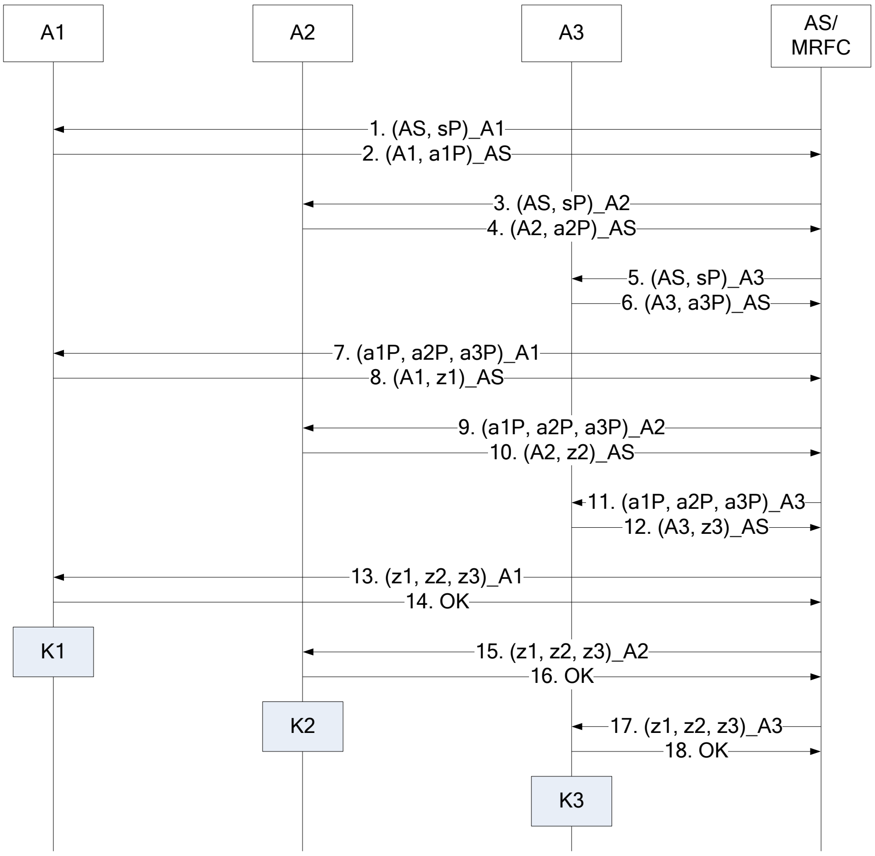

Figure 28: Group and Conference Calls

For simplicity reasons, above discussion focuses on 3 conference call participants. However, the above procedures can be generalized to *n* participants. In case of *n* participants, the group key is generated as

$K_i = na_i\, w_{i-1} + (n-1)z_i + (n-2)z_{i+1} + \ldots + z_{i-2},$

where w_i and z_i are as defined above.

7.6.2.6.2 Adding and deleting users

One of the important features of the protocol is that, the group key changes every time a new user is admitted or an existing user exits the call. This ensures that new users don't learn the group key before they were added to the call, and users who leave the call prematurely do not gain access to the conversations after the call.

Observe that, when a new user is added, and there are N users in the system already, then there will be a total of N+1 users in the system. When these users are placed in a circle, then the user next to the N-th user is now the (N+1)th user (and not the 1^{st} user, which was the case prior to admitting the N+1th user). The protocol to admit a new user works as follows:

- The new user authenticates with the conference server using IBAKE, similar to every user. This allows the user to be admitted (and authorized to the call), and the new user is guaranteed of joining the correct conference (via authentication of the conference server).
- Let $z_{N+1}= a_{N+1}P$ be the value chosen by the new user during authentication.
- The conference server then sends the set $\{z_i\}$ for all i=1 to N+1 to all users, either broadcast or uni-cast. This allows all users to learn of the new user, and determine their new neighbors. Observe that the neighbor list changes only for users 1, N, and N+1.
- Users 1, N, and N+1 then compute their corresponding value of w, and send it back to the conference server (individually).
- The server then sends an updated list of $\{w_i\}$ to all users.
- All participants then re-compute the group key using the same relation as earlier, except N is replaced by N+1 and the new values of z_i and w_i.

When a user exits the conference call, then no new authentication procedures have to be executed, but the group key changes. . The procedure works as follows:

- The conference server learns about the user exiting the conference call.
- Subsequently, the conference server informs everybody of this event and information pertaining to which user (not just identity, but also includes the order) exited the call. In order to simplify matters, the conference server may re-send the new list $\{z_i\}$
- This allows all users to re-discover their neighbors, and recompute w_i if necessary.
- All those participants remaining in the call, for whom w_i changed, will inform the conference server their new value.
- The conference server then sends the updated list $\{w_i\}$
- All participants then re-compute the group key using the same relation as earlier, except N is replaced by N-1 and the new values of w_i are used.

7.6.3 Compliance of MIKEY-IBAKE with requirements

7.6.3.1 General

Clause 5 identifies 3GPP requirements for IMS media plane security. In this clause the proposed solution is evaluated against the identified requirements.

7.6.3.2 Compliance of IBAKE with 3GPP requirements

7.6.3.2.1 General

This clause discusses the 3GPP requirements.

7.6.3.2.2 Lawful intercept

The MIKEY-IBAKE solution allows compliance with LI requirements in both the home network and visited network. As described in clause 7.2.6.1 for the case of signalling traffic routed via the home network the LI system must have access to standard user services from a KMS. For roaming scenarios, while encrypted SIP signalling and content will always be available, in order to intercept SIP signalling and decrypt the content of communication there has to be an interoperation agreement between the visited network and the entity handling KMS.

7.6.3.2.3 Security requirements

In the following discussion of the compliance with the security requirements, it is assumed that user plane traffic is properly secured based on the keys established using MIKEY-IBAKE.

MIKEY-IBAKE protocol encrypts the exchanged key components (i.e. xP and yP), therefore independent of any SIP signalling protection assumptions MIKEY-IBAKE provides security on its own. Key information is also protected while stored or handled in SIP proxies.

The protocol framework inherently supports mutual authentication of entities involved in the key exchange, therefore requirement 7 specified in clause 5.4 is satisfied.

The KMS itself may be a target for attacks. It should be protected in the same way a BSF in a GAA/GBA deployment would be.

7.6.3.2.4 Requirements related to SIP based call features

MIKEY-IBAKE solution supports secure multiparty communications where the server relaying multiparty communication (e.g. a conference bridge) does not know the group key as specified in condition 19.

7.6.3.2.5 Architectural requirements

MIKEY-IBAKE is designed such that it supports e2e security as well as e2m and e2ae security. Therefore, MIKEY-IBAKE is able to support requirements 13-16 .

The requirement to support media recording is supported by MIKEY-IBAKE independent of if the recording is of plaintext media or if it should be protected.

MIKEY-IBAKE is standalone key management protocol and as such can be implemented in non-IMS UEs. However, practical usefulness of such implementation is limited.

As for the impact on existing network entities as discussed in requirement 21, MIKEY-IBAKE requires new functionality performed by KMS. KMS may be deployed in already existing network equipment which would obviously have an impact on that particular network entity. At the same time this would reduce OPEX and CAPEX as compared to implementing a KSM as a stand alone entity. Network nodes that need to control media protection functionality in e2m scenarios would also be impacted.

7.6.3.2.6 Scalability, cost and performance

Similar to TBS with a KMS, MIKEY-IBAKE will require a KMS supporting its users. Its size/performance grows proportional to the number of users. However, there is no technical challenge to implement a KMS supporting all IMS users of an operator as can be seen from specifications and implementations of other nodes in cellular and IMS systems. As stated earlier, MIKEY-IBAKE does not require per-session KMS services which dramatically reduces the complexity of network support required.

7.6.3.2.7 Requirements regarding the access network type

MIKEY-IBAKE complies with requirements regarding the access network type requirements, in the following way:

- it is access network independent;
- It works independently of any of the different authentication methods defined for IMS.

As stated previously, MIKEY-IBAKE requires new functionality performed by KMS.

7.6.3.2.8 Backward compatibility and migration

Similar to TBS and SDES solutions, MIKEY-IBAKE complies with backward compatibility and migration requirements. In particular, keys and other parameters can be negotiated individually for each call, and downgrading attacks cannot be performed if secure SIP signalling is assumed.

7.6.3.2.9 Other requirements

MIKEY-IBAKE is standalone key management protocol and as such can also be used for exchanging keys for other media plane security protocols.

MIKEY-IBAKE provides means for a party to get assurance about the identity of any other party in the session when the party joins a point-to-point session, therefore satisfying requirement 40.

As described above MIKEY-IBAKE solution supports deferred delivery of media.

7.6.3.3 Compliance of IBAKE with IETF requirements

7.6.3.3.1 General

In this clause relevant IETF requirements are discussed.

7.6.3.3.2 Security requirements

MIKEY-IBAKE satisfies the following security requirements specified in RFC 5479 [2]:

> R-PFS: The media security key management protocol MUST be able to support perfect forward secrecy.
>
> R-COMPUTE: The media security key management protocol MUST support offering additional SRTP cipher suites without incurring significant computational expense.
>
> R-DOS: The media security key management protocol MUST NOT introduce any new significant denial-of-service vulnerabilities (e.g., the protocol should not request the endpoint to perform CPU-intensive operations without the client being able to validate or authorize the request).
>
> R-AGILITY: The media security key management protocol MUST provide crypto- agility, i.e., the ability to adapt to evolving cryptography and security requirements (update of cryptographic algorithms without substantial disruption to deployed implementations).
>
> R-DOWNGRADE: The media security key management protocol MUST protect cipher suite negotiation against downgrading attacks.
>
> R-PASS-MEDIA: The media security key management protocol MUST have a mode that prevents a passive adversary with access to the media path from gaining access to keying material used to protect SRTP media packets.
>
> R-PASS-SIG: The media security key management protocol MUST have a mode in which it prevents a passive adversary with access to the signaling path from gaining access to keying material used to protect SRTP media packets.
>
> R-SIG-MEDIA: The media security key management protocol MUST have a mode in which it defends itself from an attacker that is solely on the media path and from an attacker that is solely on the signaling path. A successful attack refers to the ability for the adversary to obtain keying material to decrypt the SRTP encrypted media traffic.
>
> R-ID-BINDING: The media security key management protocol MUST enable the media security keys to be cryptographically bound to an identity of the endpoint.
>
> R-ACT-ACT: The media security key management protocol MUST support a mode of operation that provides active-signaling-active-media-detect robustness, and MAY support modes of operation that provide lower levels of robustness.

As previously stated MIKEY-IBAKE relies on KMS to provide private keys and as such violates the following requirement:

> R-CERTS: The key management protocol MUST NOT require that end-users obtain credentials (certificates or private keys) from a third- party trust anchor.

7.6.3.3.3 Forking/retargeting

IETF-requirements in RFC 5479 [2] state as follows:

R-FORK-RETARGET: The media security key management protocol MUST securely support forking and retargeting when all endpoints are willing to use SRTP without causing the call setup to fail. This requirement means the endpoints that did not answer the call MUST NOT learn the SRTP keys (in either direction) used by the answering endpoint.

R-DISTINCT: The media security key management protocol MUST be capable of creating distinct, independent cryptographic contexts for each endpoint in a forked session.

Section 7.6.2.2 and 7.6.2.3 describe how forking and retargeting is done for the case of MIKEY-IBAKE. As described in these sections the endpoints that did not answer the call will not learn the session key (i.e., SRTP key) used by the answering endpoint. Section 7.6.2.2 also describes how distinct, independent cryptographic contexts for each endpoint in a forked session is created.

7.6.3.3.4 Early media

Similar to SDES and TBS with KMS, MIKEY-IBAKE allows decryption only after successful transmission of the SDP answer, so encrypted media would be clipped before that. However, 3GPP generally assumes SBCs in the media path that block media before the delivery of the SDP answer.

8 Conclusions

For Release 9 the following solutions are to be included in normative specifications:

- SDES described in clause 7.3 for e2ae and e2e media protection.
- MIKEY-TICKET described in clause 7.1 for high security e2e media protection.

Annex A (informative): Identity Based Encryption

Identity Based Encryption protocol [D. Boneh and M. Franklin, RFC 5091, RFC 5408 and RFC 5409] is an asymmetric cryptographic encryption protocol that allows participants to use an 'identity' (example: email-id, or domain name) as the public key. As such, the IBE protocol eliminates the need for large scale public key infrastructure which is often associated with public key encryption methods such as RSA. Boneh and Franklin's approach to the problem uses bilinear maps on an elliptic curve over a finite field, and relies on the bilinear decisional Diffie-Hellman problem.

The protocol involves the following mathematical tools and parameters:

- Let E be an elliptic curve over a finite field F, and let P be a point of large prime order.
- Let e: E x E -→ G be a bi-linear map on E. Typical example is the Weil pairing, and hence G will be the group of n-th roots of unity where n is a function of the number of points on E over F.
- Let a non-zero positive integer s, be a secret stored in a KMS. This is a system-wide secret and not revealed outside the KMS.
- Let P_{pub} = sP be the public key of the system that is known to all participants. Recall sP denotes a point in E, since E is a group.
- Let H_1 be a known hash function that takes a string and assigns it to a point on the elliptic curve, i.e., $H_1(A) = Q_A$ on E, where A is usually the identity.
- Let $d_A = sQ_A$ be the private key computed by the KMS and delivered only to A.
- Let H_2 be a known hash function that takes an element of G and assigns it to a string.

Let m be a message that has to be encrypted and sent to A. The encryption function described by Boneh-Franklin is as follows:

- Let $g_A = e(Q_A, P_{pub})$, and let r be a random number.
- $\text{Encryption}_A(m) = (rP, m \text{ xor } H_2(g_A^r))$; in other words the encryption output of m has two coordinates u and v where u=rP and v= m xor $H_2(g_A^r)$

In order to decrypt (u,v), A recovers m using the following formula:

- $m=v \text{ xor } H_2(e(d_A,u))$.

The proof of the formula is a straight forward exercise in bilinear maps, and the fact A has the secret d_A (private key known only to A but not other participants).

Annex B (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Mar 2009	SA-43	SP-090134	--	--	Presentation to SA for information	--	1.0.0
Dec 2009	SA-46	SP-090830	--	--	Presentation to SA for information	1.0.0	1.6.1
Mar 2010	SA-47	SP-100107	--	--	Presentation to SA for approval	1.6.1	2.0.0
Mar 2010	--	--	--	-	Publication of SA approved version	2.0.0	9.0.0
Jun 2010	SA-48	SP-100246	001	1	Removal of editor's notes	9.0.0	9.1.0
2012-06	SA-56				Upgraded with no technical chages	9.1.0	10.0.0
2012-06	SA-56	SP-120386	002	1	MIKEY-IBAKE deferred delivery solution	10.0.0	11.0.0
2012-09	SA-57	SP-120616	004	-	Correction to MIKEY-TICKET deferred delivery solution	11.0.0	11.1.0