3GPP TSG SA WG3 Security — S3#21 **S3-010635**

27 - 30 November, 2001

Sophia Antipolis, France

Source: Siemens Atea

Title: Protection Profiles Version Identification

Document for: Discussion / Decision

Agenda item: MAP security

Abstract

This contribution analyses how new ApplicationContexts/operations may be integrated into the existing or new MAP-PG of upcoming 3GPP releases. A certain inflexibility is observed with the current mechanism, and therefor it is proposed to include a protection profile version indication (PPVI) into the MAP SA to overcome this inflexibility.

1 Introduction

At SA3#20 in Sydney, Hutchison 3G introduced TD S3-010492 on ‘flexibility of MAP protection Profiles’. S3-010492 proposed to introduce a “catch-all” MAP-PG in order to cope with future security breaches. The discussions at this meeting triggered the question, how to handle future MAP-PG changes.

This contribution discusses the effects that upgraded or new AC have on Protection Profiles.

2 Introduction on MAP-AC and MAP-PG

Before starting the analysis in the next paragraph, some basic definitions are repeated.

- Each Application Context Name (AC) evolves independently in various versions.

E.g. the LocationUpdate context was raised from v2 to v3 in GSM phase 2+ Release 96 due to the introduction of CAMEL. On the other hand Reset is still v2 since there has not been any requirement for a version upgrade since GSM phase 2 (Release 95). Another example is that the InfoRetrieval context was raised from v2 to v3 in Release 99 due to introduction of quintets.

- Backward compatibility and TS 29.002 compliance.

Each MAP-NE node that supports an application context in version n, shall also support this application context in version n-1 (if defined).

GSM and UMTS networks that are fully compliant to TS 29.002 Rel-4 have to support all AC from the Specification. The AC (n-i) versions are needed for interworking with older network versions, which do support only the n-i versions.

- Protection Groups (MAP-PG)

The Protection Profiles are currently defined in Protection-Groups each of them contains all available applicationContext/operation combinations from earlier MAP-specifications.

Example:

InfoRetrievalContext-v3/ Send Authentication Info	3
InfoRetrievalContext-v2/ Send Authentication Info	3
InfoRetrievalContext-v1/ Send Parameters	3

- MAPsec Application Context

MAPsec is an application context (AC) of its own (the only version until now is v3, but it may evolve to v4 in Rel-5 or later if e.g. the security header needs to be changed). Messages within this MAPsec AC basically consist of containers that may carry (in protected mode) any other MAP AC in any version.

Therefor it is possible that a MAP-NE implements an AC n-1 but not the highest n, together with a MAPsec AC n and n-1 as the concept for MAPsec is independent of the AC-version of the protected messages. The MAPsec AC (n-1) is able to tunnel AC of an higher version (n). In other words, there are no effects of the definition of new (non MAPsec) AC on MAPsec AC.

3 How to handle changed Protection Profiles.

Currently the MAP-PP's are a combinations of functional groups (MAP-PG). If in future several AC’s are augmented (n+1), then the groups are in principle affected. By means of examples, it is explained how new application Contexts could affect the MAP-PG and MAP-PP.

3.1 New Application Contexts are introduced in a higher Release and need protection.

Suppose that InfoRetrievalContext-v4 and HandoverControlContext-v4 are newly defined in Rel-6.

There exist several approaches to include these 2 AC into the list of MAP-PG:

Approach-1: Define a new Protection Group to include the new application context

The table gives the example of a new PG(5) to accommodate the new AC/operation on InfoRetrievalContext-v4 (PG(5) is a superset of PG(2)). A new PG(6) for HandoverControlContext-v4 is not included in the table.

Table 1: Protection Profile Definition A

Protection profile name	**Protection group**					
	PG(0) *No protection*	PG(1) *Reset*	PG(2) *AuthInfo except handover situations*	PG(3) *AuthInfo in handover situation*	PG(4) *Non-location dependant HLR data*	PG(5) *Rel-6 AuthInfo except handover situations*
Profile A	✓					
Profile B		✓	✓			
Profile C		✓	✓	✓		
Profile D		✓	✓	✓	✓	
Profile E		✓	✓		✓	
Profile F		✓				✓
Profile G		✓		✓		✓
Profile H		✓		✓	✓	✓
Profile I		✓			✓	✓
Profile J						

PG(5) would supersede PG(2) from Rel-6 on and rendering the PG(2)-bit with fixed interpretation forever (PG(5) and PG(2) are to be used exclusively). In addition to an explosion of the used MAP-PG, this may lead to an exhaustion of the 16 available profile bits in the long term.

Conclusion: This solution is not favorable, as it does not only require extra administration, but also cause a slower introduction of the new protection in the network. Regardless of whether ‘fallback to unprotected mode’ is allowed or not, a KAC can only accept a PPI when all MAP-NE within its PLMN that support MAPsec, also support that new PPI.

Approach-2: Include them into the existing Protection Groups (where they functionally belong)

PG(2) and PG(3) are affected in Rel-6 but the protection profile name and bit-string is still the same, hence with another meaning.

Table 2: Protection Profile Definition B

Protection profile name	**Protection group**				
	PG(0) *No protection*	PG(1) *Reset*	PG(2) *AuthInfo except handover situations*	PG(3) *AuthInfo in handover situation*	PG(4) *Non-location dependant HLR data*
Profile A	✓				
Profile B		✓	✓		
Profile C		✓	✓	✓	
Profile D		✓	✓	✓	✓
Profile E		✓	✓		✓

From the KAC point of view the same protection profile string is negotiated. The KAC does not interpret the bit-string on the level of ApplicationContext (AC) but only performs a comparison against a policy database.

The MAP-NE however does interpret the string when handling the MAPsec payload (PPI dependant cryptographic processing). A MAP-NE of Rel-5 will never receive the new AC in a MAPsec container, because TCAP will abort the connection with error reason ‘UnsupportedEncapsulatedAC before the

MAPsec processing can take place. Upon receiving this error, the Rel-5 MAP-NE initiates an application Context fallback. 2 Rel-6 MAP-NE will be able to send the new AC within MAPsec immediately after they are taken into service.

Conclusion: This profile upgrade method works and requires no changes to the existing handling. Higher Application ContextVersions will be transported by MAPsec when they are used.

3.2 Adding a new MAP-PG for a new release n+1 that includes AC of current version n.

Suppose a new MAP-PG(5) is defined in Rel-6 as a functionally different group of existing ApplicationContexts that exists in Rel-5. The table of protection profiles is similar as in Table 1 with the difference that PG(2) can be combined with PG(5).

Each MAP-NE that is upgraded to Rel-6 shall support the new MAP-PG. As long as the MAP-NE is still of Rel-5, it is not obliged to support the new MAP-PG. This may even be impossible when Rel-5 did not require that MAP-NE to implement MAPsec. Regardless of whether ‘fallback to unprotected mode’ is allowed or not, a KAC can only accept a PPI when all MAP-NE within its PLMN that support MAPsec, also support that new PPI.

3.3 Adding an existing AC to a MAP-PG in version n+1.

It is obvious that approach-2 cannot be chosen for including existing AC, as this could cause a misunderstanding between a MAP-NE of Rel-n and n+1. They both interpret the PPI in another way. The example further details this.

Suppose MAP-PG(2) is defined in Rel-6 as including 1 AC more (Called AC-X within this paragraph). The protection profiles that include MAP-PG(2) have another meaning, but the same bit-string.

Table 3: Protection Profile Definition C

Protection profile name	**Protection group**				
	PG(0) *No protection*	PG(1) *Reset*	PG(2) *AuthInfo except handover situations*	PG(3) *AuthInfo in handover situation*	PG(4) *Non-location dependant HLR data*
Profile A	✓				
Profile B		✓	✓		
Profile C		✓	✓	✓	
Profile D		✓	✓	✓	✓
Profile E		✓	✓		✓

From the KAC point of view the same protection profile string is negotiated. The KAC does not interpret the bit-string on the level of ApplicationContext (AC) but only performs a comparison against a policy database.

The MAP-NE however does interpret the string when handling the MAPsec payload (PPI dependant cryptographic processing). All MAP-NE, whether Rel-6 or Rel-5 can handle the PPI. A MAP-NE Rel-6 will send the AC-X protected, which will be rejected by MAP Rel-5. A MAP-NE Rel-5 will send AC-X unprotected which will be rejected by MAP-NE Rel-6.

Conclusion: Adding/Removing an AC from an existing profile by changing the semantics of the PPI-bits, cannot be done (approach-2). The alternative approach to define a new MAP-PG for this, is feasible but has the disadvantages as described in approach-1 of clause 3.1 .

4 Conclusion: a Protection Profile Version Identifier (PPVI) is needed.

In the previous chapter, two approaches were analyzed. Only in 1 specific case the use of a semantics change in the Protection Profile Identifier could be used to avoid new MAP-PGs. The Protection Groups are therefor in a certain sense fixed from Rel-4 on and AC can only be included in new MAP-PG's. Defining new PG's will exhaust the number of bits.

To be future proof, a Protection Profile Version Identifier (PPVI) is proposed. The PPI will therefor become version dependent. This will allow SA3 to redefine MAP-PG in future 3GPP releases when required and in a flexible way. A Rel n+1 MAP-NE (and KAC n+1), shall support all PPI of previous Versions. Also the MAP DoI needs to include this PPVI during negotiation, as this becomes part of the MAP-SA.