3GPP TSG SA WG3 Security — S3#13 **S3-000382**

23-26 May, 2000

Yokohama, Japan

Source: **S3**[1]

To: **N4**

Title: **MAP security Layer III**

In their meeting #13, S3 agreed two changes to MAP security Layer III as specified in TS 33.102, v3.4.0, section 7. These changes are:

- confidentiality and integrity protection are made independent of each other by making the hash function used to provide integrity protection a keyed hash function (MAC function); for a justification of this change see doc S3-000312 with the amendment described in S3-000355;
- the time variant parameter (TVP) used for replay protection is defined as a 32 bit time-stamp; for a justification of this change see doc S3-000368.

Documents 312, 355 and 368 are attached to this document.

The changes are described below.

With these changes the specification of MAP security layer III, as specified in TS 33.102, v3.4.0, section 7, is considered to be sufficient by S3 to allow CN4 to stabilise the MAP message syntax. Please note that MAP security will not be part of the R'99 security architecture specification which will be frozen in TS 33.102, v.3.5.0. MAP security will be re-introduced into TS 33.102 Release 2000.

TS 33.102, v3.4.0, section 7.4.2.2 is replaced with the following:

7.4.2.2 Protection Mode 1

The message body of Layer III messages in protection mode 1 takes the following form:

TVP||Cleartext|| $H_{KSXY(int)}$(TVP|| MAP Header||Security Header||Cleartext)

where "Cleartext" is the message body of the original MAP message in clear text. Therefore, in Protection Mode 1 the Layer III Message Body is a concatenation of the following information elements:

- Time Variant Parameter TVP
- Cleartext
- Integrity Check Value

Authentication of origin and message integrity are achieved by applying the message authentication code (MAC) function *H* with the integrity session key KSXY(int) to the concatenation of Time Variant Parameter TVP, MAP Header, Security Header and Cleartext.

The TVP used for replay protection of Layer III messages is a 32 bit time-stamp. The receiving network entity will accept a message only if the time-stamp is within a certain time-window. The resolution of

---

[1] Contact: Peter Howard, Vodafone Ltd; tel +44 1635 676206; email peter.howard@vf.vodafone.co.uk

the clock from which the time-stamp is derived must be agreed as a system parameter, the size of the time-window at the receiving network entity need not be standardised.

TS 33.102, v3.4.0, section 7.4.2.3 is replaced with the following:

7.4.2.3 Protection Mode 2

The Layer III Message Body in protection mode 2 takes the following form:

$$TVP \| E_{KSXY(con)}(\text{Cleartext}) \| H_{KSXY(int)}(TVP \| \text{MAP Header} \| \text{Security Header} \| E_{KSXY(con)}(\text{Cleartext}))$$

where "Cleartext" is the original MAP message in clear text. Message confidentiality is achieved by encrypting Cleartext with the confidentiality session key KSXY(con). Authentication of origin and message integrity are achieved by applying the message authentication code (MAC) function *H* with the integrity session key KSXY(int) to the concatenation of Time Variant Parameter TVP, MAP Header, Security Header and $E_{KSXY(con)}$(Cleartext).

The TVP used for replay protection of Layer III messages is a 32 bit time-stamp. The receiving network entity will accept a message only if the time-stamp is within a certain time-window. The resolution of the clock from which the time-stamp is derived must be agreed as a system parameter, the size of the time-window at the receiving network entity need not be standardised.

TS 33.102, v3.4.0, section 7.4.3 is replaced with the following:

7.4.3 Structure of Security Header

The security header is a sequence of the following data elements:

- Protection Mode
- Key Identifier
- Algorithm Identifier
- Mode of Operation
- Initialisation Vector
- Sending PLMN Id

NOTE: Whether the Initialisation Vector is needed depends on the mode of operation of the encryption algorithm.

3GPP TSG SA WG3 Security — S3#13 **S3-000312**

23-26 May, 2000

Yokohama, Japan

(Ad-hoc meeting on MAP Security, Yokohama, 23 May, 2000)

Source: **Siemens AG**[1]

Title: **Independence of confidentiality and integrity in MAP Layer III and other layer III issues**

Document for: **Discussion and decision**

Agenda Item: **<Ad-hoc MAP Security meeting>**

Abstract

MAP security layer III, as described in 3G TS 33.102 v3.4.0 (UMTS Security Architecture) uses a well-known method to provide integrity using an encryption function and a hash function. In accordance with what is suggested in the literature, it is proposed here to use a MAC-function (keyed hash function) instead of a keyless hash function so as to provide independence of confidentiality and integrity protection. The impacts on message formats and computation efforts are minor. In protection mode 1, the message even gets somewhat shorter. However, separate keys will be needed for confidentiality and for integrity. The current solution uses only one key for the encryption function. We propose to compensate for this by using the same key for both directions, and distinguish the directions by the sending PLMN Id in the integrity-protected part of the message, without a reduction of the security level. This latter proposal is, however, independent of the rest. We also point out open issues to be resolved.

---

1. Introduction

MAP security layer III, as described in [1] 3G TS 33.102 v3.4.0 (UMTS Security Architecture) uses a well-known method to provide integrity using an encryption function and a hash function. This method has been described and analysed in the literature. We refer to [2, section 9.6.5.] and take up suggestions found there.

The method used in [1] takes two forms, depending on whether integrity alone is required (protection mode 1, described in [1, section 7.4.2.2]) or whether both confidentiality and integrity are required (protection mode 2, described in [1, section 7.4.2.3]).

According to [1, section 7.4.2.2], the message body of Layer III messages in protection mode 1 takes the following form:

Cleartext‖TVP‖$E_{KSXY(i)}$(*Hash*(MAP Header‖Security Header‖Cleartext‖TVP))

In other words, a message x (consisting of MAP Header‖Security Header‖Cleartext‖TVP) is followed by an integrity check value which is computed as E(*Hash* (x)) where E is an encryption function, giving (x, E(*Hash*(x)). This correponds to [2, 9.86 Remark 1].

According to [1, section 7.4.2.3], the Layer III Message Body in protection mode 2 takes the following form:

$E_{KSXY(i)}$(Cleartext‖TVP‖*Hash*(MAP Header‖Security Header‖Cleartext‖TVP))

---

[1] This document is based on work carried out in the EU-sponsored collaborative research project USECA (http://www.useca.freeserve.co.uk/). Nevertheless, only the author is responsible for the views expressed here.

In other words, a message x (consisting of MAP Header||Security Header||Cleartext||TVP) is followed by a hash value *Hash* (x) which is then encrypted with the encryption function E, together with a part x' of the message x, giving E(x', (*Hash*(x)). This correponds to [2, formula (9.2)].

2. Properties

(1) Clearly, in both protection modes 1 and 2, integrity depends on encryption. The integrity protection is only as strong as the encryption function, and when the use of an encryption function is not possible then also integrity protection is not possible, i.e. not even protection mode 1 is then possible. This is undesirable.

(2) The method used in [1] requires that the hash function be collision-resistant when used with protection mode 1. Otherwise, an attacker could substitute one message for another having the same hash and hence the same integrity check value. Due to the birthday paradoxon, the output of collision-resistant hash functions needs to be twice as long as that of a keyed hash function for the same security level (i.e. chance of finding collisions or forging a MAC respectively).

(3) It is mentioned in [2, 9.86 Remark 1.] that a key K used as for protection mode 1 must be exclusively reserved for this integrity function, and not be used for encryption also. Certain chosen-text attacks are mentioned, but it is not clear how they could be carried out in the context of MAP security as in [1]. However, if there was concern about such an attack then two different keys would be needed anyhow for the method used in [1], one for protection mode 1 and one for protection mode 2.

3. Proposal

It is proposed to choose *Hash* to be a MAC-function *H* (keyed hash function) and to use two different keys, a key KSXY(int) to be used with the MAC-function *H* and a key KSXY(con) to be used with the encryption function E (cf. [2, section 9.6.5 (iii)]. The MAC-function H and the encryption function E should be independent. Clearly, when *H* is a MAC-function then, for protection mode 1, the use of the encryption function becomes superfluous.

It is proposed in addition, to use the same key for both directions, and to include the Sending PLMN Id in the security header which is contained in the integrity-protected part of the message. This makes up for the need for separate integrity and encryption keys, without weakening security. If it was decided to keep the concept of separate keys for each directon this would not affect the rest of the proposed changes.

Including the Sending PLMN Id in the security header is useful anyhow because the receiving entity needs to know the sending entity before being able to decrypt. It is against the principles of protocol design and may create technical problems if the sending entity had to be determined from a lower protocol layer.

We also propose to put the time variant parameter TVP first in the integrity-protected message, in accordance with [1, section 6.5.3]. The TVP does not need to be confidentiality-protected. It does not harm to do it from a security point of view, but there may be a performance penalty: Replayed packets would have to be first decrypted before they could be discarded. It is the task of the initialisation vector, not the TVP, to provide sufficient variety in the initial part of the encrypted message.

Remarks pertaining to data types like OCTET strings etc. are proposed to be removed from the text because they belong in a stage 3 description.

The remark about the compatibility of protection mode 1 with the current MAP protocol is not accurate because of the presence of the security header. It is therefore proposed to remove it.

We therefore propose to replace [1, section 7.4.2.2] with the following:

7.4.2.2 Protection Mode 1

The message body of Layer III messages in protection mode 1 takes the following form:

TVP‖Cleartext‖ $H_{KSXY(int)}$(TVP‖ MAP Header‖Security Header‖Cleartext)

where "Cleartext" is the message body of the original MAP message in cleartext. Therefore, in Protection Mode 1 the Layer III Message Body is a concatenation of the following information elements:

- Time Variant Parameter
- Cleartext
- Integrity Check Value

Authentication of origin and message integrity are achieved by applying the message authentication code (MAC) function *H* to the concatenation of Time Variant Parameter TVP, MAP Header, Security Header and Cleartext.

[Note1: There is need for replay protection of Layer III messages; it is envisaged to use TVP for this purpose. The precise definition of the use of TVP is ffs.]

We propose to replace [1, section 7.4.2.3] with the following:

7.4.2.3 Protection Mode 2

The Layer III Message Body in protection mode 2 takes the following form:

$$TVP \| E_{KSXY(con)}(Cleartext \| H_{KSXY(int)}(TVP \| MAP\ Header \| Security\ Header \| Cleartext))$$

where "Cleartext" is the original MAP message in cleartext. Message confidentiality is achieved by encrypting cleartext, TVP and integrity check value with the confidentiality session key KSXY(con). Authentication of origin and message integrity are achieved by applying the message authentication code (MAC) function *H* to the concatenation of Time Variant Parameter TVP, MAP Header, Security Header and Cleartext.

[Note1: There is need for replay protection of Layer III messages; it is envisaged to use TVP for this purpose. The precise definition of the use of TVP is ffs.]

We propose to replace [1, section 7.4.3] with the following:

7.4.3 Structure of Security Header

The security header is a sequence of the following data elements:

- Protection Mode
- Key Identifier
- Algorithm Identifier
- Mode of Operation
- Initialisation Vector
- Sending PLMN Id

NOTE: Whether the Initialisation Vector is needed depends on the mode of operation of the encryption algorithm.

4. Open issues

The type, length and use of the TVP is tbd. The requirements are not fully clear yet, especially concerning the necessary length to prevent a wrap around and a suitable window size at the receiver. Possible solutions include sequence numbers and windows (e.g. IPSec uses 32 bit sequence numbers and >32 bit windows) or a mix of time-stamps

(UTC) and nonces where the nonces are used to distinguish between messages with the same time stamp. (e.g. ITU H.235 uses a 64 bit TVP of this structure.)

Security association for layer III: It is necessary for interoperability to agree on the definition of a security association which has to be negotiated in layer I and transmitted to the network entities in layer II. Parameters will include addresses, keys, protection modes, operation modes and algorithms.

5. Evaluation

Message structure: The overall message structure is preserved. So, the influence on ongoing work in 3GPP CN should be minimal.

Computational effort: A common realisation of a MAC-function is the H-MAC which requires two applications of a hash function H (cf. e.g. [3, 9.67]). In protection mode 1, the additional application of the hash function is compensated for by the fact that encryption need no more be applied. In protection mode 2, an additional application of the hash function would be required indeed if H-MAC was used. The application of hash functions is fast, so this is considered acceptable.

Message lengths: In protection mode 1, the message becomes actually shorter for the reason mentioned in section 2 (2).(A typical value for the saving would be 80 bits.) In protection mode 2, the message length becomes also shorter or remains the same, depending on whether the same or a different hash function was meant to be used in [1] for protection modes 1 and 2.

Key management: When our suggestion is accepted to use the same keys for both directions (together with the inclusion of the Sending PLMN Id in the security header) then the overall length of the key management messages in layers 1 and 2 remains the same. But even if the number of key bits to be established in layer I and transported in layer II would double this would still be considered acceptable given the overhead of the messages in these layers.

Conclusion

The additional effort, if any, caused by our main proposal to separate confidentiality and integrity, is quite small, as shown in section 4. Other disadvantages cannot be seen. The separation of integrity and confidentiality could prove quite useful in certain scenarios not all of which can be foreseen today. The other changes are meant to clarify issues. It is therefore concluded that the proposals made in section 3 should be accepted by 3GPP SA3. The open issues remain to be dealt with.

References

[1] 3G TS 33.102 v3.4.0 (UMTS Security Architecture)

[2] A. Menezes, P. van Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRC Press, Boca Raton, 1997.

3GPP TSG SA WG3 Security — S3#13 **S3-000355**

23-26 May, 2000

Yokohama, Japan

(Ad-hoc meeting on MAP Security, Yokohama, 23 May, 2000)

Source: **Motorola**

Title: **Layer III MAP Message Body in Protection Mode 2**

Document for:

Agenda Item:

---

1. Proposal

We propose that in protection mode 2 of layer III MAP secuirty, we encrypt first and then add a MAC for integrity. The reason for doing it in this order is that integrity can be checked without the need to decrypt first, so a false MAP message can be discarded with much less computation. This more efficient integrity protection provides a degree of protection against denial of service attack by flooding a node with false MAP messages.

We propose to replace TSGS3-000312 with the following:

7.4.2.3 Protection Mode 2

The Layer III Message Body in protection mode 2 takes the following form:

TVP|| $E_{KSXY(con)}$(Cleartext)|| $H_{KSXY(int)}$(TVP|| MAP Header||Security Header||Ciphertext)

where "Cleartext" is the original MAP message in cleartext. Message confidentiality is achieved by encrypting cleartext, TVP and integrity check value with the confidentiality session key KSXY(con). Authentication of origin and message integrity are achieved by applying the message authentication code (MAC) function *H* to the concatenation of Time Variant Parameter TVP, MAP Header, Security Header and Ciphertext. The integrity is performed on the encrypted message so that integrity can be checked before decryption.

[Note1: There is need for replay protection of Layer III messages; it is envisaged to use TVP for this purpose. The precise definition of the use of TVP is ffs.]

3GPP TSG SA WG3 Security — S3#13 **S3-000368**

23-26 May, 2000

Yokohama, Japan

Source: **Siemens AG / Vodafone Airtouch**

Title: **Replay protection for core network signalling messages**

Document for: **Decision**

Agenda Item:

1 Introduction

This contribution considers mechanisms for providing replay protection of core network signalling messages. It concludes that perfect protection against replay is difficult to achieve. The use of a time-stamp as time variant parameter (TVP) is considered a feasible option to provide a reasonable level of replay protection. It is also pointed out that encrypting messages, as in protection mode 2, makes it more difficult for an attacker to mount a replay attack.

2 Replay protection using TVP

In section 7 of 33.102 v3.4.0 on core network signalling security, a field is reserved in the layer III message structure for a time-varying parameter (TVP). This parameter is intended to be used as part of the integrity protection mechanism to provide *replay protection*. However, neither the length, type nor use of this field is specified. At least the length of this field must be determined to allow N4 to complete the stage 3 specifications in 29.002.

In order to determine a suitable length, or range of lengths, for TVP, it is necessary to consider the types of time-varying parameters that may be used. Possible solutions include sequence numbers and windows (e.g. IPSec uses 32 bit sequence numbers and >32 bit windows), time-stamps (UTC), or a mix of time-stamps and sequence numbers where the sequence numbers are used to distinguish between messages with the same time stamp. (e.g. ITU H.235 uses a 64 bit TVP of this structure.)

With the current key management architecture for core network signalling security, the same key may be used between many different pairs of communicating network nodes. As a result, a general form of replay protection would imply that we guard against an attacker recording any message protected under a given integrity key and then replaying it towards any receiver that accepts messages protected under the same key.

The basic requirement is that each message transmitted under the same integrity key would need to contain some information which allows the receiver to test the integrity verified message for freshness, i.e. that it has not previously been accepted as fresh. This requirement can be met by ensuring that each message includes a nonce such as a time-stamp or sequence number. The TVP is therefore used, potentially with other information in the message, to form a nonce which can be checked by the receiver.

2.1 Use of sequence numbers

One way of ensuring that each message contains a nonce would be to generate a unique sequence number for each message from a single global counter shared by all sending nodes. All receivers would then need to maintain a shared counter containing the highest sequence number previously accepted as being fresh. An alternative solution, which avoids the need for a single global counter, would be to make the sequence number unique per sending entity. This could be done by assuming that the message contains a unique identifier for the sending node (i.e. within individually protected MAP message components). The sending nodes would then generate sequence numbers from individual, local counters and the receiving nodes would maintain individual, local values of the highest

sequence number previously accepted for each sending entity. Thus, both sending and receiving nodes must store state information, with receiving nodes storing independent state information per sending node.

Furthermore, it may be conceivable that messages will arrive at the receiver out-of-order. If this is the case then receiving nodes must support a window or list mechanism which must be managed per sending entity. This further complicates the sequence number management scheme.

To avoid wrap around, the sequence number must be sufficiently larger than the maximum number of messages that may be sent between each sending and receiving node during the lifetime of a particular integrity key. Thus the expected lifetimes of the integrity key and signalling traffic estimates will determine the required length of the sequence number which must be transported within TVP.

2.2 Use of time-stamps

Another way of ensuring that each message contains a nonce would be to use a time-stamp taken from a global time source such as UTC. In this case the receiving node would check the freshness of the time-stamp by referring to a local time source which is sufficiently synchronised with the sender's time source. In order to ensure that the receiving node can determine the freshness of the message using the time-stamp, the resolution of the time source must be sufficiently large such that sequential messages sent *by any network node* using the same integrity key are not protected using the same time-stamp. Again, this time-stamp could be made unique per sending node by assuming that the message contains a unique identifier for the sending node (i.e. within individually protected MAP message components). This would allow the resolution to be sufficiently large such that sequential messages sent *by an individual network node* using the same integrity key are not protected using the same time-stamp. Thus the length of the time-stamp will depend on both the expected lifetimes of the integrity key and the required time resolution.

The receiving entity will accept a message only if the time-stamp is within a certain time-window. The size of the time-window will depend on the degree of synchronisation which may be assumed for the clocks at the sending and the receiving nodes, and the expected transmission delays of the messages.

To avoid wrap around the time-stamp must not exceed the maximum lifetime of the key. Furthermore, the time resolution must be high enough. Thus, the length of the time-stamp will depend on both the key lifetime and the required time resolution. The time resolution determines the window of opportunity during which an attacker can mount a replay attack. A replay attack will still be quite difficult to mount successfully if the window of opportunity is sufficiently small even if several messages within that window share the same time-stamp. This means that a reasonable degree of replay protection can still be provided if the window of opportunity is sufficiently small.

2.3 Evaluation of TVP alternatives

It is suggested that a solution using time-stamps is the most feasible approach in this particular application. However, because of synchronisation requirements, it is not expected that a time-stamp with a resolution is attainable which would provide for perfect replay protect, i.e. to ensure that no two messages do not have the same time-stamp. Therefore, it is worth considering the level of protection against replay that can be achieved using the maximum attainable clock resolution/synchronisation.

If we assume that a clock unit of the order of 1 second is attainable, an attacker may record and replay a message from any sending entity to any receiving entity that uses the same integrity key, within any one second period. However, if we assume that certain contextual information in the message is always used to calculate the integrity check code (i.e. information within the MAP message components) and that only certain sequences of integrity protected messages are accepted by the receiver, then it seems reasonable to conclude that it may be highly unlikely that an attacker can exploit the fact that some messages are protected using the same time-stamp. However, further analysis is required before an estimate of this probability can be obtained.

3 Replay protection using encryption

Until now we have assumed that replay protection is provided using the integrity protection mechanism alone, i.e. Protection Mode 1. However, we have seen that it is difficult to provide a perfect replay protection mechanism as part of the integrity protection mechanism. A way of enhancing replay protection would be to exploit the fact that the message can also be encrypted when Protection Mode 2 is applied. Indeed, if the content of the message is not known to the attacker it is much more difficult

for the attacker to mount a replay attack. However, we would like to point out that encryption in itself does not provide sufficient replay protection because an attacker may learn the content of an encrypted message sent in the past (e.g. due to a security breach at a node) and then replay it later.

4 Conclusion

It is recommended to use time-stamps as TVPs for replay protection. It is further recommended to use protection mode 2 whenever possible as this makes replay attacks more difficult. As regards the size of TVPs, it is proposed that 32 bits is sufficient. This allows for a time-stamp based scheme with a maximum time resolution of 1 second and a maximum key lifetime of more than 100 years. The resolution of the clock must be agreed as a system parameter, the size of the time-window at the receiving node need not be standardised.