



IPX network End to end Security Guidelines

DRAFT Version 1.0

17 Nov 2017

Security Classification: Non-Confidential

Access to and distribution of this document is restricted to the persons permitted by the security classification. This document is confidential to the Association and is subject to copyright protection. This document is to be used only for the purposes for which it has been supplied and information contained in it must not be disclosed or in any other way made available, in whole or in part, to persons other than those permitted under the security classification without the prior written approval of the Association.

Copyright Notice

Copyright © 2017 GSM Association

Disclaimer

The GSM Association ("Association") makes no representation, warranty or undertaking (express or implied) with respect to and does not accept any responsibility for, and hereby disclaims liability for the accuracy or completeness or timeliness of the information contained in this document. The information contained in this document may be subject to change without prior notice.

Antitrust Notice

The information contain herein is in full compliance with the GSM Association's antitrust compliance policy.

Table of Contents

1	Introduction	3
1.1	Overview	3
1.2	Scope	3
1.3	Definitions	4
1.4	Abbreviations	4
1.5	References	5
1.6	Conventions	5
2	Background	6
2.1	Problem Statement	6
2.2	Lessons Learned	6
2.3	Protocols used in different network generations (2G, 3G, 4G, 5G)	8
2.4	Requirements	9
3	What to Protect	9
3.1	Interfaces to be protected	9
3.2	Fields that require confidentiality or integrity protection	9
3.3	AVP protection scheme for LTE	10
4	Where to protect (LTE)	11
4.1	Secure perimeter (LTE)	11
4.2	Secure perimeter in case of intermediate changes (LTE)	12
4.3	Exceptions (LTE)	12
4.3.1	Secure perimeter when delegating security functions to the IPX Provider	12
4.3.2	Secure perimeter in case of LTE roaming hub	14
5	How to protect	15
5.1	Providing Integrity	15
5.1.1	Common approach	15
5.1.2	Proposed approach (Extended Signature Scheme)	16
5.1.3	Proposed approach with intermediate changes (Extended Signature Scheme)	17
5.1.4	Handling deletions (Extended Signature Scheme)	19
5.1.5	Handling additions (Extended Signature Scheme)	21
5.1.6	Handling multiple operations (Extended Signature Scheme)	21
5.1.7	Contents of the SIGNATURE-FIELD	22
5.2	Providing confidentiality	22
5.2.1	Common Confidentiality Model	22
5.3	Relation between providing confidentiality and integrity	27
Annex A	Document Management	28
A.1	Document History	28
A.2	Other Information	28

1 Introduction

1.1 Overview

This document provides guidelines on how Diameter and future protocol signalling exchanges within the Internetwork Packet Exchange (IPX) ecosystem should be protected in order to prevent spoofing, man-in-the-middle-attacks and eavesdropping.

1.2 Scope

The document provides guidelines on *what* to protect, *where* to protect and *how* to protect the exchange of messages within the IPX ecosystem. Therefore the scope is limited to inter-network exchange between service providers only. Parts of the solution or solution directions however may be used for intra-network messages within service providers too. The scope is limited to provide guidelines on providing integrity, authentication and confidentiality only. The document does not outline the risk classification of AVPs (or other fields in future protocols) and does not recommend which AVPs to expect within certain circumstances. The general security guidelines for IPX providers and service providers with regards to screening should still be taken into account. This document only aims to strengthen the security within the IPX ecosystem by adding measures to protocol itself.

With the introduction of 5G, integrity and confidentiality protection must be built in to the protocol, as “security by design” usually the only real chance to enforce security measures. After implementation the window of opportunity is usually closing rapidly as adoption begins. The focus of this document and corresponding workgroup has gradually moved to 5G as there is (as of November 2017) limited time remaining during which 5G standardisation work can be influenced.

Although initially the working assumption was that Diameter was the most likely candidate for the 5G core network protocol, there is now a move expected to a service based architecture with Hyper-Text Transfer Protocol (HTTP) objects and JavaScript Object Notation (JSON) object notation, so the use of Diameter becomes more unlikely. This document might still contain references to Diameter structure and AVPs – these can be interpreted as the more general term “message” for Diameter message and “field” for AVP.

The DESS group will not lose focus on a Diameter based implementation for Long Term Evolution (LTE) but will give the –draft- ideas for 5G priority.

1.3 Definitions

Term	Description
Integrity	A message is positively integrity checked if the service provider operator receiving the message can determine if the message has not been changed between the sender and the receiver of the message and if the service provider receiving such message is able to detect a legitimate change of an AVP or field by an intermediate node such as an IPX. (if any)
Authenticity	A message is authentic if the receiving service provider of the message can determine if the service provider sending the message is the one that is stated in the "Origin-Realm". If intermediate changes to fields of the message are applied in transit by IPXs, the receiving service provider must be able to determine the initiator of such modifications.
Confidentiality	A field is confidentiality protected when such a field is encrypted between the sending and the receiving service provider. Integrity may be applied on top of encryption.
Service provider	Mobile network operator (MNO), fixed network operator or other type of operator connecting to inter-service provider IP backbone for roaming and/or interworking purposes.
IPX provider	A provider that operates a part of the IPX network and that offers IPX services.
Replay attack	A replay attack is a type of network attack in which a valid message (or part of the message) is maliciously or fraudulently repeated or delayed.

1.4 Abbreviations

Term	Description
3GPP	Third Generation Partnership Project
AVP	Attribute Value Pair
CAP	CAMEL Application Part
DEA	Diameter Edge Agent
DESS	Diameter End-to-End Security
DHE	Diffie-Hellman Key Exchange
E2E	End-to-end
ECDHE	Elliptic Curve Diffie- Hellman Key Exchange
GTP-C/U	GPRS Tunnelling Protocol Control/User Plane
HTTP	Hyper-Text Transfer Protocol
IMS	IP Multimedia Subsystem
IPSec	Internet Protocol Security
IPX	Internetwork Packet Exchange
JSON	JavaScript Object Notation
LTE	Long Term Evolution
MAP	Mobile Application Part
MNO	Mobile network operator
PRD	Permanent Reference Document

Term	Description
SCCP	Signalling Connection Control Part
SGW	Security Gateway
SIP	Session Initiation Protocol
SS7	Signalling System No. 7
TCP	Transport Control Protocol
TDM	Time Division Multiplexing
TLS	Transport Level Security

1.5 References

Ref	Doc Number	Title
[1]	RFC 2119	Key words for use in RFCs to Indicate Requirement Levels
[2]	3GPP TS 33.210	3G Security; Network Domain Security; IP Network Layer Security
[3]	IR.21	GSMA Roaming Database IR. 21 Data
[4]	GSMA PRD IR.80	Technical Architecture Alternatives for Open Connectivity Roaming Hubbing Model
[5]	GSMA PRD IR.88	LTE and EPC Roaming Guidelines
[6]	draft-korhonen-dime- e2e-security-03.txt	J. Korhonen and H. Tschofenig, "Diameter AVP Level Security: Keyed Message Digests, Digital Signatures, and Encryption," Internet Engineering Task Force, 2016

1.6 Conventions

"The key words "must", "must not", "required", "shall", "shall not", "should", "should not", "recommended", "may", and "optional" in this document are to be interpreted as described in RFC2119 [2]."

2 Background

2.1 Problem Statement

Third Generation Partnership Project (3GPP) Diameter messages that are exchanged between service providers within the IPX ecosystem do not have any native integrity or confidentiality protection measures within the protocol. To secure 3GPP Diameter messages, 3GPP technical specifications reference 3GPP TS 33.210 [5], where IPSEC/TLS is suggested.

The usage of Internet Protocol Security (Ipssec/Transport Level Security (TLS) however does not provide an end to end security within the IPX ecosystem but only a hop by hop security. The IPSEC/TLS tunnel shall be established with the next hop in order to permit the correct message routing and this limit represents a problem due to the following two reasons.

First, due to this hop by hop” routing nature of the Diameter protocol, a Diameter response always follows the same path as the Diameter request, making the Diameter protocol a “spoofing friendly” protocol.

Second, the threat landscape is changing rapidly; attackers will get more interest in the Diameter protocol. For instance, if it gets harder to obtain the IMSI of a subscriber via Signalling System No. 7 (SS7), attacks will move to Diameter to reach their goals. Additionally the metadata is still present in the Diameter protocol which allows mass surveillance and location tracking of 4G/5G subscribers. End-to-end (E2E) encryption using a mobile application does not protect the subscriber against such tracking. This is yet another good reason to protect confidentiality and integrity of the Diameter protocol.

The screening measures mentioned in IR.88 [2] are sensible but do not provide the mitigation level necessary in the current threat landscape. Screening measures such as comparing the Origin-Realm with the point of ingress are in the hands of IPX providers, while the assets that attackers are after are mainly with the service providers. Service providers must be able to determine the authenticity and integrity (and apply confidentiality on a per field basis if required) “as it happens”.

It is difficult if not impossible to use standard protocols in a standard way, including the work described in [2], with the IPX ecosystem as IPX providers making changes to fields, including adding and deleting fields. The solution should cope with this requirement. In other words, the receiving service provider must not only be able to unambiguously determine the sender of a message and judge the integrity of the message, it should also be able to track if any intermediate node has changed contents and subsequently determine if this was a legitimate change.

While it is broadly understood that the IPX ecosystem for LTE needs security solutions for Diameter, this document focusses also on security solutions for 5G in order to have security by design included as part of 5G standardisation.

2.2 Lessons Learned

Legacy GSM networks use SS7 to manage location updates, subscriber profile information exchange etc. In LTE networks, MNOs currently rely on Diameter for this, but essentially they both perform the same tasks and they do it in a very similar way. The MAP signalling in

SS7 is almost entirely replicated on Diameter interfaces such as S6a/d (except the voice services which were migrated to IP Multimedia Subsystem (IMS)). While the mechanics of location update and subscriber information exchange work effectively, there were fundamental flaws built into the SS7 and Diameter protocols themselves. In SS7 we have a protocol without authentication, integrity protection and confidentiality of signalling sources, so any entity on the SS7 network is considered a trusted entity. There were efforts made to rectify this design problem with the introduction of a new node SGW (Security Gateway) but it never got any widespread recognition. Other attempts to remediate the issues in SS7 included the development of TCAPsec and TCAP Handshake to provide authentication, however they too were largely left unimplemented. Many years later we are faced with a situation where most security-aware mobile networks are deploying SS7 firewalls to safeguard against these signalling vulnerabilities, and practically all networks are performing some kind of hardening either through configuration changes or either stateful or stateless filtering.

Diameter incorporates a higher level of security from a design perspective, but many of the same flaws persist. Ipsec is recommended for Diameter networks, but the implementation only goes so far as to mandate hop-by-hop Ipsec, again leaving room for carriers to make intermediate changes to signalling traffic. However, Ipsec provides little or no security from an attacker with access to impersonate and inject traffic via the IPX. As there is no end-to-end tunnel established between peers we find yet again, just as in SS7, that anyone can impersonate anyone on the signalling network. And to make matters worse the hop-by-hop routing scheme in Diameter enables an attacker to perfectly impersonate a trusted peer and receive responses on spoofed requests. In Diameter the Diameter Edge Agent (DEA) was intended to provide some degree of security for example through topology hiding, but once again we find that the initial approach was not sufficient to provide a secure protocol and mobile networks are now procuring Diameter firewalls or evolved DEAs in order to provide secure signalling (again lack of authentication, integrity protection and confidentiality between the networks).

The issues found in both SS7 and Diameter highlight the importance of getting things right from the outset and for 5G we hope to incorporate these lessons and move towards a proper security model where the particular needs of signalling can be incorporated.

2.3 Protocols used in different network generations (2G, 3G, 4G, 5G)

The following table summarizes the protocols used on the interconnection interfaces for different mobile network generations. Colours indicate the associated risks and vulnerabilities observed in the recent years related to these protocols.

Generation	Protocol	Used on interconnects	Authentication & Integrity protection	Confidentiality	Secured alternative	Adopted by industry
2G, 3G	SS7/SIGTRAN – MAP, CAP	Yes	None	None	MAPsec TCAPsec Asymmetric encryption (opensource SigFW)	No
2G, 3G	SS7/SIGTRAN ISUP/BICC	Yes	None	None	Not known	No
3G, 4G	TDM (media)	Yes	No	No	No	No
2G, 3G, 4G	GTP-C	Yes	None	None	Not known (DTLS?)	No
2G, 3G, 4G	GTP-U (user plane)	Yes	None	None	Not known (DTLS?)	No
4G	Diameter	Yes	None	None	NDS/IP ¹ DESS (this document), Asymmetric encryption (opensource SigFW)	No
3G, 4G	Session Initiation Protocol (SIP)	Yes	Yes (only from access segment)	No	SIPs, IPSec	Yes
3G, 4G	RTP (media)	Yes	No	No	RTPS	Yes
5G	TCP/HTTP2/JSON	Yes	?	?	?	?

Table 1 – Network generation overview – protocols used and secure protocol alternative

¹ NDS/IP is optional and is an end-to-end approach (confidentiality and integrity) with its drawbacks (no intermediate changes and full encryption)

2.4 Requirements

The following high level requirements have been developed within the GSMA for IPX end-to-end security:

1. The solution must:
 - a) provide mandatory to implement and mandatory to use authentication *and* integrity protection on a per field basis as described in chapter 1
 - i. including support of intermediate adding of fields by IPX providers
 - ii. including support of intermediate modification and deletion of non-encrypted fields by IPX providers
 - b) provide mandatory to implement and mandatory to use confidentiality protection of authentication vectors
 - c) provide mandatory to implement and optional to use confidentiality protection on other fields
 - d) provide mandatory to implement and mandatory to use protection against replay attacks
 - e) be operationally feasible e.g. acceptable procedure for key management
 - f) include (at a minimum) the 5G successor of the Diameter 3GPP interfaces described as Network-Network interfaces by IR.88 (e.g. S6a)

The solution should:

- g) impact a minimal amount of network elements
 - h) Have an acceptable performance and overhead
 - i) Support the ability to delegate security functionality to another entity.
2. The exact requirements for LTE will be specified at a later stage.

3 What to Protect

3.1 Interfaces to be protected

The aim is to protect Diameter interfaces where service providers interact with other service providers either directly or via the IPX network. This includes at minimum the following interfaces: S6a, S6d and Gy and their corresponding interfaces in 5G.

3.2 Fields that require confidentiality or integrity protection

Fields that require confidentiality protection are fields where eavesdropping is a realistic threat *and* IPX providers have no need to inspect, add, modify or delete fields.

Fields that require integrity protection are AVPs where eavesdropping near the IPX providers' network is less of a realistic threat (hop-by-hop encryption with for instance Ipsec may still be applied).

The general rule of thumb is the following:

1. Encrypt fields that contain sensitive information and where IPX providers do not need to modify or inspect them

2. Protect the integrity of all fields with the expectation of Route-Record
3. Allow changes within the secure perimeter (as long as those can tracked in the to be designed protocol)
4. Do not allow changes to encrypted fields *and* Origin-Realm (to accomplish authenticity), Origin-Host, Destination-Realm and Destination-Host within the secure perimeter. See section “secure perimeter” in chapter 4 for the definition.

3.3 AVP protection scheme for LTE

For the classification of confidentiality and integrity protection of AVPs, the protection values are specified as part of the Diameter Risk Classification matrix / The matrix specifies per AVP the type of protection that applies to a specific AVP. This is an approach for LTE only. For 5G similar measures should be taken

The following list shows the allowed values of the security columns that classify the type of protection per AVP. The additional column “Semantics” clarifies the relationship between the values in the columns for the AVP protection scheme.

Confidentiality Protected	Integrity		Protected	Semantics
	Protected	Modifiable		
Yes	Yes	No	Not applicable	If “Confidentially Protected”, then an AVP becomes automatically “Integrity Protected Unmodifiable” Recommended default value
No	Yes	No	Not applicable	If “Integrity Protected”, an AVP may be “Modifiable” (includes addition and deletion) or “Unmodifiable”
No	Yes	Yes	Not applicable	
No	No	n.a.	No	If neither “Integrity Protected”, then an AVP becomes automatically “Not Protected”

Table 2 – AVP protection scheme

In case of grouped AVPs the following rules apply:

- If a higher layer AVP is “integrity protected unmodifiable” the layers below are “integrity protected unmodifiable”

Below are some examples of AVPs and the reasoning why the AVP belongs in a certain category.

AVP	Reasoning
1448 XRES	The AVP contains sensitive key material and there is no use case to inspect or change the contents between the sending and the receiving service provider. The AVP falls in the category “ Confidentiality and

AVP	Reasoning
	integrity Protected ".
296 Origin-Realm	Integrity protecting the Origin-Realm provides authenticity. Therefore the AVP must be integrity protected and unmodifiable <u>within</u> the secure perimeter. The AVP falls in the category " Integrity Protected unmodifiable ". There are numerous examples where the Origin-Realm may be changed (LTE roaming hub, Origin-Realm correction, etc.). These changes occur <u>outside</u> the secure perimeter; refer to chapter 4 for more information.
1433 STN-SR	The AVP contains does not contain sensitive user information and IPX providers have a need to modify this AVP. The AVP falls in the category " Integrity Protected modifiable "
282 Route-Record	The AVP contains information about hops. Confidentiality protection is not appropriate since this AVP is used for routing in the Diameter base protocol. For the same reason integrity protection does not make sense, the amount route record AVPs change hop by hop resulting in every hop singing for changes which would drastically impact the performance. The AVP falls in the category " not protected "

Table 3 – AVP Example Categorisation

4 Where to protect (LTE)

The most logical place to protect confidentiality and integrity is at the edge of the network of the service provider. As the exact network topology for 5G is unclear, this chapter solely focusses on LTE.

It is important to limit the impact to a single network element; the most likely candidate is the DEA. Another good reason for implementing Diameter security at the DEA is that some MNOs implement topology hiding in their DEA. In doing this, the Origin-Host AVP is usually modified, and this may affect the result of an integrity check. A Diameter firewall is another element that could fulfil the function described in this document.

4.1 Secure perimeter (LTE)

The secure perimeter reaches from the edge of the sending entity to the edge of the receiving entity. In other words, the sending entity, usually a service provider, will apply integrity and/or confidentiality protection of AVPs for a receiving identity based on the edge of its network. The receiving entity, usually a service provider, then decrypts the confidentiality-protected AVPs and judges the integrity of the integrity-protected AVPs.

Scenarios for deriving the key to apply confidentiality and integrity vary slightly in case of symmetric or asymmetric-key cryptographic algorithms, but the Origin-Realm and Destination- Realm play an important role here.

The secure perimeter is outlined in Figure 1 below. The AVP protection is applied after topology hiding.

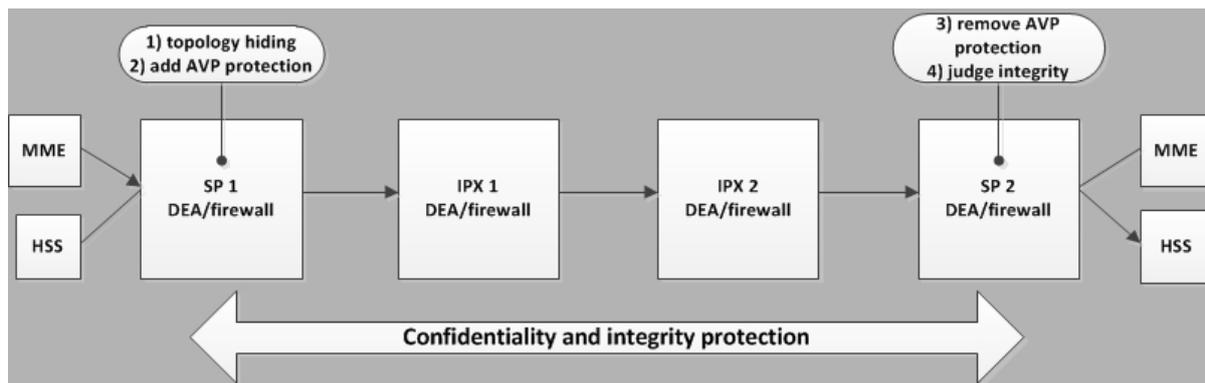


Figure 1 – Secure perimeter

Both the sending and receiving operator use the Origin-Realm and Destination-Realm to apply the correct keys.

4.2 Secure perimeter in case of intermediate changes (LTE)

IPX providers also make changes to AVPs without them necessarily having security functionality delegated to them. Figure 2 illustrates such a case, in which an IPX provider deletes an AVP (STN-SR)

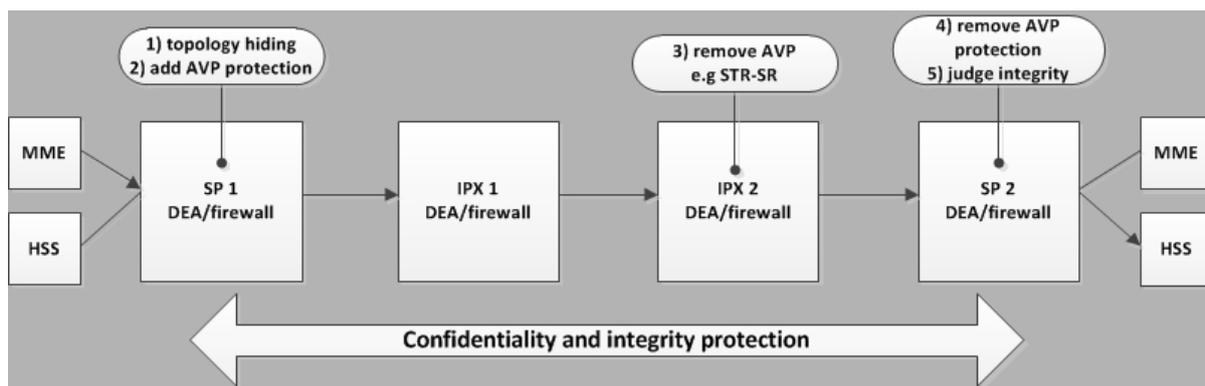


Figure 2 – Secure perimeter in case of intermediate AVP modifications.

The secure perimeter is in this case still between service provider 1 and service provider 2 even though IPX provider 2 removes an AVP. The protocol must make sure those changes can be detected. In other words, service provider 2 must be able to detect that IPX provider 2, removed the AVP STN-SR. It must also be able to detect that other AVPs are not changed, not between service provider 1 and IPX provider 2 and not between IPX provider 2 and service provider 2.

4.3 Exceptions (LTE)

This section describes a number of non-standard approaches that are not encouraged but could be implemented.

4.3.1 Secure perimeter when delegating security functions to the IPX Provider

The protection function may be delegated from the service provider to the IPX provider. There are a number of cases where this is likely to happen:

- IPX provider provides topology hiding for the service provider
- IPX provider sets or modifies service providers' Origin-Realm on request of the service provider
- IPX provider provides IMSI-to-EPC-Realm mapping
- IPX provider delivers the Diameter end-to-end security function as a service to the service provider

In those cases both the sender and the receiver of the message must use the correct key to process the message correctly. In order to do so the protocol must make sure that both sender and receiver understand which key to use. Also, the sending or receiving service provider must advertise the delegation of security functions in their IR.21[2].

Figure 3 shows the secure perimeter if the sending service provider has delegated the Diameter end-to-end security function to its IPX provider:

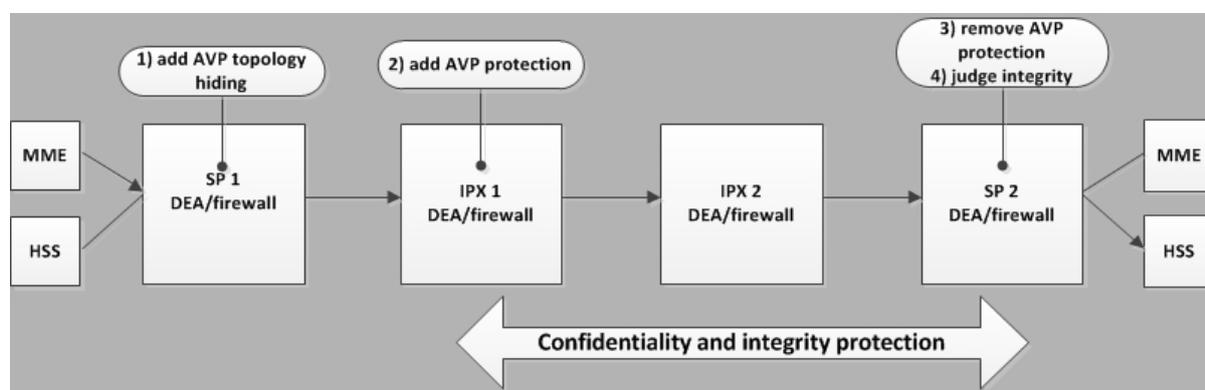


Figure 3 – Secure perimeter in case of delegation of Diameter end to end security by the sender

In this example service provider 1 has delegated the Diameter end-to-end security function² to IPX provider 1. This means that the confidentiality and integrity protection reaches from IPX provider 1 to service provider 2 only. In this case the receiver (service provider 2) must understand that although the Origin-Realm points to service provider 1, cryptographic keys between IPX provider 1 and service provider 2 are used.

Figure 4 shows the secure perimeter if the receiving service provider has delegated the security function to its IPX provider.

² In this example topology hiding is done by SP 1, but also this function may be delegated to IPX 1

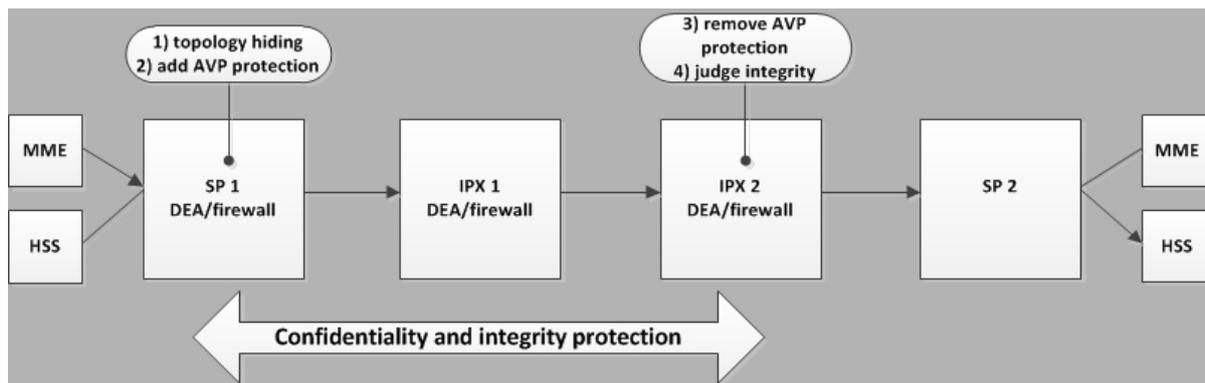


Figure 4 – Secure perimeter in case of delegation of Diameter end-to-end security by the receiver

In this case the receiving service provider has delegated the Diameter end-to-end security function to its IPX provider. Service provider 1 must be aware, when adding AVP protection that the end of the secure perimeter is IPX provider 2³. Although the Destination-Realm points to service provider 2, AVPs need to be encrypted with the key of IPX 2.

4.3.2 Secure perimeter in case of LTE roaming hub

IR.80 [2] specifies a number of interconnection models for the roaming hubbing service. Table 4 contains a description of the impact:

LTE roaming hub scenario	Impact
Direct connection	No impact: See scenario described in 4.1
Origin/Destination Realm Based Routing	In case the Origin-Realm is modified to the LTE roaming hub to enforce routing of the answer to the hub, the secure perimeter needs to be split in two, See Figure 5. Otherwise no impact
Destination-Realm modification	No impact, the Destination-Realm will get the suffix .hub-realm but is removed before routing to the receiving service provider

Table 4 – Impact in case of LTE roaming hub

Figure 5 below shows the scenario where the roaming hub changes the Origin-Realm:

³ This is a challenge if service provider 2 has more than one IPX providers

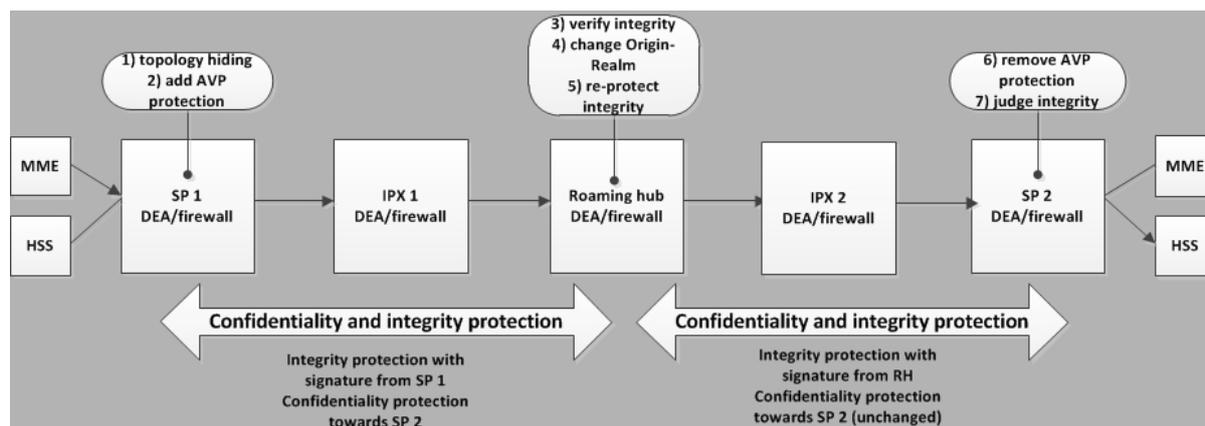


Figure 5 – Origin-Realm modification by roaming hub

In case the Origin-Realm is changed by the roaming hub there are two secure perimeters. Service provider 1 protects integrity for its Origin-Realm and applies confidentiality towards SP 2. The roaming hub will verify the integrity, change the Origin-Realm, re-apply integrity protection and optionally protect the confidentiality of additional AVPs towards service provider 2.

5 How to protect

Solution directions are divided into integrity (including authenticity), confidentiality and key management. Based on the current threat landscape, integrity is more urgent than confidentiality where key management is required for all of the mentioned measures.

5.1 Providing Integrity

A digital signature⁴ is the most common way of providing integrity (and authenticity by signing fields such as Origin-Realm and Origin-Host) in an asymmetric cryptographic system.

5.1.1 Common approach⁵

The common way of providing integrity is to sign (the hash of) the entire contents of the message that needs to be integrity protected and subsequently store that signature in a newly to-be-defined field. Figure 6 illustrates a high level description of such a flow.

⁴ The exact protocol to be defined later

⁵ By the common approach is referenced the initial work published on <https://github.com/P1sec/SigFW>

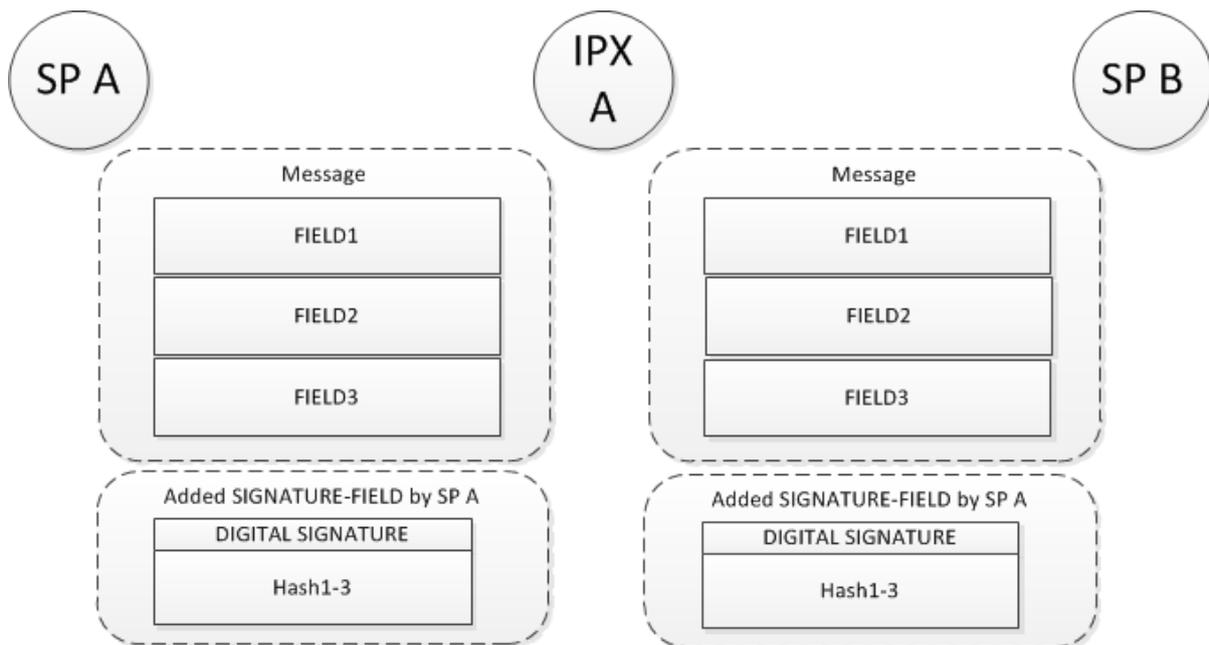


Figure 6 – Standard approach of protecting integrity of messages

Service provider A sends a message to service provider B via IPX provider A which is in this case a simplified message containing only 3 fields. At the border of service provider A, a digital signature is added containing the hash of field 1 to 3 signed with the private key of service provider A. service provider B will apply the same operation. It will hash field 1 to 3 and compares the calculated hash with the hash in the signature.

This approach is challenging if one want to allow intermediate changes to be made without changing the contents of the message. Any change to a field by IPX provider A makes the signature useless.

5.1.2 Proposed approach (Extended Signature Scheme)

An alternative approach will bring more flexibility into the ecosystem without comprising the security. In this approach which is outlined in Figure 7 the sender (service provider A) will now also sign the contents of field 1 to 3 but in this case service provider A will hash the contents of each field separately (H1,H2 and H3) and XORs the calculated hashes with each other and will put this a newly created field, SIGNATURE-FIELD.

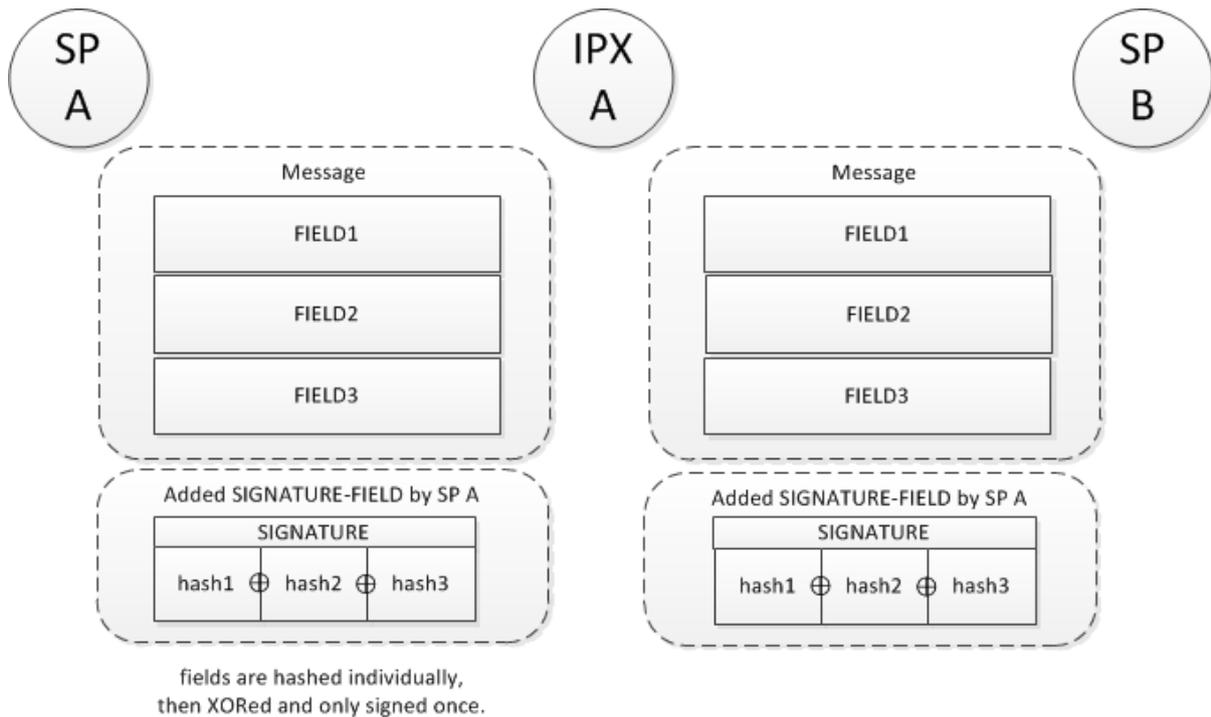


Figure 7 – Proposed approach of protecting integrity of messages

Also in this case service provider B can verify the integrity by doing the same operations as service provider A did and verify the hash, which is in fact the XOR operation between the individual hashes ($H1 \oplus H2 \oplus H3$). If there are no intermediate changes expected, this way of signing is overly complex, but it becomes beneficial if an IPX provider or providers need to make intermediate changes.

5.1.3 Proposed approach with intermediate changes (Extended Signature Scheme)

If in the previous example IPX provider A needed to make a change to a single field. It should be possible to do so without making the original signature by service provider A irrelevant.

This case is illustrated in Figure 8.

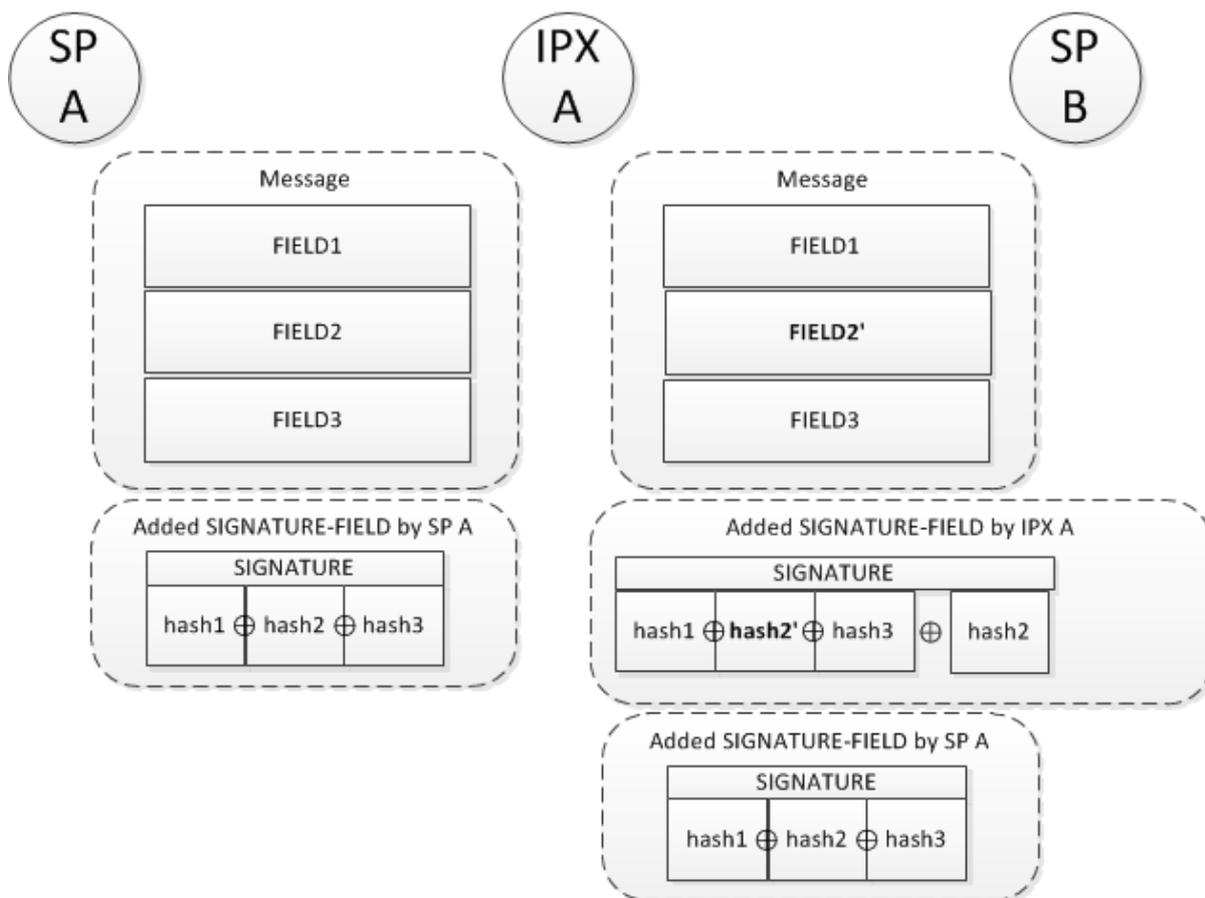


Figure 8 – Protecting integrity of messages with intermediate changes

In this case service provider A includes 3 fields into the message. Service provider A will sign the XOR operation of individual hashes of the 3 fields (signature= $H1 \oplus H2 \oplus H3$) and will include this value into a newly created SIGNATURE-FIELD.

IPX provider A wants to modify FIELD2, indicated in figure 8 as **FIELD2'**. IPX provider A signs all fields that should be protected again ($H1 \oplus H2' \oplus H3$) but does an XOR with the hash of the original version of AVP 2 (signature= $H1 \oplus H2' \oplus H3 \oplus H2$) and will include this value into a newly created Signature-FIELD. The new SIGNATURE-FIELD is put on top, leaving the previous signature by service provider A intact.

The receiver of the message, service provider B will analyse the two signatures top down. Service provider B can calculate $H1 \oplus H2' \oplus H3$ by the contents of the message and can XOR that with $H1 \oplus H2' \oplus H3 \oplus H2$ received in the signature. Due to unique properties of the XOR operations this will lead to H2 (the hash of the original value of FIELD2). This is because

$$(H1 \oplus H2' \oplus H3 \oplus H2) \oplus (H1 \oplus H2' \oplus H3) = H2$$

With the calculated value of H2 (that will only be correct if the message is untampered between IPX A and service provider B), the second signature is analysed. Service provider B can calculate H1 and H3 from the received message and can therefore calculate $H1 \oplus H3$. It takes H2 (the hash from where the original contents are not available anymore) from the previous calculation and calculates $H1 \oplus H3 \oplus H2$ which can be compared with the value

inside the signature. If both match the signature is valid since $(H1 \oplus H3 \oplus H2) = (H1 \oplus H2 \oplus H3)$, the order in XOR operations is irrelevant.

If the signature inside Signature-AVP is correct, service provider B can now conclude that:

- The message is untampered between IPX provider A and service provider B
- The message is untampered by IPX provider A (except FIELD2)
- The message is untampered between service provider A and IPX provider A

In other words the message is untampered between service provider A and service provider B (except for FIELD2).

An attacker would not be able to modify a field unless the attacker finds the exact hash of another field leading to the same result when one would do an XOR operations between the hashes. This is not considered feasible, provided that a cryptographically strong hashing algorithm is used.

To prevent unauthorized deletion of previous SIGNATURE-FIELDS and replay attacks, the result is hashed once more with the hash of the previous signature and the system time. This was not included in the explanation to improve readability.

5.1.4 Handling deletions (Extended Signature Scheme)

Deletions require the same type of calculations as modifications. The metadata should point to a certain field that is deleted instead of modified. In Figure 9, FIELD2 is deleted:

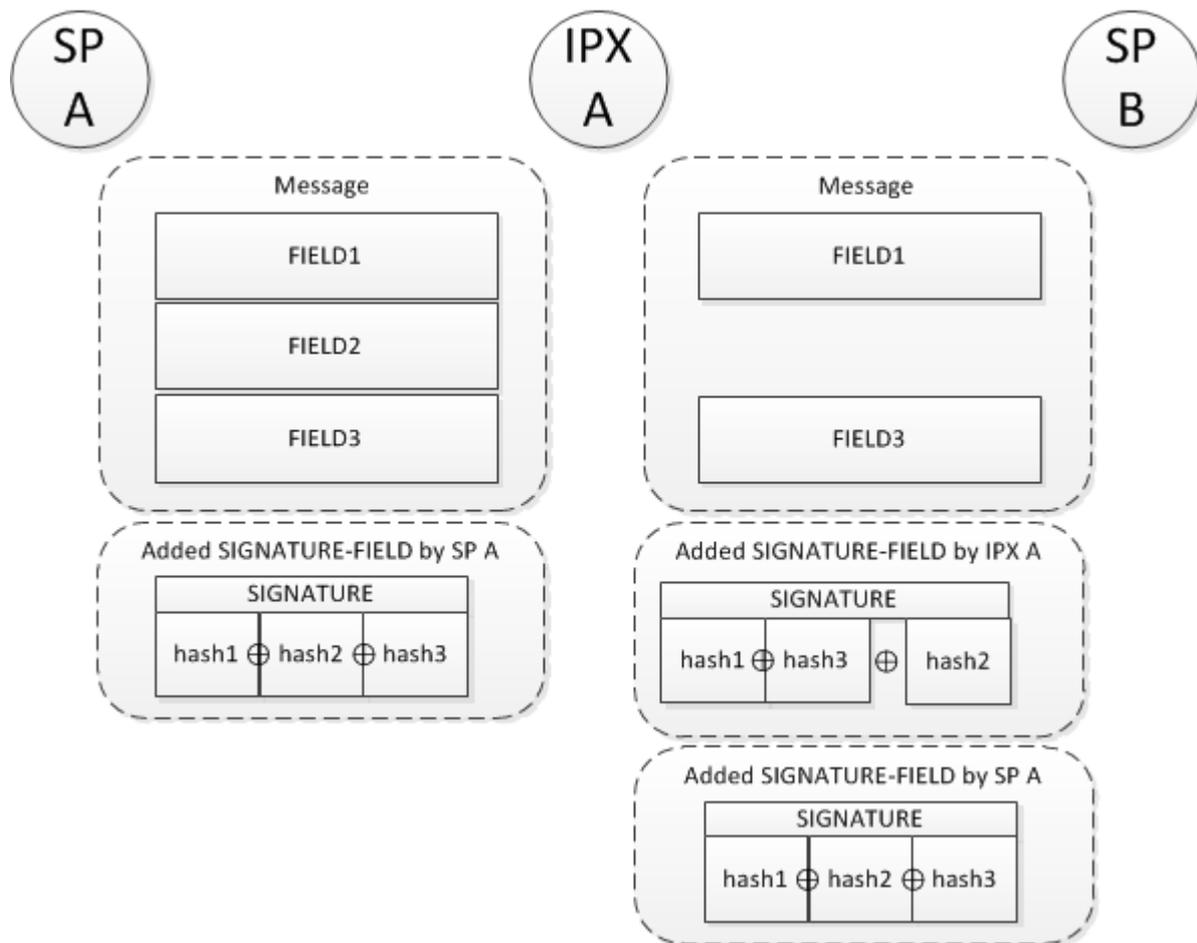


Figure 9 – Protecting integrity of messages with intermediate deletions

5.1.5 Handling additions (Extended Signature Scheme)

Additions are just another signature signed with the key of the one including current and new fields as shown in Figure 11.

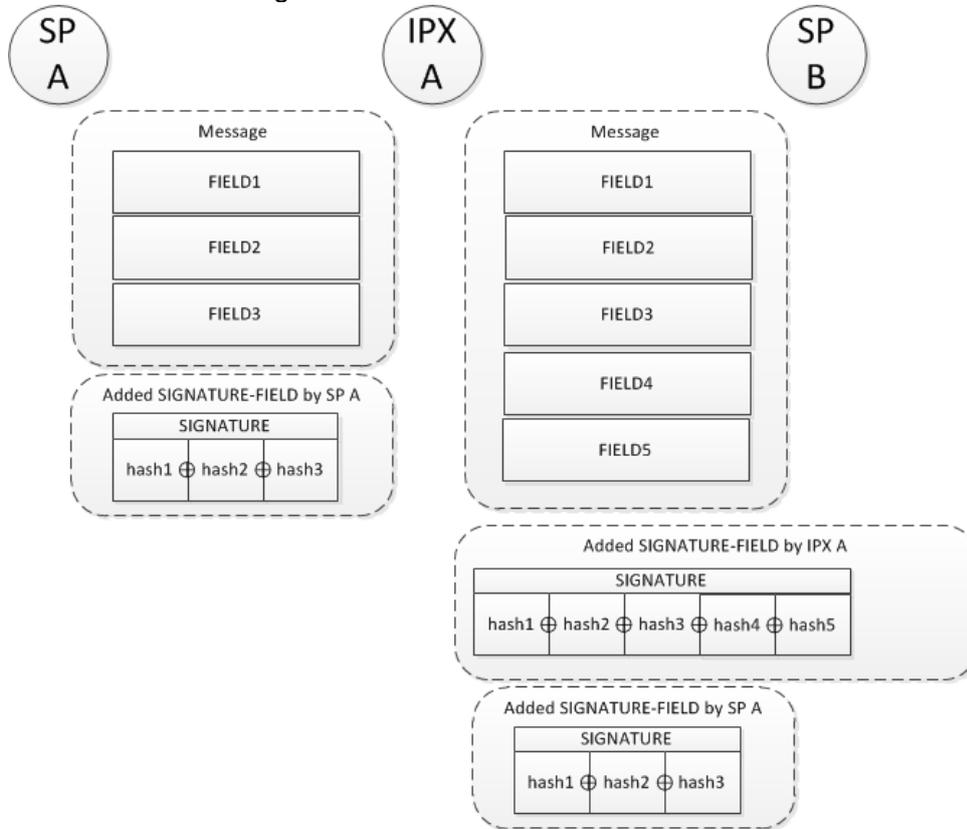


Figure 10 – Protecting integrity of messages with intermediate additions

5.1.6 Handling multiple operations (Extended Signature Scheme)

Any signature can be divided into a “head” and a “tail”. The head is the XOR between the individual hashes of the fields that are currently present. The tail is the XOR between the individual hashes of the fields that have been modified or deleted. Each signature is then effectively Head ⊕ Tail, best explained with an example. If a message consists of FIELD1-3 and an IPX provider wants to add FIELD4, modify FIELD2 and delete FIELD3, it would sign what is depicted in Figure 12.

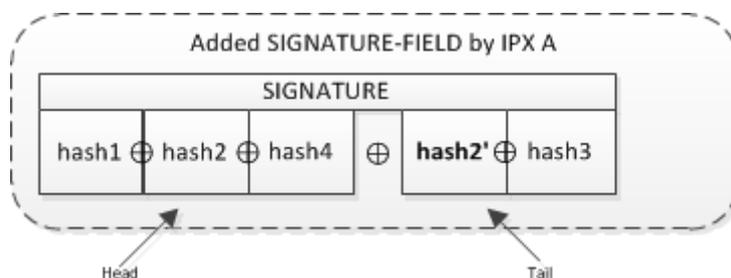


Figure 11 – Signature in case of multiple operations

The metadata should then contain a “trail” of additions, modifications and deletions in order to for the receiver to determine what has been done.

5.1.7 Contents of the SIGNATURE-FIELD

The SIGNATURE-FIELD is a field containing:

Name	Description
Network indicator	The indicator pointing to entity that signed the message, if not present the Origin-Realm applies
Signature space	A digital signature with the result of the individual hashes of the field XORed with each other XORed with the time (to prevent replay attacks) and the hash of the previous signature signed with the private key of the sender
Time	The system time to compare with the time in the signature space
Action	Modification, Addition or Deletion
AVP code	The codes of the field ⁶ that where added, deleted or modified

Table 5 – Contents of the SIGNATURE-FIELD

5.2 Providing confidentiality

This section covers confidentiality of signalling data with the main focus on end-to-end security. Compared to regular IT type end-to-end security there are special considerations in signalling traffic where partial confidentiality may be required. Signalling traffic is often carried not from point to point but through intermediate or transit points known as IPX or Signalling Connection Control Part (SCCP) carriers. This is true for SS7 as well as Diameter and is likely to hold true also for 5G networks. The carriers are sometimes required to perform changes to packets as they are transited between two points and this inhibits the ability to use end-to-end encryption of the entire packet or signalling unit.

5.2.1 Common Confidentiality Model⁷

The goal of the confidentiality scheme proposed is to provide a method which allows an endpoint to protect any field in a transmitted packet while leaving the rest of the fields open to manipulation by intermediate nodes while the packet is in transit between two end points. For such a scheme to be viable there are a few considerations that have to be made.

- Encryption Type – Symmetric vs Asymmetric
 - Key exchange
- Encryption container structure
- Processing Overhead
 - Overhead imposed by metadata required

⁶ In case of Diameter this would be the AVP code

⁷ By the common approach is referenced the initial work published on <https://github.com/P1sec/SigFW>

- Overhead imposed by block cipher padding
- Overhead imposed by additional processing

5.2.1.1 Encryption Type – Symmetric vs Asymmetric

There are distinct differences between symmetric and asymmetric encryption and there are advantages and drawbacks to both models.

- **Asymmetric Encryption benefits**
 - Simple key exchange. The two parties simply exchange public keys at the outset of their business relation and can start communicating
 - Scalability – The public key can be distributed to all edge nodes of the network without any concerns for key exchange process
- **Asymmetric Encryption drawbacks**
 - CPU intensive – Asymmetric encryption an order of magnitude more processing power than symmetric encryption
 - Lack of forward secrecy – If or when an asymmetric key is compromised it means that also all previously exchanged data is compromised.
- **Symmetric Encryption benefits**
 - High performance – Symmetric key offers much higher performance. So much higher that compression could be introduced to compensate for the encryption size overhead.
 - Key can be changed at any time – The symmetric key approach requires a key exchange procedure, which also means that the key can be updated at interval or at any time.
 - Forward secrecy – As the key exchange allows for regular update of encryption key we only loose the data related to one specific key if or when a key is compromised, and the lifetime of a key can be specified by the two communicating peers.
 - Multiple peers could share the same dynamic key – Using a symmetric key we would also be able to share the key with multiple peers, such as an intermediate carrier. If this is desired.
- **Symmetric Encryption drawbacks**
 - Key exchange is more complicated – To enable use of a symmetric key there must exist a method of transmitting the key between peers securely
 - Old key needs to be retained until all answers or congested packets are processed or dropped

Given the traits of the two schemes we are inclined to recommend the symmetric key option as it provides multiple major benefits both with regards to processing requirements as well as additional options. The main obstacle for the symmetric key model appears to be in the key exchange scheme, especially if this takes place between multiple hosts on each side. A solution however could be realized using DHE (Diffie-Hellman Key Exchange) or ECDHE (Elliptic Curve Diffie-Hellman Key Exchange). In the example Network X and Y are

communicating with each other, they each have a mated pair of edge nodes (Edge 1 and 2 on each respective side) that manage the end to end encryption of specific sensitive fields in their signalling communications. In the example illustrated in Figure 13, Edge X1 is the one initiating the key exchange procedure; this procedure can be initiated by any of the edge nodes. The exchange can be driven by a request for key update or a timer mandating key update at regular interval. The procedure begins by Edge X1 generating a fresh symmetric key. This can be done in the background as to not introduce latency in the traffic flow. When the key has been generated we can track the flow as in in Figure 13.

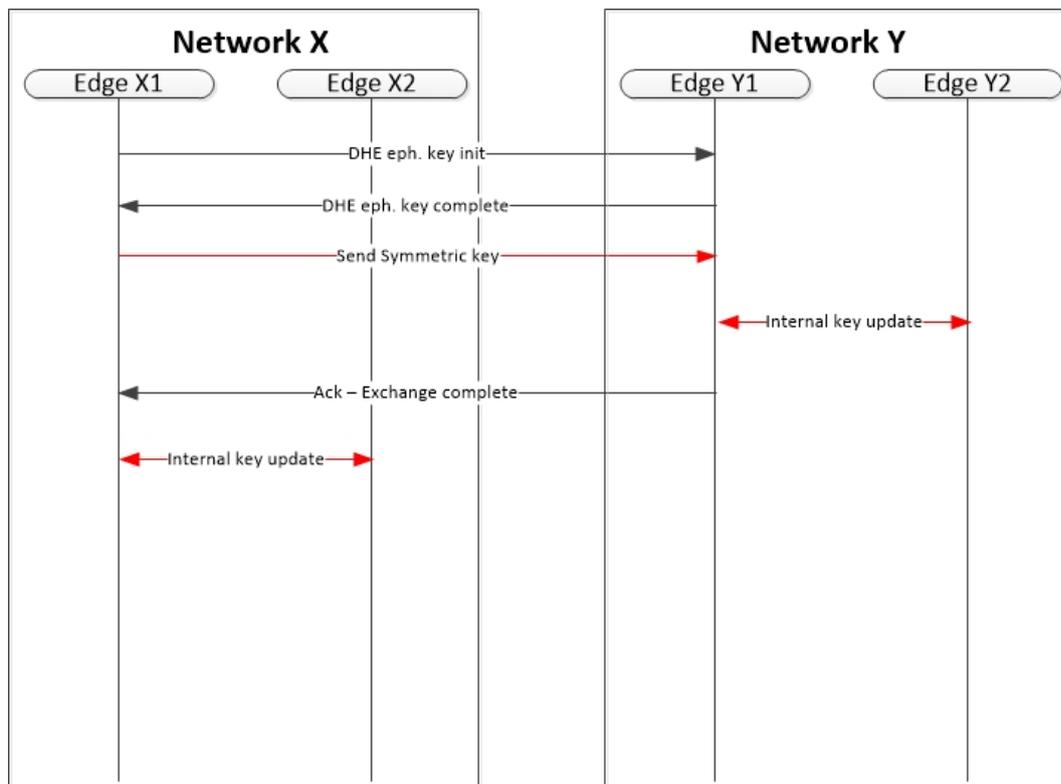


Figure 12 – Symmetric key exchange

Edge X1 initiates the Diffie-Hellman ephemeral session with Edge Y1 (or Y2, it can be either of the two or more edge nodes of the receiving Network). Edge Y1 responds to complete the ephemeral key setup. Edge X1 encrypts the symmetric key using the established ephemeral key and then transmits the symmetric key securely (illustrated in red) to Edge Y1.

Edge Y1 uses a secure method to propagate the symmetric key to Edge Y2 (and Y3, 4 and as many other Edge nodes that may exist on the network). Once the key has been propagated Edge Y1 can acknowledge back to Edge X1 that Network Y has updated all keys. Please note that Network Y does not have to rely on the standardized key exchange method internally. As long as the internal key exchange is secure it may be managed in a proprietary way.

When Edge X1 receives the acknowledgement from Network Y it will go ahead and propagate the key within Network X. This propagation may be managed just as the internal key propagation within Network Y. As long as it is managed securely, it may use any proprietary method. When all edge nodes in Network X have updated their keys, the key exchange is complete.

Both Network X and Y are now using the new symmetric key for the confidentiality scheme where specific fields are protected. An important aspect of key update is that the old key must be retained to manage decryption of data that was in transit between the networks during the key update, or packets that were rejected due to congestion or any other reason. This however poses no real world problem as the respective networks can simply try the new key first and upon failure try the previous key. As soon as all congested or leftover packets have been decrypted or timed out, the old key can be discarded.

It is important to note that a symmetric key for each network must be maintained within the edge node, similar to a routing table a key table must be established for the networks with which protected communication is required.

5.2.1.2 Encryption container structure

For performance and security reasons, the number of encryption operations should be reduced to minimum and all of the fields (AVPs in Diameter) should be put together before performing encryption (to prevent dictionary/pre-computed attacks on the separate fields which could often contain the same value). Additionally, into the input it is always useful to put some random/pseudorandom value or time information which will also make it harder to use pre-computed attacks.

As a solution, the encryption container (new grouped AVP in Diameter) could be created where the fields (AVPs in Diameter) which will be encrypted are moved. Into the container also the time information is added and then the container is encrypted. A similar algorithm has been already implemented in the SigFW project for SS7 and Diameter⁸.

This approach does not introduce additional metadata. The limitation with the Diameter is only that the AVPs after decryption are reconstructed in a different order and the other limitation is that all the nested fields are encrypted in the parent field (i.e. the encryption is done only on top-protocol level)

5.2.1.3 Encryption container structure (Extended Confidentiality Scheme: allowing selective fields Encryption)

Encryption of small data blocks such as fields in a signalling packet is costly in terms of both processing and storage overhead and could have also security implications. It is therefore desirable to store several such fields inside a single container prior to encrypting them. This will help align the data and avoid overhead created by padding required by the encryption procedure and it will limit the amount of CPU time spent encrypting the data. For an encryption container to be viable, it must also present a method of restoring encrypted data to its original format on arrival. This method could rely on metadata, which is data containing information regarding what data is contained within the encrypted container as well as where the data is intended to be inserted when restoring the original unprotected message.

Signalling data often contains a method of nesting data to help organize sub fields that are part of a wider context defined by a root field. For example in SS7 this is known as a “Constructed method” and one example could be found in the MAP operation sendAuthenticationInfo returnResultLast, there we find the root field authenticationSetList

⁸ <https://github.com/P1Sec/SigFW>

that contains a set of nested quintupletLists which in turn contain a set of AuthenticationQuintuplets. The same type of nesting is part of the Diameter protocol for LTE where such nested fields are known as “Grouped AVP”, for example in the Insert-Subscriber-Data Request we find nested Subscription-Data containing a nested APN-Configuration-Profile which in turn contains an APN-Configuration that is further nested. In summary, there is extensive nesting in signalling protocols and we may find that simply protecting the root field and all sub fields may not be possible, due to the nature of signalling traffic and the intermediate changes that may have to be made by transit nodes between two communicating networks (the case where IPX and SCCP carriers require access to some nested fields).

5G based on JSON offers the same possibility of nesting through “Parent objects” which in turn can contain other objects and so forth. We can expect 5G signalling data to make use of this feature to create abstractions and separation between different types of data being transmitted, some of which may be partially sensitive to eavesdropping or manipulation. An ideal encryption container should therefore support nesting and be able to protect selectively within nested fields. This can be accommodated in several ways and some required features already exist in Diameter specification such as the P bit for AVPs that illustrate that a particular AVP is protected. Such a P bit or JSON text-based equivalent can serve as an indicator for intermediate nodes to disregard a specific field and for receiving nodes it may serve as an indicator that the particular field will be found inside the encryption container.

The proposed encryption container model consists of a method where metadata is created from the “Header” information of each field. Such header information is assumed to contain an identifier that uniquely identifies the type of field, any specific traits the field may have such as defining flags, such as if a field is vendor specific, mandatory, protected and so forth. In Diameter we would also expect the length of the field data to be defined in the header but this may not be required in the 5G JSON scenario.

Each field will thus contain a header portion and a data portion. The data portion being the actual value contained in the field. It is the data that should be confidentiality protected and this could be realized as illustrated in Figure 14 below:

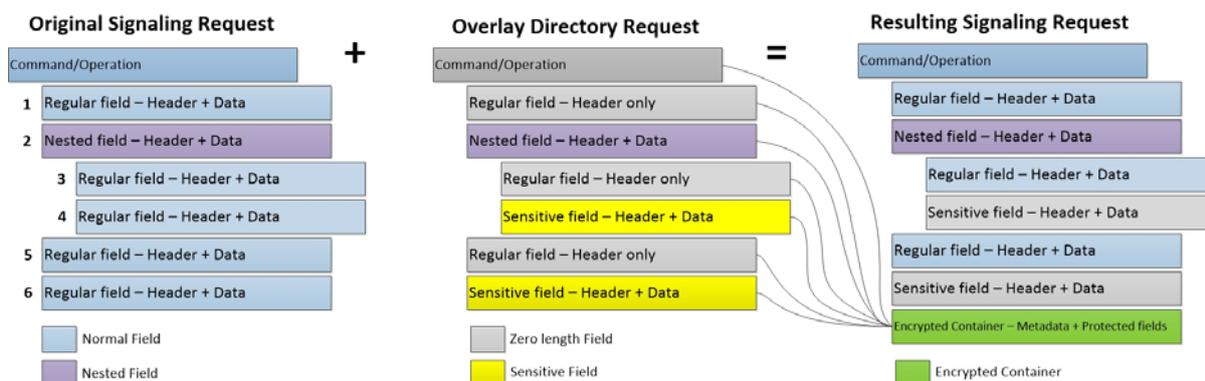


Figure 13 – Proposed encryption container

The Original Signalling Request in the above example contains a Command/Operation that indicates what type of operation to execute on the receiving side, this could be for example a location update. Furthermore the Original Signalling Request contains 6 information fields,

each consisting of a header part and a data part. 5 of the fields are normal fields, and 1 field is a nested field. Normal fields are simple one-dimensional fields and illustrated in light blue and the nested field is a parent field that contain 2 normal fields. We assume that field 4 and 6 are particularly sensitive and therefore should be protected. As illustrated, field 4 is part of the nested field 2, and field 6 is a simple normal field.

In order to create viable metadata we can create an Overlay Directory Request that contains all of the header information of the original request, but only the data that requires confidentiality protection. Essentially this becomes a skeleton copy of the original request that holds a perfect representation of the original request so that the protected fields can easily be restored when the request reaches the receiving network. This Overlay Directory Request is then encrypted during a single encryption pass. The encrypted data is then appended to the original request as a single field, and the original data in the sensitive fields are either erased, or replaced with dummy data to preserve the structure of the message.

5.2.1.4 Processing Overhead

Introducing confidentiality protection will also introduce several types of overhead in terms of CPU usage and signalling unit size and the proposed scheme is no exception. The overhead generated by the introduction of metadata depends on multiple factors, for example how many fields there are in the signalling request or answer, and the size of each header. In a protocol such as Diameter, the header size is generally fixed and limited to two different sizes either 8 or 12 byte depending on if a Vendor Id is included or not, in JSON the header sizes may be expected to be slightly larger, but that is likely to be true also for the data carried by the signalling request or answer. One way of mitigating the size overhead of the introduced metadata would be to compress the container prior to encryption. Compression is generally perceived as costly in terms of CPU time but there are high performance methods such as LZ4 that would be viable. Compression of JSON data can also be expected to yield good results as text generally compresses well even when the data is relatively small, as an indicator, compression can work well on data sizes well below 1000 bytes.

There are however other aspects that may have to be considered when introducing encryption. Encryption is generally managed in chunks relative in size to the length of the key being used, so for a key length of 2048 bits a block size would be 256 bytes (2048/8). Any blank space left behind data smaller than 256 would become padding due to the nature of block ciphers. This means that regardless of the size of the data being protected, we would still always be encrypting at least 256 bytes. In addition to this the encrypted data must be harmonized with JSON, essentially converted to a text format which would increase the overhead further. This may not necessarily be a problem if compression of the container is introduced, as this could negate a substantial part of the overhead and if a symmetric key approach is chosen the impact on CPU usage should be acceptable.

5.3 Relation between providing confidentiality and integrity

Integrity and confidentiality measures need to work together. Since confidentiality measures are expected to be encrypted end-to-end it makes more sense to apply confidentiality measures first and then apply the integrity measures on top, whereby an encrypted container and the fields that are represented in the container may not be changed.

Annex A Document Management

A.1 Document History

Version	Date	Brief Description of Change	Approval Authority	Editor / Company
1.0	17 Nov 2017	First approved version	DESS	Ewout Pronk, (KPN), Fredrik Söderlund (Symsoft), Martin Kacer (P1 Security)

A.2 Other Information

Type	Description
Document Owner	Fraud and Security Group
Editor / Company	Ewout Pronk, KPN

It is our intention to provide a quality product for your use. If you find any errors or omissions, please contact us with your comments. You may notify us at prd@gsma.com

Your comments or suggestions & questions are always welcome.