3GPP TSG WG3 S3#18
Phoenix, Arizona
23rd May – 27th May, 2001

Tdoc S3-0100199

Agenda Item: 9.3

Source: Ericsson

Title: Integrity protection for SIP signaling

Document for: Discussion

1 Scope and objectives

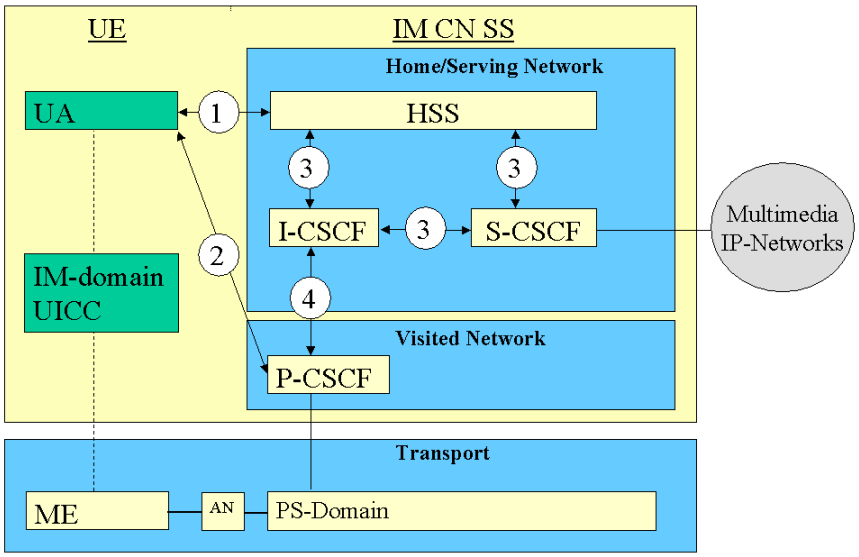
The scope of this document is to discuss the integrity protection of Session Initiation Protocol (SIP). This document describes two optional solutions: Cryptographic Message Syntax (CMS) and IP Sec. Preliminary arguments for and against both solutions are given.

2 Background

The scope of the discussion is the integrity protection of SIP signaling traffic [SIP]. It has been recognized that SIP does not offer appropriate security mechanism for integrity protection. It has also been proposed that SA3 should NOT develop any new mechanisms because some existing ones could be applied.

A working assumption in SA3 has been that AKA defined in R'99 shall be applied for SIP. This means that long-term secret key K used for SIP authentication is shared between USIM and home network. Shorter-term secret key used for integrity protection (IK) is created during the registration in the USIM and home network. In the roaming case, P-CSCF receives the needed key from the home network.

A working assumption on confidentiality has been that SIP signaling using UMTS access will rely on UMTS confidentiality (terminated at RNC) and DNS mechanisms.



3 SIP level protection

IETF S/MIME working group has developed CMS from PKCS#7 [CMS, PKCS7]. CMS defines encapsulation syntax for data protection. CMS facilitates the exchange of arbitrary cryptographic messages since it defines how digital signatures, encryption and message authentication codes are constructed and interpreted. Even though most cryptographic operations defined in CMS are based on public-key cryptography, the use of secret keys is also possible for some operations.

CMS defines six content types: data, signed-data, enveloped-data, digested-data, encrypted-data, and authenticated-data. Authenticated-data Content Type can be used to protect the integrity of any type of data by the means of symmetric cryptographic keys and message authentication codes (MAC). This content type should be used for SIP integrity protection if CMS is applied.

When delivering via the Internet, CMS packets are often attached into a S/MIME message. S/MIME provides at least the following complementary features for the usage of CMS:

- S/MIME defines how binary CMS packages can be encoded and transmitted in ASCII format.
- It defines how CMS packages are interpreted in PKI context.
- It provides a mechanism for S/MIME entities to communicate their cryptographic capabilities.
- It provides information in ASCII format about the type of CMS package attached: it is not necessary to decode the CMS package to find out the content.
- It defines basic requirements for supported algorithms.

Unfortunately, S/MIME does not support all CMS content types. Especially those, which are in the interest of SIP integrity protection, are not supported. Supported content types are data, signed-data and enveloped-data, in other words, only those which are needed for PKI applications. For these reasons, it is easier to apply CMS alone than together with S/MIME for SIP integrity protection.

3.1 How to use CMS for integrity protection

CMS Authenticated-data Content Type supports the use of *hashed message authentication codes* (HMAC). In general, HMAC serves as integrity protection mechanism but also as an authentication service. In HMAC, both the content to be protected and the symmetric cryptographic key are taken as input for HASH operation. HMAC is sent to the receiver together with the content. Repeating the hash operation with the received data and a secret key can prove integrity of the message to the receiver. The identity of the message originator can be concluded because only the owners of the secret keys are capable of generating HMAC tokens.

In the case of SIP, protected content, used as input for HMAC, should include some selected fields from the SIP header and a secret key shared between communicating entities. HMAC value together with algorithm information should be attached to the SIP message. Unfortunately, there is no such mechanism in SIP. Current SIP specification do introduce *Encryption* header which could be used for integrity protection, however, it can be used only in end-to-end cases and with public key cryptography.

For these reasons, two new header types must be specified for SIP. *End-to-end header* would be created if the home network requires integrity protection. In this case, the full SIP message can not be integrity protected because the intermediate SIP proxies are allowed to manipulate some data fields for routing or other purposes. The specific content of the end-to-end header is ffs. *Hop-by-hop header* would be used between UE and P-CSCF. In this case, the whole SIP message could be integrity protected because the integrity header is removed when the integrity has been checked in P-CSCF.

The example below illustrates how integrity protection using CMS would look in SIP.

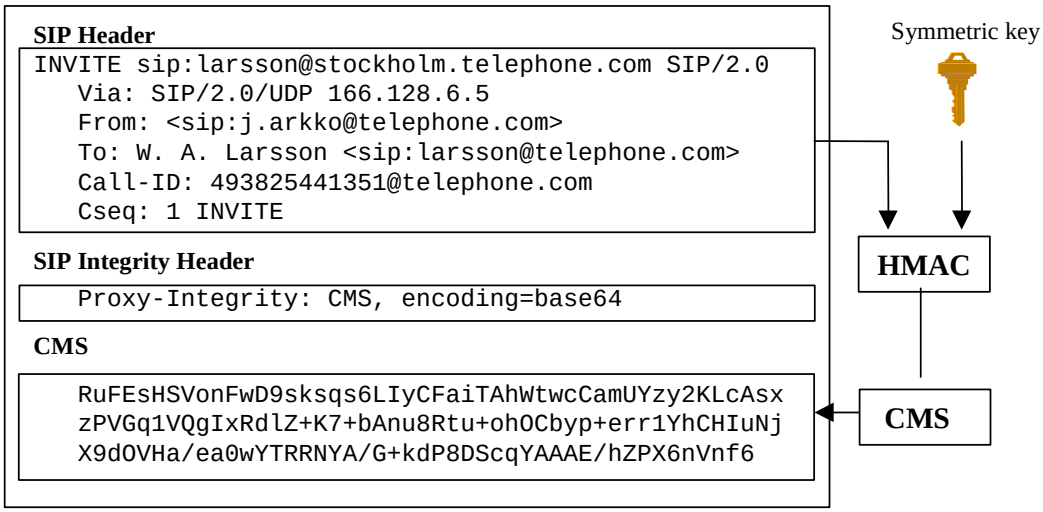


Figure 1 Integrity protected SIP message using CMS

When UE sends a SIP REGISTER request for P-CSCF, long-term secret key (K) situated in the home network must be applied. P-CSCF forwards the registration message to the home network for user authentication. During the authentication process, P-CSCF receives short-term integrity key (IK) from the UE's home network. At the same time, the same key is generated in the USIM based on a random number and authentication token received from the home network. When UE sends SIP INVITE command to the P-CSCF, the integrity of the request can be verified against the secret key.

3.2 Message payload

Usage of full Authenticated-data structure would consume a lot of message payload. For this reason, minimum set of data fields should be defined. In a standard CMS format, some optimization can be done since optional data field can be left away. However, there are some mandatory data fields, which are not vital in 3G environment and which consume too much space. For example, Authenticated-data structure includes authenticated data inside the cryptographic message. In our case, this would cause double amount of data because the SIP header (which is authenticated) has to appear also in the message body itself.

Table 2 demonstrates how the data fields could be optimized for 3G purposes.

Data fields	Standard	Optimized
Object Identifier		Shall be present
Version		Shall be set to 0
Originator	OPTIONAL	May be present
Recipient		May be present
Mac Algorithm Identifier		Shall contain the HMAC-SHA1 identifier
Digest Algorithm	OPTIONAL	May be present
Content type		Shall contain content type identifier
Content		Shall not be present
Authenticated Attributes	OPTIONAL	May be present
Message Authentication Code		Shall be present
Unauthenticated Attributes	OPTIONAL	May be present

Table 2 Example of optimized authenticated-data content type

The amount of overhead caused by the SIP Integrity Header together with CMS data can be very roughly estimated as follows:

- 37 bytes of SIP Integrity Header
- 70 bytes of CMS
- 17 bytes of base 64 encoding of CMS

This makes a total of 124 bytes. (Note: Exact amount of bytes depends on further standardisation.)

3.3 Effects to UMTS and IETF Standardization

In order to make this possible, the following standardization has to take place:

- SIP Integrity mechanism should be defined in IETF. The exact requirements on what is needed to do this are ffs, but probably include the publication of an RFC. Following issues should be included:
  - Hop-by-hop integrity protection: header type and list of protected data types.
  - SIP is not able to transfer binary data. For this reason, encoding of attached, non-ASCII data blocks should be defined in SIP. Allowed encoding methods should be defined.
- CMS has no mechanisms for replay protection. Some mechanism for replay protection must be provided in order to prevent, for instance, a replay of an INVITE for an expensive call. The exact implementation of replay protection in CMS is ffs, but could perhaps mean for instance an IPsec-like sequence number field, or the time stamp option of CMS could be specified to act also for replay protection.
- An optimized profile of CMS Authenticated-data Content Type must be defined. The exact requirements on what is needed to do this are ffs, but probably include the publication of an RFC.
  - For example, CMS mandates the presence of *recipientInfos* data field in Authenticated.data structure. RecipientInfos field carries a specific content encryption key, which is encrypted with another cryptographic key (key transport, agreement or encryption key). In 3G, this kind of ‘double’ encryption produces unnecessary message expansion.
- All new object identifiers, such as SIP Integrity Headers, must be registered to IANA/IETF.

4 IP level protection

4.1 Introduction

IPsec is a standard security mechanism for the protection of IP packets. In this section we study the use of IPsec ESP for the integrity protection between UE and P-CSCF.

4.2 How to use ESP for SIP

The IPsec ESP protocol [ESP] should be used in transport mode, using one of the standard algorithms such as SHA1 for integrity protection. Since no encryption is performed (so called ESP NULL case), IVs and padding to the block size will not be used. However, padding must be performed to a four byte boundary according to the RFC rules.

The amount of overhead caused by the ESP can be calculated as follows:

- 4 bytes of SPI
- 4 bytes of sequence number
- 0..3 bytes of padding (we can assume an average of 2 bytes for ease of calculations)
- 2 bytes of pad length and next protocol
- 12 bytes of SHA1 MAC

This makes a total of 24 bytes.

4.3 Protection offered

The above use of ESP offers the following protection:

- Full integrity protection for the packet, excluding the IP header.
- Replay protection.

Note that the SIP registration procedure will have to act as the security setup mode, and create the SA with the desired parameters, such as setting the algorithm, keys, and replay protection flags correctly.

4.4 Fixed Policies

Once the creation of an IPsec SA has been agreed through the authentication and security mode setup procedures, both the SIP client and the proxy must install a fixed security policy to the IP layer concerning the particular pairs of IP addresses and ports. For instance, the following policy could be installed by a terminal running at fe80::1 on port 12345 and contacting P-CSCF at fe80::2 on port 5000:

```
Fe80::1 port 12345 -> Fe80::2 port 5000: use IPsec SA_out_1

Fe80::2 port 5000 -> Fe80::1 port 12345: use IPsec SA_in_1
```

Such policy, addressing, and security association information must be kept both in the CMS and in the IPsec solutions. There may be some differences in terms of how easy it is on various operating systems to dynamically modify the IPsec policy and SA data bases, and how efficiently larger proxies can treat large numbers of policies and SAs. However, on a conceptual level there is little difference in, for instance, memory requirements of the two solutions.

4.5 ESP and Multiple-Client SIP case

It has been stated that IPsec can't securely be used in situations with multiple SIP clients, because client 1 might send packets through an IPsec SA that was originally created for client 2. We will discuss whether this is in fact the case or not.

First we will discuss the case of multiple SIP clients and users running on one UE. Here it is necessary for the 'operating system' or 'IP stack' of the UE to exert certain amount of control over the UDP/TCP ports opened by the SIP clients. Typically, operating systems prevent a port to be reopened once it has been allocated to a running application. This is sufficient to prevent another application from sending or receiving a packet that matches the same fixed policies and leads to the use of another user's connection. It is required for an application to stay alive and hold the port reserved for as long as the security association exists. This sounds like a reasonable requirement given that an application that didn't do this couldn't receive any incoming SIP communications either.

Then we will discuss the case of a split UE, for instance a phone and a laptop. In this case, the worry is that since the two operating systems on these devices will not run in a co-ordinated manner, the phone will happily accept packets from the laptop. There is a danger that the phone will also place the laptop packets under the protection of an IPsec SA that was meant for a SIP client running on the phone. This will only happen if the laptop and the phone use the same IP address. There are two reasons why this isn't a problem:

- Since IPsec transport mode is used, packets coming from the laptop shouldn't be encapsulated in this manner according to the RFCs.
- Even if they were, the assumption about the same IP address on both devices simultaneously will break normal IP communications without security. For instance, if both devices open the same port for their SIP clients, communications to the P-CSCF will be mixed up through packets being sent only to one client, mixed with each other and so on.

4.6 Effects to UMTS Standardization

In order to make this possible, the following standardization has to take place:

- Profiling of IPSec and its algorithms for this particular use must be specified by SA3.

5 Conclusions

This document demonstrated on how the integrity of SIP signaling traffic could be protected both on the SIP level using CMS Authenticated-data Content type and on the IP level using IPSec ESP. The solutions had following main characteristics:

SIP level protection:

- New SIP Integrity Headers were used for e2e and h2h integrity protection
- Optimization of CMS Authenticated-data Content Type was proposed
- Mechanism for attaching non-ASCII data for SIP messages was required

IP level protection:

- IPSec ESP was applied on transport mode
- The use of fixed security policies was proposed

Both solutions are relevant for discussion. SIP level protection requires more standardization actions but is more suitable for future development of SIP (see table below). For example, it can be used for e2e SIP as well as for other applications in terminal. Furthermore, CMS will play an important role in the implementation of mobile commerce security and application level PKI. On the other hand, IP level protection is more effective and efficient than SIP level protection.

Evaluation criteria for integrity protection	**SIP level protection**	**IP level protection**
Extra bandwidth	124 bytes (longer headers and base64 encoding)	24 bytes
Reuse of the mechanism for other purposes in the terminal.	Yes, for e-mail and e2e SIP	Not at present. Future applications for terminals unknown.
Status and completeness of specifications	Stable, but some additional features are needed	Stable, no additional features needed
Possibility to use the solution end-to-end	Yes	No

Ericsson proposes to define the integrity protection mechanisms at SIP level mainly because of the ease of implementation due to reuse reasons, and because the same scheme could perhaps be used also for later end-to-end security in SIP. As an alternative to SIP level security IPSec-ESP with fixed policies is also acceptable. This would be a somewhat more bandwidth-efficient mechanism due to the longer headers and base64 encoding in CMS.

References

[CMS] Cryptographic Message Syntax, RFC 2630, IETF, June 1999.

[ESP] IP Encapsulating Security Payload (ESP), RFC 2406, IETF, November 1998.

[PKCS7] PKCS #7 - Cryptographic Message Syntax Standard, An RSA Laboratories Technical Note, Version 1.5, November 1, 1993.

[SIP] SIP: Session Initiation Protocol, Internet Draft, IETF, November 24, 2000.