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.