3GPP TR 33.855 V16.1.0 (2020-09)

Technical Report

3rd Generation Partnership Project;

Technical Specification Group Services and System Aspects;

Security Aspects;

Study on security aspects of the 5G Service Based Architecture (SBA)

(Release 16)

** 

The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP.  
The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented.  
This Report is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification.  
Specifications and Reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices.

***3GPP***

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis

Valbonne - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

***Copyright Notification***

No part may be reproduced except as authorized by written permission.  
The copyright and the foregoing restriction extend to reproduction in all media.

© 2020, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

UMTS™ is a Trade Mark of ETSI registered for the benefit of its members

3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  
LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners

GSM® and the GSM logo are registered and owned by the GSM Association

Contents

Foreword [10](#__RefHeading___Toc51259067)

1 Scope [12](#__RefHeading___Toc51259068)

2 References [12](#__RefHeading___Toc51259069)

3 Definitions of terms, symbols and abbreviations [13](#__RefHeading___Toc51259070)

3.1 Terms [13](#__RefHeading___Toc51259071)

3.2 Symbols [13](#__RefHeading___Toc51259072)

3.3 Abbreviations [13](#__RefHeading___Toc51259073)

4 Key Issues [13](#__RefHeading___Toc51259074)

4.1 General SBA Key Issues [13](#__RefHeading___Toc51259075)

4.1.1 Key Issue #1: Confidentiality protection of signalling messages [13](#__RefHeading___Toc51259076)

4.1.1.1 Issue description [13](#__RefHeading___Toc51259077)

4.1.1.2 Threat description [13](#__RefHeading___Toc51259078)

4.1.1.3 Potential security requirements [14](#__RefHeading___Toc51259079)

4.1.2 Key Issue #2: Integrity protection of signalling messages while allowing for modifications [14](#__RefHeading___Toc51259080)

4.1.2.1 Issue description [14](#__RefHeading___Toc51259081)

4.1.2.2 Threat description [14](#__RefHeading___Toc51259082)

4.1.2.3 Potential security requirements [14](#__RefHeading___Toc51259083)

4.1.3 Key Issue #3: Replay protection of signalling messages [14](#__RefHeading___Toc51259084)

4.1.3.1 Issue description [14](#__RefHeading___Toc51259085)

4.1.3.2 Threat description [15](#__RefHeading___Toc51259086)

4.1.3.3 Potential security requirements [15](#__RefHeading___Toc51259087)

4.1.4 Key Issue #4: NF-NF Authentication [15](#__RefHeading___Toc51259088)

4.1.4.1 Issue description [15](#__RefHeading___Toc51259089)

4.1.4.2 Threat description [15](#__RefHeading___Toc51259090)

4.1.4.3 Potential security requirements [15](#__RefHeading___Toc51259091)

4.1.5 Key Issue #5: NF-NF Authorization [15](#__RefHeading___Toc51259092)

4.1.5.1 Issue description [15](#__RefHeading___Toc51259093)

4.1.5.2 Threat description [15](#__RefHeading___Toc51259094)

4.1.5.3 Potential security requirements [15](#__RefHeading___Toc51259095)

4.1.6 Key Issue #6: NF-NRF Authentication [16](#__RefHeading___Toc51259096)

4.1.6.1 Issue description [16](#__RefHeading___Toc51259097)

4.1.6.2 Threat description [16](#__RefHeading___Toc51259098)

4.1.6.3 Potential security requirements [16](#__RefHeading___Toc51259099)

4.1.7 Key Issue #7: NF-NRF Authorization [16](#__RefHeading___Toc51259100)

4.1.7.1 Issue description [16](#__RefHeading___Toc51259101)

4.1.7.2 Threat description [16](#__RefHeading___Toc51259102)

4.1.7.3 Potential security requirements [16](#__RefHeading___Toc51259103)

4.1.8 Key Issue #8: NRF-NRF Authentication [17](#__RefHeading___Toc51259104)

4.1.8.1 Issue description [17](#__RefHeading___Toc51259105)

4.1.8.2 Threat description [17](#__RefHeading___Toc51259106)

4.1.8.3 Potential security requirements [17](#__RefHeading___Toc51259107)

4.1.9 Key Issue #9: NRF-NRF Authorization [17](#__RefHeading___Toc51259108)

4.1.9.1 Issue description [17](#__RefHeading___Toc51259109)

4.1.9.2 Threat description [17](#__RefHeading___Toc51259110)

4.1.9.3 Potential security requirements [17](#__RefHeading___Toc51259111)

4.1.10 Key Issue #20: Protection of SeCoP interfaces [17](#__RefHeading___Toc51259112)

4.1.10.1 Issue description [17](#__RefHeading___Toc51259113)

4.1.10.2 Threat description [18](#__RefHeading___Toc51259114)

4.1.10.3 Potential security requirements [18](#__RefHeading___Toc51259115)

4.1.11 Key Issue #21: Secure message transport via the SeCoP [19](#__RefHeading___Toc51259116)

4.1.11.1 Issue description [19](#__RefHeading___Toc51259117)

4.1.11.2 Threat description [19](#__RefHeading___Toc51259118)

4.1.11.3 Potential security requirements [19](#__RefHeading___Toc51259119)

4.1.12 Key Issue #22: Authorization of NF service access in Indirect Communication [19](#__RefHeading___Toc51259120)

4.1.12.1 Issue description [19](#__RefHeading___Toc51259121)

4.1.12.2 Threat description [19](#__RefHeading___Toc51259122)

4.1.12.3 Potential security requirements [20](#__RefHeading___Toc51259123)

4.1.13 Key Issue #23: NF to NF authentication and authorization in Indirect communication [20](#__RefHeading___Toc51259124)

4.1.13.1 Issue description [20](#__RefHeading___Toc51259125)

4.1.13.2 Threat description [20](#__RefHeading___Toc51259126)

4.1.13.3 Potential security requirements [20](#__RefHeading___Toc51259127)

4.1.14 Key Issue #24: Service access authorization based on NF Set [20](#__RefHeading___Toc51259128)

4.1.14.1 Issue description [20](#__RefHeading___Toc51259129)

4.1.14.2 Threat description [20](#__RefHeading___Toc51259130)

4.1.14.3 Potential security requirements [21](#__RefHeading___Toc51259131)

4.1.15 Key Issue #25: Indirect communication in roaming scenarios [21](#__RefHeading___Toc51259132)

4.1.15.2 Threat description [21](#__RefHeading___Toc51259133)

4.1.15.3 Potential Architecture requirements [21](#__RefHeading___Toc51259134)

4.1.16 Key Issue #26: Protection of N9 interface [21](#__RefHeading___Toc51259135)

4.1.16.1 Issue description [21](#__RefHeading___Toc51259136)

4.1.16.2 Threat description [21](#__RefHeading___Toc51259137)

4.1.16.3 Potential security requirements [21](#__RefHeading___Toc51259138)

4.1.17 Key Issue #27: Support of a UP gateway function on the N9 interface [22](#__RefHeading___Toc51259139)

4.1.17.1 Issue description [22](#__RefHeading___Toc51259140)

4.1.17.2 Threat description [22](#__RefHeading___Toc51259141)

4.1.17.3 Potential security requirements [23](#__RefHeading___Toc51259142)

4.1.18 Key Issue #28: Service access authorization in the delegated "Subscribe-Notify" scenarios [23](#__RefHeading___Toc51259143)

4.1.18.1 Issue description [23](#__RefHeading___Toc51259144)

4.1.18.2 Threat description [23](#__RefHeading___Toc51259145)

4.1.18.3 Potential security requirements [23](#__RefHeading___Toc51259146)

4.1.19 Key Issue #29: Resource level authorization of NF consumers [23](#__RefHeading___Toc51259147)

4.1.19.1 Issue description [23](#__RefHeading___Toc51259148)

4.1.19.2 Threat description [23](#__RefHeading___Toc51259149)

4.1.19.3 Potential security requirements [24](#__RefHeading___Toc51259150)

4.1.20 Key Issue #30: Service access authorization for non-delegated subscribe-notify [24](#__RefHeading___Toc51259151)

4.1.20.1 Issue description [24](#__RefHeading___Toc51259152)

4.1.20.2 Threat description [24](#__RefHeading___Toc51259153)

4.1.20.3 Potential security requirements [24](#__RefHeading___Toc51259154)

4.2 SEPP-/N32-specific Key Issues [24](#__RefHeading___Toc51259155)

4.2.1 Key Issue #10: Termination points of N32 security [24](#__RefHeading___Toc51259156)

4.2.1.1 Issue description [24](#__RefHeading___Toc51259157)

4.2.1.2 Threat description [24](#__RefHeading___Toc51259158)

4.2.1.3 Potential security requirements [24](#__RefHeading___Toc51259159)

4.2.2 Key Issue #11: Local provisioning of SEPP protection policies [25](#__RefHeading___Toc51259160)

4.2.2.1 Issue description [25](#__RefHeading___Toc51259161)

4.2.2.2 Threat description [25](#__RefHeading___Toc51259162)

4.2.2.3 Potential security requirements [25](#__RefHeading___Toc51259163)

4.2.3 Key Issue #12: Provisioning of SEPP protection policies over N32 [25](#__RefHeading___Toc51259164)

4.2.3.1 Issue description [25](#__RefHeading___Toc51259165)

4.2.3.2 Threat description [25](#__RefHeading___Toc51259166)

4.2.3.3 Potential security requirements [25](#__RefHeading___Toc51259167)

4.2.4 Key Issue #13: SEPP session setup [26](#__RefHeading___Toc51259168)

4.2.4.1 Issue description [26](#__RefHeading___Toc51259169)

4.2.4.2 Threat description [26](#__RefHeading___Toc51259170)

4.2.4.3 Potential security requirements [26](#__RefHeading___Toc51259171)

4.2.5 Key Issue #14: Application of ciphering and integrity protection to JSON object using JOSE [26](#__RefHeading___Toc51259172)

4.2.5.1 Issue description [26](#__RefHeading___Toc51259173)

4.2.5.2 Threat description [26](#__RefHeading___Toc51259174)

4.2.5.3 Potential security requirements [27](#__RefHeading___Toc51259175)

4.2.6 Key Issue #15: Malicious messages received on the N32 interface [27](#__RefHeading___Toc51259176)

4.2.6.1 Issue description [27](#__RefHeading___Toc51259177)

4.2.6.2 Threat description [29](#__RefHeading___Toc51259178)

4.2.6.3 Potential security requirements [29](#__RefHeading___Toc51259179)

4.2.7 Key Issue #16: N32 error signalling [29](#__RefHeading___Toc51259180)

4.2.7.1 Issue description [29](#__RefHeading___Toc51259181)

4.2.7.2 Threat description [29](#__RefHeading___Toc51259182)

4.2.7.3 Potential security requirements [29](#__RefHeading___Toc51259183)

4.2.8 Key Issue #17: Modifications by authorized intermediaries on N32 [30](#__RefHeading___Toc51259184)

4.2.8.1 Issue description [30](#__RefHeading___Toc51259185)

4.2.8.2 Threat description [30](#__RefHeading___Toc51259186)

4.2.8.3 Potential security requirements [30](#__RefHeading___Toc51259187)

4.2.9 Key Issue #18: Inter-PLMN routing and TLS [30](#__RefHeading___Toc51259188)

4.2.9.1 Key issue detail [30](#__RefHeading___Toc51259189)

4.2.9.2 Security threats [31](#__RefHeading___Toc51259190)

4.2.9.3 Potential security requirements [31](#__RefHeading___Toc51259191)

4.2.10 Key Issue #19: Configurational error handling by the SEPP [31](#__RefHeading___Toc51259192)

4.2.10.1 Issue description [31](#__RefHeading___Toc51259193)

4.2.10.2 Threat description [32](#__RefHeading___Toc51259194)

4.2.10.3 Potential security requirements [32](#__RefHeading___Toc51259195)

5. Void [32](#__RefHeading___Toc51259196)

6 Solutions [32](#__RefHeading___Toc51259197)

6.1 Solution #1: Authorization of NF service access [32](#__RefHeading___Toc51259198)

6.1.1 Introduction [32](#__RefHeading___Toc51259199)

6.1.2 Solution details [33](#__RefHeading___Toc51259200)

6.1.2.1 Service authorization procedure for non-roaming scenarios [33](#__RefHeading___Toc51259201)

6.1.2.2 Authorization of NF service access for roaming scenario [34](#__RefHeading___Toc51259202)

6.1.3 Evaluation [34](#__RefHeading___Toc51259203)

6.2 Solution #2: Application layer protection based on JSON Object Signing and Encryption (JOSE) [34](#__RefHeading___Toc51259204)

6.2.1 General [34](#__RefHeading___Toc51259205)

6.2.2 Application layer protection based on JOSE [35](#__RefHeading___Toc51259206)

6.2.2.1 JSON based IEs that require protection (WHAT) [35](#__RefHeading___Toc51259207)

6.2.2.2 Integrity and Confidentiality protection schemes (HOW) [35](#__RefHeading___Toc51259208)

6.2.2.2.1 Integrity protection based on JSON patch [35](#__RefHeading___Toc51259209)

6.2.2.2.2 Authorization of modifications based on JSON patch [38](#__RefHeading___Toc51259210)

6.2.2.2.3 Authentication of intermediaries [39](#__RefHeading___Toc51259211)

6.2.2.2.4 Rewriting of HTTP message into JSON-object [39](#__RefHeading___Toc51259212)

6.2.2.3 Void [39](#__RefHeading___Toc51259213)

6.2.3 Evaluation [39](#__RefHeading___Toc51259214)

6.3 Solution #3: NF service registration process [40](#__RefHeading___Toc51259215)

6.3.1 Void [40](#__RefHeading___Toc51259216)

6.3.2 Solution Details [40](#__RefHeading___Toc51259217)

6.3.3 Evaluation [40](#__RefHeading___Toc51259218)

6.4 Solution #4: Authorization of NF service access [40](#__RefHeading___Toc51259219)

6.4.1 Introduction [40](#__RefHeading___Toc51259220)

6.4.2 Solution details [41](#__RefHeading___Toc51259221)

6.4.2.1 Authorization of NF service access in the same PLMN [41](#__RefHeading___Toc51259222)

6.4.2.2 Authorization of NF service access in different PLMNs [43](#__RefHeading___Toc51259223)

6.4.1 Evaluation [44](#__RefHeading___Toc51259224)

6.5 Solution #5: Using mediation services with end-to-end encryption [44](#__RefHeading___Toc51259225)

6.5.1 Generic [44](#__RefHeading___Toc51259226)

6.5.2 End-to-end encryption using HTTPS or TLS [44](#__RefHeading___Toc51259227)

6.5.3 End-to-end security using JOSE [45](#__RefHeading___Toc51259228)

6.5.4 Migration paths after accepting this solution [47](#__RefHeading___Toc51259229)

6.5.5 Possible deployments [47](#__RefHeading___Toc51259230)

6.5.6 Evaluation [47](#__RefHeading___Toc51259231)

6.6 Solution #6: Policies for protection on the N32 interface [48](#__RefHeading___Toc51259232)

6.6.1 Void [48](#__RefHeading___Toc51259233)

6.6.2 Solution details [48](#__RefHeading___Toc51259234)

6.6.3 Evaluation [48](#__RefHeading___Toc51259235)

6.7 Solution #7: Signaling based provisioning of message protection policy in partner SEPPs [49](#__RefHeading___Toc51259236)

6.7.1 Void [49](#__RefHeading___Toc51259237)

6.7.2 Solution details [49](#__RefHeading___Toc51259238)

6.7.3 Evaluation [49](#__RefHeading___Toc51259239)

6.8 Solution #8: Inter PLMN routing and TLS: Solution Options [50](#__RefHeading___Toc51259240)

6.8.1 Introduction [50](#__RefHeading___Toc51259241)

6.8.3 TLS tunnel or VPN from NF to SEPP [50](#__RefHeading___Toc51259242)

6.8.4 Using local SEPP FQDN in request URI [50](#__RefHeading___Toc51259243)

6.8.5 Mapped FQDN in request URI [51](#__RefHeading___Toc51259244)

6.8.6 Evaluation [52](#__RefHeading___Toc51259245)

6.9 Solution #9: N32 message anti-spoofing within the SEPP [52](#__RefHeading___Toc51259246)

6.9.1 [52](#__RefHeading___Toc51259247)

Void [52](#__RefHeading___Toc51259248)

6.9.2 Solution Details [52](#__RefHeading___Toc51259249)

6.9.3 Evaluation [52](#__RefHeading___Toc51259250)

6.10 Solution #10: Mitigation against fraudulent registration attack between SEPPs [52](#__RefHeading___Toc51259251)

6.10.1 Introduction [52](#__RefHeading___Toc51259252)

6.10.2 Solution Details [52](#__RefHeading___Toc51259253)

6.10.3 Evaluation [52](#__RefHeading___Toc51259254)

6.11 Solution #11: Security policy provisioning for SEPP [53](#__RefHeading___Toc51259255)

6.11.1 Void [53](#__RefHeading___Toc51259256)

6.11.2 Solution Details [53](#__RefHeading___Toc51259257)

6.11.3 Evaluation [53](#__RefHeading___Toc51259258)

6.12 Solution #12: End-to-end data protection in hop-by-hop network communication links [53](#__RefHeading___Toc51259259)

6.12.1 Introduction [53](#__RefHeading___Toc51259260)

6.12.2 Integrity protection with non-repudiation and traceability of changes [54](#__RefHeading___Toc51259261)

6.12.3 Integrity protection with non-repudiation, traceability of changes, and authorization [54](#__RefHeading___Toc51259262)

6.12.4 Confidentiality protection with authorization [55](#__RefHeading___Toc51259263)

6.13 Solution #13: Content and structure of protection policies [55](#__RefHeading___Toc51259264)

6.13.1 Introduction [55](#__RefHeading___Toc51259265)

6.13.2 Data-type encryption policy [56](#__RefHeading___Toc51259266)

6.13.3 NF API data-type placement mapping [56](#__RefHeading___Toc51259267)

6.13.4 Modification policy [56](#__RefHeading___Toc51259268)

6.13.5 Evaluation [57](#__RefHeading___Toc51259269)

6.14 Solution #14: Provisioning and negotiation of protection policies [57](#__RefHeading___Toc51259270)

6.14.1 Introduction [57](#__RefHeading___Toc51259271)

6.14.2 Provisioning of the policies in the SEPP [57](#__RefHeading___Toc51259272)

6.14.3 Negotiation of protection policies [57](#__RefHeading___Toc51259273)

6.14.4 Evaluation [57](#__RefHeading___Toc51259274)

6.15 Solution #15: Service access authorization in the delegated "Subscribe-Notify" interaction scenarios [58](#__RefHeading___Toc51259275)

6.15.1 Introduction [58](#__RefHeading___Toc51259276)

6.15.2 Solution details [58](#__RefHeading___Toc51259277)

6.15.3 Evaluation [60](#__RefHeading___Toc51259278)

6.16 Solution #16: OAuth 2.0 based authorization for Indirect communication without Delegated Discovery (Model C) [61](#__RefHeading___Toc51259279)

6.16.1 Introduction [61](#__RefHeading___Toc51259280)

6.16.2 Solution details [61](#__RefHeading___Toc51259281)

6.16.3 Evaluation [61](#__RefHeading___Toc51259282)

6.17 Solution #17: Protection of SeCoP interfaces [61](#__RefHeading___Toc51259283)

6.17.1 Introduction [61](#__RefHeading___Toc51259284)

6.17.2 Solution details [62](#__RefHeading___Toc51259285)

6.17.3 Evaluation [62](#__RefHeading___Toc51259286)

6.18 Solution #18: Support NDS/IP on the inter-PLMN N9 interface [62](#__RefHeading___Toc51259287)

6.18.1 Introduction [62](#__RefHeading___Toc51259288)

6.18.2 Solution details [62](#__RefHeading___Toc51259289)

6.18.3 Evaluation [63](#__RefHeading___Toc51259290)

6.19 Solution #19: Service access authorization based on NF Set in non-roaming scenario [63](#__RefHeading___Toc51259291)

6.19.1 Introduction [63](#__RefHeading___Toc51259292)

6.19.2 Solution details [63](#__RefHeading___Toc51259293)

6.19.2.0 General [63](#__RefHeading___Toc51259294)

6.19.2.1 Service access authorization for NF producers within a NF set (Model B) [63](#__RefHeading___Toc51259295)

6.19.2.2 Service access authorization based on NF Set by verifying the token on the service producer (Model C) [65](#__RefHeading___Toc51259296)

6.19.3 Evaluation [65](#__RefHeading___Toc51259297)

6.20 Solution #20: UP Gateway function on the inter-PLMN N9 interface [66](#__RefHeading___Toc51259298)

6.20.1 Introduction [66](#__RefHeading___Toc51259299)

6.20.2 Solution details [66](#__RefHeading___Toc51259300)

6.20.2.1 Interface between SEPP-U and Core Network control plane entity [66](#__RefHeading___Toc51259301)

6.20.2.2 Interface between UPFs and SEPP-U [67](#__RefHeading___Toc51259302)

6.20.3 Evaluation [67](#__RefHeading___Toc51259303)

6.21 Solution #21: OAuth 2.0 based authorization for Indirect communication with Delegated Discovery (Model D) [68](#__RefHeading___Toc51259304)

6.21.1 Introduction [68](#__RefHeading___Toc51259305)

6.21.2 Solution details [69](#__RefHeading___Toc51259306)

6.21.2.1 SeCoP obtaining access token on behalf of the NF consumer [69](#__RefHeading___Toc51259307)

6.21.2.2 SeCoP authorizing NF consumer based on token verification [70](#__RefHeading___Toc51259308)

6.21.2.4 SeCoP includes access token in the Service Response message [70](#__RefHeading___Toc51259309)

6.21.3 Evaluation [70](#__RefHeading___Toc51259310)

6.22 Solution #22: Authentication and authorization between Network Functions for Indirect Communication models [71](#__RefHeading___Toc51259311)

6.22.1 Introduction [71](#__RefHeading___Toc51259312)

6.22.2 Solution details [71](#__RefHeading___Toc51259313)

6.22.3 Evaluation [71](#__RefHeading___Toc51259314)

6.23 Solution #23: Token-based authorization for Scenario D using stateless SeCoP [71](#__RefHeading___Toc51259315)

6.23.1 Introduction [71](#__RefHeading___Toc51259316)

6.23.2 Solution Description [72](#__RefHeading___Toc51259317)

6.23.2.1 General [72](#__RefHeading___Toc51259318)

6.23.2.2 Assumptions on authentication and interface protection [72](#__RefHeading___Toc51259319)

6.23.2.3 Authorization and service invocation procedure [72](#__RefHeading___Toc51259320)

6.23.2.4 Trust model [73](#__RefHeading___Toc51259321)

6.23.3 Solution Evaluation [73](#__RefHeading___Toc51259322)

6.24 Solution #24: Token-based authorization for Scenario C using stateless SeCoP [73](#__RefHeading___Toc51259323)

6.24.1 Introduction [73](#__RefHeading___Toc51259324)

6.24.2 Solution Description [74](#__RefHeading___Toc51259325)

6.24.2.1 General [74](#__RefHeading___Toc51259326)

6.24.2.2 Assumptions on authentication and interface protection [74](#__RefHeading___Toc51259327)

6.24.2.3 Authorization and service invocation procedure [74](#__RefHeading___Toc51259328)

6.24.2.4 Trust model [75](#__RefHeading___Toc51259329)

6.24.3 Solution Evaluation [75](#__RefHeading___Toc51259330)

6.25 Solution #25: NF service consumer verification during service access authorization in the direct communication scenario [75](#__RefHeading___Toc51259331)

6.25.1 Introduction [75](#__RefHeading___Toc51259332)

6.25.2 Solution details for the non-roaming scenario [75](#__RefHeading___Toc51259333)

6.25.2.0 General [75](#__RefHeading___Toc51259334)

6.25.2.1 Access token generation with the certificate of the NF service consumer [76](#__RefHeading___Toc51259335)

6.25.2.2 NF service Producer authenticates NF consumer [77](#__RefHeading___Toc51259336)

6.25.3 Solution details for the roaming scenario [77](#__RefHeading___Toc51259337)

6.25.4 Evaluation [78](#__RefHeading___Toc51259338)

6.26 Solution #26: OAuth 2.0 based resource level authorization of NF service consumers [79](#__RefHeading___Toc51259339)

6.26.1 Introduction [79](#__RefHeading___Toc51259340)

6.26.2 Solution Description [79](#__RefHeading___Toc51259341)

6.26.3 Solution Evaluation [80](#__RefHeading___Toc51259342)

6.27 Solution #27: Policy based authorization for Indirect communication between Network functions [80](#__RefHeading___Toc51259343)

6.27.1 Introduction [80](#__RefHeading___Toc51259344)

6.27.2 Solution Description [81](#__RefHeading___Toc51259345)

6.27.2.1 Policy files [81](#__RefHeading___Toc51259346)

6.27.2.2 Procedure [81](#__RefHeading___Toc51259347)

6.27.3 Solution Evaluation [82](#__RefHeading___Toc51259348)

6.28 Solution #28: Authorization between Network Functions in Scenario D [82](#__RefHeading___Toc51259349)

6.28.1 Introduction [82](#__RefHeading___Toc51259350)

6.28.2 Solution details [82](#__RefHeading___Toc51259351)

6.28.3 Evaluation [82](#__RefHeading___Toc51259352)

6.29 Solution #29: Telescopic FQDN for the SeCoP [82](#__RefHeading___Toc51259353)

6.29.1 Introduction [82](#__RefHeading___Toc51259354)

6.29.2 Solution Description [82](#__RefHeading___Toc51259355)

6.29.3 Solution Evaluation [82](#__RefHeading___Toc51259356)

6.30 Solution #30: Token-based authorization for NF Sets / NF Service Sets by existing methods [83](#__RefHeading___Toc51259357)

6.30.1 Introduction [83](#__RefHeading___Toc51259358)

6.30.2 Solution Description [83](#__RefHeading___Toc51259359)

6.30.3 Solution Evaluation [83](#__RefHeading___Toc51259360)

6.31 Solution #31: Service access authorization based on of a NF Set in roaming scenario [83](#__RefHeading___Toc51259361)

6.31.1 Introduction [83](#__RefHeading___Toc51259362)

6.31.2 Solution details [83](#__RefHeading___Toc51259363)

6.31.2.0 General [83](#__RefHeading___Toc51259364)

6.31.2.1 Service access authorization for NF producers within a NF set (Model C) [84](#__RefHeading___Toc51259365)

6.31.3 Solution Evaluation [84](#__RefHeading___Toc51259366)

6.32 Solution #32: OAuth 2.0 based resource level authorization of NF service consumers [85](#__RefHeading___Toc51259367)

6.32.1 Introduction [85](#__RefHeading___Toc51259368)

6.32.2 Solution Description [85](#__RefHeading___Toc51259369)

6.32.2.0 General [85](#__RefHeading___Toc51259370)

6.32.2.1 NF OAuth 2.0 client (NF service consumer) registration with the OAuth 2.0 authorization server (NRF) [85](#__RefHeading___Toc51259371)

6.32.2.2 NF OAuth 2.0 resource server (NF service producer) registration with the OAuth 2.0 authorization server (NRF) [86](#__RefHeading___Toc51259372)

6.32.2.3 NF Access token request before service access [86](#__RefHeading___Toc51259373)

6.32.2.4 NF Service access request based on token verification [88](#__RefHeading___Toc51259374)

6.32.3 Solution Evaluation [89](#__RefHeading___Toc51259375)

6.33 Solution #33: NF service consumer verification during service access authorization in indirect communication scenario [89](#__RefHeading___Toc51259376)

6.33.1 Introduction [89](#__RefHeading___Toc51259377)

6.33.2 Solution Description [89](#__RefHeading___Toc51259378)

6.33.2.1 General [89](#__RefHeading___Toc51259379)

6.33.2.2 Solution details for the scenario C [89](#__RefHeading___Toc51259380)

6.33.2.2.0 General [89](#__RefHeading___Toc51259381)

6.33.2.2.1 Access token generation with the certificate of the NF service consumer [90](#__RefHeading___Toc51259382)

6.33.2.2.2 SeCoP authenticates NF consumer [91](#__RefHeading___Toc51259383)

6.33.2.3 Solution details for the scenario D [91](#__RefHeading___Toc51259384)

6.33.2.3.0 General [91](#__RefHeading___Toc51259385)

6.33.2.3.1 Access token generation with the certificate of the NF service consumer [92](#__RefHeading___Toc51259386)

6.33.2.3.2 SeCoP authenticates NF consumer [93](#__RefHeading___Toc51259387)

6.33.2.4 Solution details for the roaming scenario [93](#__RefHeading___Toc51259388)

6.33.3 Evaluation [94](#__RefHeading___Toc51259389)

6.34 Solution #34: Security of indirect communication in roaming scenarios [94](#__RefHeading___Toc51259390)

6.34.1 Introduction [94](#__RefHeading___Toc51259391)

6.34.2 Solution Description [94](#__RefHeading___Toc51259392)

6.34.3 Solution Evaluation [95](#__RefHeading___Toc51259393)

6.35 Solution #35: Service access authorization in the non-delegated "Subscribe-Notify" interaction scenarios [95](#__RefHeading___Toc51259394)

6.35.1 Introduction [95](#__RefHeading___Toc51259395)

6.35.2 Solution details [95](#__RefHeading___Toc51259396)

6.35.3 Evaluation [97](#__RefHeading___Toc51259397)

7 Conclusions [97](#__RefHeading___Toc51259398)

7.1 Conclusion on KI #20 [97](#__RefHeading___Toc51259399)

7.2 Conclusions on Key Issue #21: Secure message transport via the SeCoP [97](#__RefHeading___Toc51259400)

7.3 Conclusions on Key issue #22: Authorization of NF service access in indirect communication [97](#__RefHeading___Toc51259401)

7.4 Conclusion on KI #23 [97](#__RefHeading___Toc51259402)

7.5 Conclusion on KI #24 [98](#__RefHeading___Toc51259403)

7.6 Conclusions on Key issue #25: Indirect communication in roaming scenarios [98](#__RefHeading___Toc51259404)

7.7 Conclusion on KI #26 [98](#__RefHeading___Toc51259405)

7.8 Conclusion on KI #27 [98](#__RefHeading___Toc51259406)

7.9 Conclusion on KI #29 [98](#__RefHeading___Toc51259407)

Annex A : Void [99](#__RefHeading___Toc51259408)

Annex B: Options for integrity protection on the N32 interface [100](#__RefHeading___Toc51259409)

Annex C: Deployment options for the UP gateway [101](#__RefHeading___Toc51259410)

C.1 Deployment option 1: UP Gateway per slice [101](#__RefHeading___Toc51259411)

C.2 Deployment option 2: UP Gateway as shared appliance [101](#__RefHeading___Toc51259412)

Annex D: Change history [103](#__RefHeading___Toc51259413)

# Foreword

This Technical Report has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

In the present document, modal verbs have the following meanings:

**shall** indicates a mandatory requirement to do something

**shall not** indicates an interdiction (prohibition) to do something

The constructions "shall" and "shall not" are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions "must" and "must not" are not used as substitutes for "shall" and "shall not". Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

**should** indicates a recommendation to do something

**should not** indicates a recommendation not to do something

**may** indicates permission to do something

**need not** indicates permission not to do something

The construction "may not" is ambiguous and is not used in normative elements. The unambiguous constructions "might not" or "shall not" are used instead, depending upon the meaning intended.

**can** indicates that something is possible

**cannot** indicates that something is impossible

The constructions "can" and "cannot" are not substitutes for "may" and "need not".

**will** indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**will not** indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**might** indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

**might not** indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

**is** (or any other verb in the indicative mood) indicates a statement of fact

**is not** (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions "is" and "is not" do not indicate requirements.

# 1 Scope

3GPP TS 23.501 [2] defines 5G services with a new service based architecture (SBA) approach.

The present document reviews the interactions in this new architecture, determines key issues relating to the security of SBA elements and interfaces, details potential solutions and recommends normative work .

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 23.501: "System Architecture for the 5G System".

[3] 3GPP TS 23.502: "Procedures for the 5G System".

[4] JSON Object Signing and Encryption (<https://datatracker.ietf.org/wg/jose/charter>).

[5] IETF RFC 7515: "JSON Web Signature" (<https://tools.ietf.org/html/rfc7515>).

[6] IETF RFC 7516: "JSON Web Encryption" (<https://tools.ietf.org/html/rfc7516>).

[7] IETF RFC 7518: "JSON Web Algorithms" (<https://tools.ietf.org/html/rfc7518>).

[8] V. Goyal, O. Pandey, Amit Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proc. CCS '06, New York, 2006, pp. 89-98.

[9] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption" in Proc. SP '07, 2007, pp. 321-334.

[10] C. Chen, J. Chen, H.-W. Lim, Z. Zhang, D. Feng, "Combined Public-Key Schemes: The Case of ABE and ABS" in Proc. Provable Security '12, Lecture Notes in Computer Science, vol. 7496, 2012, pp. 53-69.

[11] 3GPP TS 33.501: "Security architecture and procedures for 5G System".

[12] Void

[13] Void

[14] GSMA: "IPX Network End-to End Security Guidelines", V1.0, Nov 2017.

[15] IETF RFC 8446: "The Transport Layer Security (TLS) Protocol Version 1.3".

[16] IETF RFC 1123: "Requirements for Internet Hosts -- Application and Support".

[17] 3GPP TR 23.742: "Study on Enhancements to the Service-Based Architecture".

[18] 3GPP TS 29.244: "Interface between the Control Plane and the User Plane nodes".

[19] IETF RFC 6749: "The OAuth 2.0 Authorization Framework".

[20] 3GPP TS 33.210: "Network Domain Security (NDS); IP network layer security".

# 3 Definitions of terms, symbols and abbreviations

## 3.1 Terms

For the purposes of the present document, the terms given in TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in TR 21.905 [1].

## 3.2 Symbols

Void.

## 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 21.905 [1].

SeCoP Service Communication Proxy

NOTE: TS 23.501 [2] uses the multiply overloaded acronym SCP for the Service Communication Proxy.

# 4 Key Issues

## 4.1 General SBA Key Issues

### 4.1.1 Key Issue #1: Confidentiality protection of signalling messages

#### 4.1.1.1 Issue description

Confidentiality protection of (some information elements in) signalling messages transferred between NFs ensures that third parties are unable to extract any relevant information from the communication. Failing to provide such protection for signalling traffic between any two NFs by means of ciphering can be considered a key issue for both security and user privacy.

In inter-PLMN communication, IPX providers offer valuable services to operators that may require them to access certain information contained in the transferred signalling messages. Thus, protecting the confidentiality of every single information element might not be desirable.

In particular, leaking the IMSI, cryptographic material and location data may constitute a breach of user's privacy and may lead to fraud towards the operator.

#### 4.1.1.2 Threat description

A lack of confidentiality protection for 5GC signalling messages may lead to the following security threats:

- Leakage of sensitive information about mobile customers, such as SUPI and/or location data

- Leakage of potentially sensitive information about the PLMN itself

- Theft of service, if authentication information/authorization credentials are transferred unciphered

#### 4.1.1.3 Potential security requirements

Relevant IEs in 5GC signalling messages exchanged via intra-PLMN communication shall be confidentiality protected.

As for inter-PLMN communication, certain sensitive information elements shall always be confidentiality protected:

- Authentication Vectors

- Cryptographic Material

- Location Data

As per agreement between operators, certain sensitive information elements shall be confidentiality protected:

- Identifiers such as IMSI, SUPI, NAI, PEI and/or IMEI;

Additionally, it is recommended to cipher the remaining information elements as well, unless agreed otherwise with the roaming partner or IPX provider.

### 4.1.2 Key Issue #2: Integrity protection of signalling messages while allowing for modifications

#### 4.1.2.1 Issue description

Integrity protection ensures that signalling messages transferred between NFs cannot be modified without the receiving party noticing that such modification occurred. In combination with authenticity, integrity is an essential guarantee for the correctness and validity of messages.

In inter-PLMN communication, however, IPX providers offer valuable services to operators that may require them to modify certain information contained in the transferred signalling messages. Thus, it might be desirable to allow for certain modifications by authorized intermediates on N32 even if the messages are integrity protected.

#### 4.1.2.2 Threat description

The lack of integrity protection for 5GC signalling messages may lead to the following security threats:

- Man in the Middle attacks that actively modify signalling messages between NFs

- Lack of availability caused by malformed messages due to unnoticed modifications during transfer

- Theft and fraud towards an operator

#### 4.1.2.3 Potential security requirements

5GC signalling messages in both intra- and inter-PLMN communication shall be integrity protected.

In intra-PLMN communications, intermediaries shall be able to apply modifications without breaking the integrity protection.

A receiving PLMN shall be able to identify the intermediary that applied modifications and verify that this was authorized.

### 4.1.3 Key Issue #3: Replay protection of signalling messages

#### 4.1.3.1 Issue description

Replay protection of signalling messages transferred between NFs ensures that it is not possible to successfully gain service from an NF by recording genuine messages and re-sending them at a later point in time.

#### 4.1.3.2 Threat description

The lack of replay protection for 5GC signalling messages may lead to the following security threats:

- Theft of service

- Leakage of potentially sensitive information about mobile customers and the network itself

- Loss of control, in case authorization credentials are replayed

#### 4.1.3.3 Potential security requirements

5GC signalling messages in both intra- and inter-PLMN communication shall be replay protected.

### 4.1.4 Key Issue #4: NF-NF Authentication

#### 4.1.4.1 Issue description

Authentication ensures that an attribute claimed by a given entity is actually correct. In the context of communication and NF-NF signalling in particular, authenticating the communication peer's identity is a key objective in order to guarantee the validity of transferred messages. Authentication is also a prerequisite for conducting authorization which itself is another key issue for 5GC signalling. already mentioned in TS 23.501 [2].

#### 4.1.4.2 Threat description

Not mutually authenticating two NFs that is not mutually authenticated could potentially allow attackers to perform the following types of attacks:

- Operating malicious NFs claiming to be genuine peers in order to request certain services (theft of service) or information (data leakage)

- Man in the Middle attacks between any genuine NFs of a given PLMN

#### 4.1.4.3 Potential security requirements

NFs shall be able to mutually authenticate each other.

### 4.1.5 Key Issue #5: NF-NF Authorization

#### 4.1.5.1 Issue description

Authorization comprises the definition and enforcement of access rights or privileges to certain resources or services. TS 23.501 [2] outlines a high-level NF service authorization framework, in which the consuming and producing NF both play a key role. After the Consumer NF has been successfully authorized by the NRF, it may subsequently be subject to another authorization procedure conducted by the Producer NF, which decides whether a certain service request is allowed based on information on "request type granularity".

#### 4.1.5.2 Threat description

If an NF does not authorize incoming requests from other NFs in the same PLMN, attackers would potentially be able to perform the following types of attacks:

- Requesting and successfully obtaining services from the NF that are not allowed for third parties, e.g. in order to extract potentially sensitive information about the network

- Causing a Denial of Service situation by successfully forcing the NF to perform resource-demanding operations

#### 4.1.5.3 Potential security requirements

An NF shall validate whether a requesting NF is authorized to request a given service(s).

### 4.1.6 Key Issue #6: NF-NRF Authentication

#### 4.1.6.1 Issue description

Since the NRF comprises the central repository of registered NFs and available NF services in a 5GC, there are several key message exchanges between NRF and its peer NFs that require secure communication. Therefore, just as in NF-NF signalling, authentication of the message source in any communication between NF and NRF is crucial. Furthermore, authentication between NF and NRF is also a prerequisite for conducting authorization which itself is another key issue for 5GC signalling and already mentioned in TS 23.501 [2].

#### 4.1.6.2 Threat description

Communication between NF and NRF that is not mutually authenticated could potentially allow attackers to perform the following types of attacks:

- Performing Man in the Middle attacks between any genuine NFs and NRFs of a given PLMN

- Registering a malicious NF with the genuine NRF of a given PLMN

- Operating a malicious NRF in order to trick genuine NFs not to register with the genuine NRF of a given PLMN

#### 4.1.6.3 Potential security requirements

Communication between NF and NRF shall be mutually authenticated.

### 4.1.7 Key Issue #7: NF-NRF Authorization

#### 4.1.7.1 Issue description

Authorization comprises the definition and enforcement of access rights or privileges to certain resources or services. TS 23.501 [2] outlines a high-level NF service authorization framework, in which both the consuming NF and NRF play a key role. In particular, the NRF checks whether the Consumer NF is allowed to discover Producer NF instances of the requested services: "This is performed on a per NF granularity by NRF".

#### 4.1.7.2 Threat description

If an NRF does not authorize incoming requests from NFs in its PLMN, attackers would potentially be able to perform the following types of attacks:

- Sending registration requests from malicious NFs which are controlled by the attacker

- Requesting services from the NRF that third parties are not allowed to consume, e.g. in order to gain information about the PLMN (potentially considered secret)

- Flooding the NRF with resource-demanding operations that may lead to a Denial of Service situation

#### 4.1.7.3 Potential security requirements

An NRF shall validate whether a given NF is authorized to register with or request certain services – such as NF service discovery – from the NRF.

### 4.1.8 Key Issue #8: NRF-NRF Authentication

#### 4.1.8.1 Issue description

For any NF service discovery that is carried out in inter-PLMN communication, the initial discovery request will be transferred from the NRF in the Consumer NF's PLMN to the NRF in the Producer NF's PLMN. The same applies to the response, potentially containing information about how to address the Producer NF and a token for authorization purposes. In order to ensure that NF service discovery is performed by valid peers, communication between two NRFs relies on mutual authentication.

#### 4.1.8.2 Threat description

Communication between NRFs across different PLMNs that is not mutually authenticated could potentially allow attackers to perform the following types of attacks:

- Performing Man in the Middle attacks between the local NRF and an NRF of a genuine roaming partner

- Querying of address information from an NRF, thereby leaking potentially secret data about the PLMN

#### 4.1.8.3 Potential security requirements

Communication between NRFs shall be mutually authenticated.

### 4.1.9 Key Issue #9: NRF-NRF Authorization

#### 4.1.9.1 Issue description

As described in Key Issue #8, communication between two NRFs precedes every service discovery an NF may send to a foreign PLMN. Being exposed to such external requests, the NRF is a valuable target for attackers that may try to send malicious requests in order to obtain illegitimate service or information. Therefore, the NRF is a key element in the 5GC NF service authorization framework and needs to ensure its communication peers are authorized to request a certain service.

#### 4.1.9.2 Threat description

If an NRF does not authorize incoming requests from NRFs in other PLMN, attackers would potentially be able to perform the following types of attacks:

- Sending discovery requests from malicious NRF which are controlled by the attacker

- Requesting services from the NRF that third parties are not allowed to consume, e.g. in order to gain information about the PLMN (potentially considered secret)

#### 4.1.9.3 Potential security requirements

An NRF shall validate whether a requesting NRF is authorized to request the given service(s).

### 4.1.10 Key Issue #20: Protection of SeCoP interfaces

#### 4.1.10.1 Issue description

If a PLMN operator utilizes an SeCoP to enable indirect communication, connections between the SeCoP and other 5GC Network Functions need to offer the same level of security as direct NF-NF or NF-NRF communication. Therefore, the interfaces between SeCoP and all communicating parties connected to it have to be mutually authenticated. With the exception of special cases, such as NRF or SEPP, which allow for service requests without prior authorization, the SeCoP may also verify whether a connected NF is allowed to make use of certain SeCoP services before accepting any such requests.

Furthermore, all examples of SeCoP in TS 23.501 [2] represent SeCoP as a distributed system (cf. example architecture in Figure 4.1.10.1-1). Therefore, SeCoP very likely contains internal interfaces, which also need to be secured.



Figure 4.1.10.1-1: Architecture from TS 23.501 [2] showing a distributed SeCoP

The purpose of this key issue is to study the following security aspects of the SeCoP:

- Authentication and authorization of the connecting NFs by the SeCoP

- SeCoP interfaces with the NRF to perform service registration/de-registration and service discovery on behalf of another NF

- Confidentiality protection, integrity protection and replay protection SeCoP at the SeCoP interface towards the Network Functions connected to it

#### 4.1.10.2 Threat description

Unauthenticated access to the SeCoP may lead to the following threats:

- Spoofing attacks

- Theft of service

- Man in the middle attacks between SeCoP and N(R)F

A lack of confidentiality protection may lead to leakage of sensitive information.

A lack of integrity protection may lead to unnoticed modification of information in transit.

A lack of replay protection may lead to several negative impacts as a result of replay attacks, such as theft of service, leakage of sensitive information, or loss of control.

#### 4.1.10.3 Potential security requirements

Communication between the SeCoP and its communication peers as well as communication within the SeCoP shall support confidentiality protection, integrity protection and replay protection.

The SeCoP shall perform mutual authentication with each communication peer before granting access to its services.

The SeCoP may further perform authorization of a requesting NF before granting access to certain services.

### 4.1.11 Key Issue #21: Secure message transport via the SeCoP

#### 4.1.11.1 Issue description

As of 3GPP Rel-16, 5GC not only supports direct communication between N(R)Fs within a single PLMN, but also indirect communication via a Service Communication Proxy (SeCoP). According to TR 23.742 [17], this SeCoP: "should support [to] protect the integrity and confidentiality of the communication."

In order to offer a similar level of protection as in the case of direct communication that is secured by TLS, the SeCoP is required to support replay protection as well. As an entity sitting between two NFs, the SeCoP provides integrity protection, confidentiality protection and replay protection of communication between NFs. If the SeCoP is a distributed system, it also protects its internal messages.

#### 4.1.11.2 Threat description

A lack of confidentiality protection by the SeCoP may lead to leakage of sensitive information.

A lack of integrity protection by the SeCoP may lead to unnoticed modification of information in transit.

A lack of replay protection by the SeCoP may lead to several negative impacts as a result of replay attacks, such as theft of service, leakage of sensitive information, or loss of control.

#### 4.1.11.3 Potential security requirements

The 5G system shall support confidentiality and integrity protection as well as replay protection of transferred messages for indirect communication.

The SeCoP shall provide confidentiality, integrity and replay protection for its internal communication over SeCoP internal network interfaces.

NOTE: Within a deployment unit the service interfaces are not mapped to a network interface, therefore it is possible to assume physical security within a deployment unit.

### 4.1.12 Key Issue #22: Authorization of NF service access in Indirect Communication

#### 4.1.12.1 Issue description

In Rel-15, verification of claims in the OAuth 2.0 access token is performed solely at the NF service producer. The token is assumed to be a self-contained JWT token, and the NF service producer, which receives the access token from the NF service consumer, is assumed to possess the necessary NRF public key to validate the token. There is no other specified method to verify the token.

In Rel-16, Service Communication Proxy (SeCoP) is introduced to enable indirect communication between two NFs.

The purpose of this key issue is to study security aspects related to authorization of NF service access in Indirect Communication between NFs.

#### 4.1.12.2 Threat description

Not applicable

#### 4.1.12.3 Potential security requirements

Not applicable

### 4.1.13 Key Issue #23: NF to NF authentication and authorization in Indirect communication

#### 4.1.13.1 Issue description

SA2 has enhanced the Network Function Service Framework to support indirect communication between NF services via a Service Communication Proxy (SeCoP).

The purpose of this key issue is to study the impact of indirect communication on security aspects of service access authentication and authorization:

a) NF to NF authentication in intra-PLMN scenarios relied on authentication at the transport layer. This needs to be studied further since end to end protection at the transport layer between NFs may no longer be possible.

b) In R15 there was no separate procedure for OAuth 2.0 Client authentication. It relies on authentication at the transport layer (for e.g. with TLS), which may no longer be possible in indirect communication mode.

c) Execution of OAuth 2.0 procedures for service access authorization will need to be clarified.

d) In R15, NF service producer could handle specific token claims (such as PLMN ID) in a UE specific manner (e.g. which NF instance may modify a UEs context may depend on the PLMN ID of the consumer NF). This could be considered "dynamic" authorization, depending on state in the producer NF. This may no longer be possible in a scenario where verification of authorization claims in the token is delegated to SeCoP.

e) It may be that there are multiple SeCoPs within a PLMN, each coordinating many NFs. Communication between these SeCoPs needs to be secured, as well as access control to NFs needs to be managed.

#### 4.1.13.2 Threat description

Lack of mutual authentication between two NFs could allow attackers to perform the following types of attacks:

- Malicious Network functions gaining access services (theft of service) or information (data leakage)

- Man-In-The-Middle (MITM) attacks between any genuine NFs of a given PLMN

#### 4.1.13.3 Potential security requirements

The information from the token claims shall be available at the producer NF.

### 4.1.14 Key Issue #24: Service access authorization based on NF Set

#### 4.1.14.1 Issue description

Rel-16 is introducing the concept of NF instance Set/NF Service instance Set, which are essentially a group of interchangeable NF instances/NF Service instances of the same type, supporting the same services and the same Network Slice(s). Rel-16 also allows re-selection of a NF instance or a NF Service instance within the Set for subsequent transaction.

The purpose of this key issue is to study the impact of this concept on OAuth 2.0 based authorization of Network Function service access.

#### 4.1.14.2 Threat description

Unauthorized access to NF instances or NF Service Instance within a set may potentially lead to attackers performing the following types of attacks:

- Obtaining services from a NF Instance or a NF Service Instance to extract potentially sensitive information about the network

- Launching Denial of Service attacks by performing resource-demanding operations on the NF instance

#### 4.1.14.3 Potential security requirements

The 5GS shall support service access authorization based on NF Set.

### 4.1.15 Key Issue #25: Indirect communication in roaming scenarios

4.1.15.1 Issue description When indirect mode of communication is used between two NFs in the roaming scenario, the two NFs do not connect with their respective SEPPs directly. The NFs communicates via the SeCoP. The SeCoP will have to establish a secure connection with the SEPP.

In Rel-15, telescopic FQDN feature with wildcard certificate for SEPP is used to enable NFs to establish a TLS connection with the SEPP.

Rel-16 also allows both direct and indirect communications between NFs to co-exist within one and the same network.

The purpose of this key issue is to study the impact of Indirect communication between NFs in roaming scenarios.

#### 4.1.15.2 Threat description

Not applicable.

#### 4.1.15.3 Potential Architecture requirements

System shall support interoperability with Rel-15.

### 4.1.16 Key Issue #26: Protection of N9 interface

#### 4.1.16.1 Issue description

In the Roaming 5G System architecture – Home Routed scenario, a GTP\_U tunnel is established per PDU session between two User Plane Functions (UPF) in the VPLMN and HPLMN. The GTP\_U traffic is sent over the N9 interface between the two UPFs.

Inter-PLMN user traffic on N9 contains any home-routed user traffic from the visited network to the home network. This traffic may contain sensitive information that needs to be protected from external visibility and manipulation.

According the objectives of this study, improved flexibility for protection of data exchanged on N9 needs to be studied, as different levels of protection may be required for different use cases.

#### 4.1.16.2 Threat description

Lack of protection of Inter-PLMN user traffic on N9 may lead to attacks that cause leakage of sensitive information and unauthorized modification of information.

#### 4.1.16.3 Potential security requirements

Confidentiality, integrity and replay protection shall be supported for Inter-PLMN GTP\_U communication on the N9 interface.

### 4.1.17 Key Issue #27: Support of a UP gateway function on the N9 interface

#### 4.1.17.1 Issue description

In the 5G system, the roaming interfaces of user plane (N9) and control plane (N32) traffic are separate. In the Rel-15 5G architecture, the SEPP acts as protection function on the control plane (N32) interface. However, in Rel-15 a protection function on the user plane (N9) interface is missing. Aim of this key issue is to study the introduction of a new UP gateway function on the N9 roaming interface for the protection of the user plane.



Figure 4.1.17.1-1: UP gateway function in the roaming 5G System architecture with home routed scenario. For local breakout, the key issue is not applicable as N9 is not a roaming interface.

The new function needs to be able to bind incoming user plane traffic to PDU sessions established, managed and released at the control plane. Hence the new function needs an interface to a control plane function handling PDU sessions. It seems that the SMF, the SEPP or the UPF are natural candidates for the other endpoint of this new interface. Hence another aim of this key issue is to study the introduction of a new interface with the new UP gateway function as one endpoint.

Solutions to this key issue need to address the following architecture requirements:

NOTE 1: Requirements on the N9 interface are not addressed in the present clause.

- The 5G system shall support a UP gateway function for user plane protection on the N9 roaming interface.

- The 5G system shall support a new interface with the UP gateway function as one endpoint, which enables exposure of session information to the UP gateway function.

- The new interface shall support the following functionalities:

- Inform the new UP gateway function with at least the TEID for established GTP-U sessions on N9 and the IP addresses of the tunnel endpoints.

- Inform the new UP gateway function when PDU and GTP-U sessions are released via control plane.

- Inform the control plane function about events detected on N9.

Different deployment options for the UP gateway are described in the Annex C.

#### 4.1.17.2 Threat description

Not applicable

#### 4.1.17.3 Potential security requirements

Not applicable

### 4.1.18 Key Issue #28: Service access authorization in the delegated "Subscribe-Notify" scenarios

#### 4.1.18.1 Issue description

"Subscribe-Notify" NF Service specified in TS 23.501, clause 7.1.2, allows one NF (e.g. NF\_A) to subscribe the service of NF producer (e.g. NF\_B) on behalf of another NF (NF\_C), which means that the NF\_A sending the subscribe service request is not the receiver NF\_C of the corresponding notification service. The access token defined in SA3 currently cannot be reused directly for this delegated "Subscribe-Notify" scenario, since the subject part of the generated token only includes the instance ID of the NF\_A.

#### 4.1.18.2 Threat description

If there is no specific an authorization mechanism for the delegated "Subscribe-Notify" scenario, NF\_A can invoke the subscribe service of NF\_B on behalf of any NF. This may lead the unauthorized NF\_C be able to use the service of NF\_B.

#### 4.1.18.3 Potential security requirements

The 5G system shall support an authorization mechanism for the delegated "Subscribe-Notify" scenarios, in which NF\_A subscribes the service of NF\_B on behalf of NF\_C.

### 4.1.19 Key Issue #29: Resource level authorization of NF consumers

#### 4.1.19.1 Issue description

An NF producer such as UDR provides one service to all the NF consumers, while different types of data may have different data access authorizations. The NF producer therefore needs to have authorization management mechanism to guarantee the safety of data access.

For e.g. the UDR provides one Nudr\_DataRepository service to all the NF consumers. Any NF can use the Nudr interface to access resources managed by UDR including UE's Subscription Data. There is no specific access permission to UDR operations with Nudr\_DataRepository service. The UE's subscription information (Subscription Data) is a very sensitive information and needs to only be accessible to UDM Network Functions (NF). Other NFs, such as NEF or PCF need not be able to obtain access to the Nudr\_DataRepository service and then have free access to all types of data including UE's subscription data. Thus, it should be possible to restrict UDM to only access Subscription Data.

In general, different types of data within a NF may have different data access authorizations. The NF needs to be able to have the authorization management mechanism to guarantee the safety of data access.

NOTE 1: The level of granularity required for authorization of resources is not addressed in the present document.

NOTE 2: Whether resource-level authorization is necessary if sensitive data is contained in a specific service is not addressed in the present document.

#### 4.1.19.2 Threat description

Unauthorized data access will allow attackers to potentially perform the following types of attacks:

- Requesting and successfully obtaining services from the NF that are not allowed for third parties, e.g. in order to extract potentially sensitive information about the network

- Causing a Denial of Service situation by successfully forcing the NF to perform resource-demanding operations

#### 4.1.19.3 Potential security requirements

An NF shall validate whether a requesting NF is authorized to access the requested resource.

### 4.1.20 Key Issue #30: Service access authorization for non-delegated subscribe-notify

#### 4.1.20.1 Issue description

"Subscribe-Notify" NF Service specified in TS 23.501, clause 7.1.2, allows one NF (e.g. NF\_A) to subscribe to notifications of NF producer (e.g. NF\_B). The subscription request includes the notification endpoint (e.g. the notification URL) of the NF Service Consumer to which the event notification from the NF Service Producer is sent to. For the scenario that NF\_A subscribes the service of NF\_B for itself, the access token defined in SA3 for service authorization currently cannot assure whether the notification URL sent in the subscription request is authorized by the NRF or not, since the subject part of the generated token only includes the instance ID of the NF\_A.

#### 4.1.20.2 Threat description

If there is no specific an authorization mechanism for the "Subscribe-Notify" scenario, NF\_A could invoke the subscribe service of NF\_B on behalf of any NF This may lead an unauthorized NF\_C to receive the notification from NF\_B, or to a reflected denial of service attack on NF\_C.

#### 4.1.20.3 Potential security requirements

The 5G system shall support an authorization mechanism for the non-delegated "Subscribe-Notify" scenarios for the scenario that NF\_A subscribes the service of NF\_B for itself.

## 4.2 SEPP-/N32-specific Key Issues

### 4.2.1 Key Issue #10: Termination points of N32 security

#### 4.2.1.1 Issue description

Protection measures are applied to information transferred via N32 between two PLMN which can potentially terminate in different endpoints of the communication, i.e.:

- In the communicating NFs themselves (i.e. end-to-end protection)

- In a SEPP of each PLMN (i.e. SEPP-to-SEPP protection)

- At each of the (maximum two) IPX providers in the path between SEPPs (i.e. hop-by-hop protection)

Each of these options offers a different level of security and requires careful consideration with regards to the complexity and the potential impact on the overall 5GC SBA architecture. The final solution needs to offer as much protection as possible whilst catering for industry-specific requirements on the interconnect, as outlined by GSMA DESS [14].

#### 4.2.1.2 Threat description

A N32 security solution that is too restrictive may not fit the Rel-15 SBA model of SA3, thus leading to additional complexity. A N32 security solution that is too complex may decrease the chances of it being used in real-world deployments.

A N32 security solution that is too permissive may not fulfil the requirements posed by GSMA DESS [14].

#### 4.2.1.3 Potential security requirements

The N32 security solution should have minimal impact on the Network Functions involved.

The N32 security solution should support the ability for mobile network operators to delegate security functionality to another entity.

### 4.2.2 Key Issue #11: Local provisioning of SEPP protection policies

#### 4.2.2.1 Issue description

When a SEPP receives a message to be sent out to a different PLMN via N32, it applies certain protection measures to it. In order to allow for a uniform protection of any given information element, independent of the sending NF, the SEPP needs to be able to detect what data-types are contained in a received message and by what means to protect them. This information is contained in a Protection Policy Suite, comprised of Data-type Encryption Policy and Data-type Modification Policy and the related attribute mappings that describe what data-types (e.g. subscription identifier or location data) are contained in the individual information elements. Thus, there needs to be a standard format for these policies as well as a standard way of locally provisioning them to the SEPP.

#### 4.2.2.2 Threat description

If there is no standard format for such Protection Policy Suites, ensuring a uniform protection of all outgoing messages on N32 will be next to impossible.

Furthermore, since the Protection Policies Suites will also be exchanged between roaming partners, a standard format is a fundamental prerequisite for ensuring interoperability between different SEPPs. Failing to provide such may impede working inter-PLMN signalling.

#### 4.2.2.3 Potential security requirements

There shall be a standard format for Protection Policy Suites, comprised of Data-type Encryption Policy and Data-type Modification Policy and the related attribute mappings.

A SEPP shall apply confidentiality and integrity protection to outgoing N32 messages according to its local Protection Policy Suite.

A receiving SEPP shall enforce modifications to incoming N32 messages according to the sender's Modification Policy.

### 4.2.3 Key Issue #12: Provisioning of SEPP protection policies over N32

#### 4.2.3.1 Issue description

In order to correctly interpret incoming messages on N32, a SEPP needs to know which attributes are protected in what way (i.e. only integrity protected or ciphered). While the ciphering information in form of the Data-type Encryption Policy is likely to be agreed by both roaming partners in advance as part of their business agreement, there is merit in validating that both SEPPs are configured with the same policy during the initial handshake.

As for the second part of the Protection Policy Suite, it is crucial that the SEPP of a roaming partner is provided with the Data-Type Modification Policy, thereby enabling it to verify the validity of modifications performed by intermediates.

#### 4.2.3.2 Threat description

If two SEPPs are unable to agree on a Protection Policy Suite, there is a possibility of ambiguous states when rewriting messages or misaligned protection measures at both ends of the communication, thereby rendering N32 communication faulty or even completely defective.

Furthermore, if a SEPP does not provide its local Data-type Modification Policy to its peer SEPP, the latter is unable to verify whether the intermediate IPX provider it does not have a business agreement with only made modifications in adherence to the policy.

#### 4.2.3.3 Potential security requirements

The Data-type Encryption Policy shall be negotiated during the initial SEPP handshake.

A SEPP shall be able to negotiate Data-type Encryption Policies with its peer SEPPs via N32-c. This includes the respective attribute mapping necessary for correctly interpreting a message. Both information shall be transferred confidentiality, integrity and replay-protected.

The Data-type Modification Policy should be exchanged during the initial SEPP handshake.

A SEPP shall be able to exchange Data-type Modification Policies with its peer SEPPs via N32-c. This includes the respective attribute mapping necessary for correctly interpreting a message. Both information shall be transferred confidentiality, integrity and replay-protected.

### 4.2.4 Key Issue #13: SEPP session setup

#### 4.2.4.1 Issue description

In order to apply the required protection mechanisms to outgoing messages as well as verifying and rewriting incoming messages on the N32 interface, two peer SEPPs need to agree on some principles of communication first. This comprises the following information:

- Type of N32 Protection, i.e. TLS or ALS

- Cipher Suites for ALS

- ALS keys for confidentiality protection

- ALS keys for integrity protection

- ALS key expiry

The exchange of above information can be summarized as SEPP session setup, which needs to be performed whenever two SEPPs start to exchange messages.

#### 4.2.4.2 Threat description

If there is no standard way of exchanging the information outlined in the previous clause, two SEPPs may not be able to successfully establish a secure N32 connection, thereby impeding inter-PLMN communication and roaming for customers from/to the related PLMNs.

#### 4.2.4.3 Potential security requirements

There shall be a standard session setup procedure performed by the SEPP in order to agree on core principles required for secure message transfer over N32.

### 4.2.5 Key Issue #14: Application of ciphering and integrity protection to JSON object using JOSE

#### 4.2.5.1 Issue description

In order to protect all parts of an outgoing message on N32, including HTTP Request Line and potential HTTP headers, the SEPP will have to rewrite every message into a defined JSON structure before applying JOSE protection mechanisms to it.

#### 4.2.5.2 Threat description

If a SEPP is unable to correctly re-write N32 messages or apply JOSE algorithms, there is no way to ensure the protection of confidentiality or integrity for messages on the N32 interface.

Furthermore, ambiguities in the application of rewriting rules and protection measures are to be avoided for interoperability reasons. If there is no standard way of conducting these message transformations, a SEPP may not be able to re-build the original HTTP message from protected one received on N32, thereby impeding roaming traffic between two operators.

#### 4.2.5.3 Potential security requirements

There shall be a standard way of re-writing messages and applying JOSE protection measures for integrity and confidentially on N32.

A SEPP shall verify the integrity of all incoming N32 messages. Messages with fail this integrity check shall be discarded.

A SEPP shall verify whether intermediates that performed changes on incoming N32 messages are authorized to do so according to the contained signature and the related Data-Type Modification Policy.

### 4.2.6 Key Issue #15: Malicious messages received on the N32 interface

#### 4.2.6.1 Issue description

In order to properly analyse the potential impact of malicious messages on the N32 interfaces and how to mitigate their security risk, the analysis is structured into three different parts. Specifically, possible message origins, destinations, as well as threat categories are differentiated as outlined below.

**A. Message origin -** Any incoming message received by the SEPP on N32 originates from one of the following groups:

1) Genuine roaming partners

2) IPX providers

3) Other parties in the IPX network

**B. Message destination -** Messages received by the SEPP on N32 can have one of the following destinations:

1) The SEPP itself (i.e. SEPP-to-SEPP signalling)

2) Network Functions within one's own PLMN

3) Others (incl. Network Functions in PLMNs of 3rd parties, invalid addresses, etc.)

**C. Threat category –** Expected message types on N32 can be broadly grouped into the following categories:

1) 3GPP application signaling (Session management, Mobility management, etc. – known from previous Releases)

2) SBA specific signaling (Service Registration, Service Discovery, Service Access, Service Subscription)

3) SEPP-to-SEPP signaling

Using this model, every possible attack vector of malicious messages is taken into account by exhaustively combining all the categories above, i.e. each origin with each destination, with each threat. Note that some of the combinations can be ruled out definitively by considering the basic, already agreed SEPP functionalities.

**Observation 1: During the initial N32-c handshake the SEPP authenticates any peer SEPP that it receives messages from based on the other party's root certificate, which has been exchanged previously via out-of-band measures. Incoming N32-c messages from SEPPs that cannot be authenticated by a root certificate are discarded.**

**Observation 2: An N32-f connection utilizes encryption and integrity keys that are derived during the initial N32-c session. Incoming N32-f messages that do not belong to an active N32-f connection with a valid set of cryptographic keys are discarded by the SEPP.**

Based on Observation 1 and 2, message origin A.2 and A3 are excluded from further analysis. It is fair to assume that IPX providers will not operate their own SEPP in order to act as an individual PLMN. While some operators may very well choose to outsource their SEPP to an IPX provider, the messages originating from their PLMNs would still be authenticated on the basis of the operator's own root certificate, not that of the IPX provider.   
However, operators will most certainly have to exchange root certificates with IPX providers to authenticate intermediate IPX providers that perform message modifications. Therefore, it needs to be ensured that an IPX provider is not able to pose as an individual roaming partner, i.e. a genuine source of N32 signaling on the basis of these certificates. In order to clearly differentiate between certificates that are used to authenticate roaming partners and certificates that are used to authenticate message modifications by intermediates, the SEPP will have to support separate certificate storages.

**Potential security requirement 1: The SEPP shall be able to clearly differentiate between certificates used for authentication of peer SEPPs and certificates used for authentication of intermediates performing message modifications, e.g. by implementing separate certificate storages.**

If above Potential security requirement 1 is realized, the authentication of messages from all other parties in the IPX network is bound to fail, since the SEPP's certificate storage for authentication is only provisioned with root certificates of genuine roaming partners. This leaves us with only genuine roaming partners (A.1) as a source for malicious messages. The possible message origins types are shown in the figure below.



Figure 4.2.6-1: Potential N32 message types originating in PLMNs of genuine roaming partners

As for the SEPP as the final destination of messages (B.1), it can be safely assumed that a hardened SEPP will only accept SEPP-to-SEPP signaling which is needed to authenticate peers, negotiate N32 session keys, etc. Any other form of Control Plane traffic, i.e. 3GPP application (C.1) and SBA-specific signaling (C.2), will usually not terminate in the SEPP. If the SEPP does receive N32 messages that it is unable to understand anyway, these messages need to, of course, be discarded.

**Potential security requirement 2: The SEPP shall discard malformed N32 signaling messages.**

Thus, only the combination of B.1/C.3 is worth analysing further. It has been already established that the SEPP will authenticate incoming SEPP-to-SEPP signaling and will discard malformed messages. Another potential threat on N32 is excessive SEPP-to-SEPP signaling, e.g. key re-negotiation requests, in order to cause a denial of service on the receiver's side. Thus, an additional protection mechanism that is necessary on N32 is rate limitation.

**Potential security requirement 3: The SEPP shall implement rate-limiting functionalities to defend itself and subsequent Network Functions against excessive Control Plane signaling. This includes SEPP-to-SEPP signaling messages.**

Control Plane signaling by successfully authenticated roaming partners and with valid source/destination addresses will eventually be routed by the SEPP to the receiving NF. However, this does not rule out malicious contents completely. A genuine roaming partner could e.g. still send fraudulent messages that may result in a denial of service for a user connected to a different PLMN as well as additional cost for the HPLMN.  
Up till now, most kinds of malicious messages discussed in the present document were related to unauthenticated or unauthorized parties trying to send messages to a certain PLMN – an issue that is best prevented at the foremost edge of the network, i.e. by the SEPP. To counter the problem of fraudulent 3GPP application signaling (e.g. session management, mobility management, etc.), the NFs themselves need to implement certain security functionalities as well. Detailed measures depend on Stage 3 message contents, but they will be similar to measures performed for legacy protocols by SS7 firewalls and Diameter Edge Agents.

**Potential security requirement 4: Each network function shall implement anti-spoofing measures by validating every incoming message for plausibility and against its internal state machine. Messages that are not valid according to the protocol specification and network state shall be discarded by the NF.**

Incoming messages on N32 may also contain spoofed destination addresses or alternatively, valid addresses that do not belong to the SEPP's own PLMN (B.3). Whether or not this is due to any malicious intent or caused by a misconfiguration, and regardless of the message type (C.1/C.2/C.3), the SEPP shall never accept or forward such messages. Similarly, anti-spoofing checks needs to be applied for origin identities on different protocol layers that should belong to the same origin, e.g. source addresses, FQDNs, PLMN IDs. Again, detailed measures depend on Stage 3 message contents.

**Potential security requirement 5: The SEPP shall implement anti-spoofing mechanisms that enable cross-layer validation of source and destination address and identifiers (e.g. FQDNs or PLMN IDs). If there is a mismatch between different layers of the message or the destination address does not belong to the SEPP's own PLMN, the message shall be discarded.**

#### 4.2.6.2 Threat description

As the primary element of filter and policy enforcement functionality for inter-PLMN signalling, it is one of the main tasks of a SEPP to protect the NF of its own PLMN from malicious traffic. If it fails to do so, attackers might be able to abuse the roaming interface to perform various types of fraud, cause leakage of information or induce Denial of Service situations, thereby preventing genuine customers or roaming partners from being served.

#### 4.2.6.3 Potential security requirements

The receiving SEPP shall be able to verify whether the sending SEPP is authorized to use the PLMN ID in the received N32 message.

The SEPP shall be able to clearly differentiate between certificates used for authentication of peer SEPPs and certificates used for authentication of intermediates performing message modifications.

The SEPP shall discard malformed N32 signaling messages.

The SEPP shall implement rate-limiting functionalities to defend itself and subsequent NFs against excessive CP signaling. This includes SEPP-to-SEPP signaling messages.

The SEPP shall implement anti-spoofing mechanisms that enable cross-layer validation of source and destination address and identifiers (e.g. FQDNs or PLMN IDs).

### 4.2.7 Key Issue #16: N32 error signalling

#### 4.2.7.1 Issue description

The N32 interface connects two SEPP that are potentially controlled by different mobile network operators or even other parties in case an operator decides to outsource certain security functionalities. The fact that N32 also allows for modifications by authorized intermediaries arguably renders it one of the most complex interfaces of the 5GC. Therefore, N32 needs to support error signalling capabilities for all significant message exchanges and foreseeable error cases in order to enable SEPP operators to perform testing and debugging during setup and operation.

#### 4.2.7.2 Threat description

Missing support for error signalling on the N32 interface will prevent SEPP operators to efficiently analyse and debug errors that may occur. In turn, this may negatively affect the stability of inter-PLMN signalling, thereby violating the essential security principle of availability.

#### 4.2.7.3 Potential security requirements

The SEPP shall support the functionality to signal errors that may occur on the receiving end of communication back to the sending SEPP.

The SEPP shall support operator configuration for handling of received error messages and resulting actions.

### 4.2.8 Key Issue #17: Modifications by authorized intermediaries on N32

#### 4.2.8.1 Issue description

Mobile network operators transfer signalling data via the IPX network with the help of third parties (i.e. IPX providers) offering services that, for example, alleviate interoperability issues or mitigate common fraud scenarios. According to GSMA DESS [14], to allow for such value-added services, IPX providers require access to certain message contents on N32.

#### 4.2.8.2 Threat description

If the N32 security solution does not allow for modifications by authorized intermediaries, operators will not be able to make use of mediation and security services offered by IPX providers.

Furthermore, if operators depend on certain services offered by their IPX providers, not allowing for authorized modifications might provide an incentive for not enabling security on the N32 interface at all.

#### 4.2.8.3 Potential security requirements

The N32 security solution shall allow for authentication and authorization of intermediaries.

The N32 security solution shall allow modifications by authorized intermediaries.

The N32 security solution shall protect information elements on a per-field basis.

The N32 security solution shall allow mobile network operators to control what IEs may or may not be modified by authorized intermediaries.

The N32 security solution shall allow mobile network operators to identify how IEs were modified by authorized intermediaries.

The N32 security solution shall allow mobile network operators to revoke the permission to modify messages in transit.

### 4.2.9 Key Issue #18: Inter-PLMN routing and TLS

#### 4.2.9.1 Key issue detail

For service invocation between different PLMNs, the SEPP needs to terminate TLS in order to modify requests and responses. This request rewriting is needed because of topology hiding and for the application layer security solution.

The FQDN in the Request URIs contain the FQDN of the remote PLMN. In order to terminate TLS, the SEPP needs to provide a certificate on behalf of remote PLMN.

TLS 1.3 [15] does not support static RSA and Diffie-Hellman cipher suites, which enable the server's private key to be shared with server-side middleboxes. It is no longer possible for a server to share a key with the middlebox and allow middlebox to access TLS session data.

The situation is illustrated in more detail in the service discovery and service request flows in Figure 4.2.9.1-1:



Red arrows represent service discovery flows.

Blue arrows represent service request flows.

Figure 4.2.9.1-1: Service discovery and service request flows

The TLS tunnels to, from, or between IPXs may not be present.

#### 4.2.9.2 Security threats

If the SEPP is issued certificates on behalf of the remote PLMN, it impersonates the remote PLMN.

#### 4.2.9.3 Potential security requirements

SEPPs shall not be provisioned with TLS certificates containing domain names belonging to other PLMNs.

Chosen solution shall be compatible with TLS 1.3 [15].

### 4.2.10 Key Issue #19: Configurational error handling by the SEPP

#### 4.2.10.1 Issue description

Application Layer Security (ALS) as described in TS 33.501 [11] allows for modification of N32-f messages in transfer by authorized third parties. It is the receiving SEPP's responsibility to validate whether the added JSON-patches are legitimate with regards to the modifying party as well as the modification policies of that particular N32 connection. This message validation may fail for a variety of reasons, such as:

- Missing patch by first IPX-provider in the path

- Invalid IPX-provider signature

- Attempted modification of a non-modifiable IE, according to the modification policy

What is missing in the current specification in TS 33.501 [11] is a clear ruleset describing how the receiving SEPP deals with different error scenarios, both in terms of local error handling as well as error signalling to the source of the N32-f message. In order to properly handle an incoming N32-f message, the SEPP needs to be able to determine which of the following actions 1.X and 2.X to take, should an error occur in one of the contained IPX-provider patches:

1. Local error handling

1.1 Drop message

1.2 Drop individual patch

1.3 Forward message

2. Error signalling

2.1 No error signalling

2.2 Error signalling to the original source

2.3 Error signalling to both the original source and the IPX-provider

Since 3GPP standards can hardly dictate how SEPPs of each individual operator are to behave, there needs to be a requirement for a configuration which allows to control the error handling described above for each individual scenario.

#### 4.2.10.2 Threat description

A SEPP that does not allow for configurational error handling will be unable to adapt to the specific tasks an operator outsources to its IPX-providers. Depending on what information elements the IPX-provider has to access and/or modify, a flawed or missing patch may render the whole message useless. In other scenarios, an IPX-provider may offer services that are "nice-to-have" but not strictly necessary. In these cases, discarding the message as a whole and waiting for retransmission may not be justified.

If the SEPP is not capable of conveying details on errors that occurred in received N32 messages, analysis and subsequent resolution of issues in the inter-PLMN communication will be unnecessarily difficult. Well-defined error signalling between SEPP and IPX provider can help to minimize operational effort. In order to avoid a potential attacker from exploiting the error signalling as a side channel to gain valuable information for an attack, it needs to be possible to configure whether error signalling is sent to the IPX providers or not.

#### 4.2.10.3 Potential security requirements

The SEPP shall support configurable error handling and error signalling per individual error cause and per individual N32 connection.

NOTE 1: The granularity of configuration for error handling and error signalling is not addressed in the present document.

The SEPP shall support configurable error signalling towards peer IPX providers.

# 5. Void

# 6 Solutions

## 6.1 Solution #1: Authorization of NF service access

### 6.1.1 Introduction

This clause specifies authorization procedures for authorizing NF service consumer to access services provided by NF service producer.

Granularity of authorization is per service based. In the case of authorization by NRF, prior to accessing a service defined in TS 23.502 [3], the NF service consumer requests a token from NRF. The token records and proves that NF service consumer is permitted to access the service provided by the service producer. The NF service producer verifies the token before executing the requested service. The authorization token can be reused to avoid requesting authorization for every service access.

NOTE 1: It is assumed that NRF authenticates the NF service consumer before authorization. The authentication method is not addressed in the present document.

### 6.1.2 Solution details

#### 6.1.2.1 Service authorization procedure for non-roaming scenarios



Figure 6.1.2.1-1: Service authorization procedure for non-roaming scenario

1) NF service consumer to NRF: Service Authorization Request (NF type and NF instance ID of service consumer, NF type and NF instance ID of service producer, NF service name). Service Authorization Request is included in Nnrf\_NFDiscovery\_Request [3] if the NF Service Consumer requests service authorization along with NF service discovery request.

2) NRF to NF Service Consumer: Authorization Result (Token).

NRF checks whether the access can be permitted according to the maintained authorization information. If the service can be authorized, NRF sends the result along with a token that proves this authorization. The token should include the NF type and NF instance ID of NF service consumer, the NF type and NF instance ID of NF service producer, the NF service name that will be accessed, and a credential such as MAC (Message Authentication Code) or digital signature. If the token can be reused within a period of time, the expiration date should also be included. If Service Authorization Request is included in Nnrf\_NFDiscovery\_Request, NF service producer should include Authorization Result in Nnrf\_NFDiscovery\_Request Response [3] which will be sent to the NF Service Consumer.

3) NF service consumer to NF service producer: NF Service Request (NF type and NF instance ID of service consumer, NF service name, Token).

4) NF service producer to NRF: Token Verification Request (Token).

If NF service producer is able to verify the token, step 4 and step 5 are skipped. Otherwise, NF service producer requests NRF to verify the token through Token Verification Request.

5) NRF to NF service producer: Token Verification Response.

NRF informs NF service producer the verification result. Token Verification Request and Response could introduce much overhead, thus it is recommended to verify the token by NF service producer itself.

6) NF service producer to NF service consumer: NF Service Response.

If the token is valid and the NF service Request is consistent with the information in the token, NF service producer executes the requested service and response to NF service consumer.

NOTE 1: Parameters of the messages and parameters in the token are not addressed in the present document.

NOTE 2: How to compute and verify the credential included in the token is not addressed in the present document.

#### 6.1.2.2 Authorization of NF service access for roaming scenario



Figure 6.1.2.2-1: Authorization of NF service access for roaming scenario

1) NF service consumer to NF service producer: NF Service Request (NF type and NF instance ID of service consumer, NF type and NF instance ID of service producer, NF service name).

2) NF service producer to NRF in Home PLMN: Authorization Request (NF type and NF instance ID of service consumer, NF type and NF instance ID of service producer, NF service name).

3) NRF in Home PLMN to NF service producer:

NRF in Home PLMN checks whether the access can be permitted according to the maintained authorization information (static policies). If the service can be authorized, NRF in Home PLMN sends the Authorization Response to the NF service producer.

4) NF service producer to NF service consumer:

If authorized, NF service producer executes the requested service and response to NF service consumer.

NOTE 1: The authentication mechanism between different PLMNs is not addressed in the present document.

### 6.1.3 Evaluation

Void.

## 6.2 Solution #2: Application layer protection based on JSON Object Signing and Encryption (JOSE)

### 6.2.1 General

Following aspects are considered when designing a solution for e2e protection of application layer information in the HTTP payload:

- Which protocol to use to secure JSON content

- Where to implement e2e security in the network

- Which JSON information elements to protect and what kind of protection is required

- Algorithms to use for protection and their negotiation between two Edge Proxy end points

- Key management aspects including key distribution to the Edge Proxies

- Protection mechanism that allows selective protection of the payload while allowing other unprotected payload to be modified by the intermediaries

### 6.2.2 Application layer protection based on JOSE

JOSE [4] provides a set of specifications to protect JSON based data structures. These include standards for:

- representation of integrity-protect JSON data based on public-key digital signatures as well as symmetric-key MACs using JSON Web Signing (JWS) [5],

- representation of encrypted data using JSON Web Encryption [6],

- specifying how to encode public keys as JSON-structured objects,

- specifying algorithms and algorithm identifiers using JSON Web Algorithm [7],

- specifying a means to protect private and symmetric keys via encryption.

JOSE is used to protect JSON based application content in SBA.

#### 6.2.2.1 JSON based IEs that require protection (WHAT)

JOSE framework will be used to integrity protect all the JSON IEs in the HTTP message payload. The JSON Web Signature [5] applies integrity protection either based on digital signatures (asymmetric protection) or Message Authentication Codes (symmetric protection). The resulting data structure is of JSON type and contains JWS Signature representing a digitally signed or MACed message payload.

JOSE framework will be used to confidentiality protect Authentication Vector (AVs), cryptographic keys, SUPI and Location data (e.g. Cell ID and Physical Cell ID) contained in the HTTP message. The JSON Web Encryption [6] is based on the use of Authenticated Encryption with Associated Data (AEAD) based encryption algorithms. Hence it applies both confidentiality protection and integrity protection on the Authentication Vectors.

#### 6.2.2.2 Integrity and Confidentiality protection schemes (HOW)

##### 6.2.2.2.1 Integrity protection based on JSON patch

There is a requirement for "e2e" integrity protection in conjunction with requirement for intermediaries to be able to modify the message in a verifiable way.



Figure 6.2.2.2-1: Message flow across N32 interface

1. The vSEPP receives an HTTP request.

2. The vSEPP encapsulates the HTTP request into a JSON object encapsulatedRequest consisting of three JSON objects:

- the request lines is put into an element called requestLine containing an element each for the method, the URI, and the protocol of the request received in step 1.

- the header of the request received in step 1 is put in into an element called httpHeaders, with one element per header of the original request.

- the body of the request received in step 1 is put into an element called http body.

Editor's note: how to deal with multipart messages is FFS.

The vSEPP includes its own identity and the encapsulatedRequest into a JSON object called partRequest as well to allow the hSEPP to identify the originator.

NOTE 1: Whether the vSEPP needs to include the first intermediary's ID in the partRequest for authorization of the first intermediary to perform modications is not addressed in the present document.

NOTE 2: Only authorized intermediaries are allowed to perform modifications. The authorization mechanisms are not addressed in the present document.

NOTE 3: Whether the hSEPP needs to include a policy regarding which elements are allowed to be changed by the first intermediary is not addressed in the present document.

The vSEPP integrity protects the complete partRequest using JWS.

The integrity protected partRequest is put into an array.

3. The vSEPP uses HTTP POST to send the encapsulated request to the first intermediary (visited network's IPX provider).

4. The first intermediary (e.g. visited network's IPX provider) checks the integrity and authenticity of the encapsulated request. It parses the encapsulated request and determine which changes are required. The first intermediary creates a JSON element called operations, taking the syntax and semantic from RFC 6902, that, when applied as a JSON patch to the encapsulated request, will result in the desired request. If no patch is required, the operations element is empty.

NOTE 4: Error handling in case of failed integrity check is not addressed in the present document.

The first intermediary creates a JSON element called partRequest that includes the intermediary's identity, and integrity protect the partRequest in a JWS.

NOTE 5: Whether the partRequest includes the hSEPP ID or the next intermediaries' ID to authorize further changes is not addressed in the present document. Inclusion of a policy is not required, because this would be under the home network's remit.

The integrity protected partRequest is appended to the array inside the encapsulated request created in step 2.

5. The first intermediary sends the encapsulated request to the second intermediary (home network's IPX) as in step 3.

6. The second intermediary checks the integrity and authenticity of the encapsulated request and the partRequest. It parses the encapsulated request, apply the modifications described in the partRequest and determine further modifications required to result in the desired request. These modifications are recorded as a further patch request. Further processing is like in step 4 (create a pertRequest and integrity protect).

NOTE 6: If a policy is included in step 2, how and whether the second intermediary can check that the first intermediary only changed allowable elements is not addressed in the present document.

7. The second intermediary sends the encapsulated request to the hSEPP as in step 3.

NOTE 7: The behaviour of the intermediaries is not normative, but the hSEPP assumes that behaviour for processing the resulting request.

8. The hSEPP checks the integrity and authenticity of the encapsulated request and the partRequests. The hSEPP checks whether the modifications performed by the intermediaries were permitted by policy. The hSEPP decapsulates the encapsulated request, verify signatures, apply the patches in the partRequests in order, perform filtering on the resulting request, and create a new HTTP request according to the "patched" encapsulatedRequest.

NOTE 8: Which signatures the hSEPP needs to verify is not addressed in the present document.

9. The hSEPP sends the HTTP request resulting from step 8 to the home network's NF.

10.-18. These steps are analogous to steps 1.-9., but treating the HTTP response like the HTTP request.

Below is an example to illustrate the elements in the JSON:

partRequest created by vSEPP

{

"partRequest": {

"previousSignature": "",

"originatorIdentity": "some MNO's SEPP",

"encapsulatedRequest": {

"requestLine": {

"method": "POST",

"URI": "APIroot/nausf\_auth/v1/ue\_authentications",

"protocol": "HTTP/2"

},

"httpHeader": {

"Accept: application/json",  
 "Content-Type: application/json",

"host: ": "hplmn.f.q.dn",

"content-length: ": 100

},

"body": {

"UE-id": "maguro\_suci",  
 "Serving network name": "some\_VPLMN",  
 "access\_type": "5G" }

},

"nextHopIdentity": "next intermediaries name"

}

}

partRequest created by Intermediary

{

"partRequest": {

"previous": "<signature of previous request entry in requesthistory array>",

"next": "<expected next originator>",

"originator": "intermediary name",

"operations": [

{

"op": "replace",

"path": "/HTTP-headers/Host",

"value": "HPLMN2.com"

},

{

"op": "replace",

"path": "/HTTP-headers/Content-Length",

"value": "131"

},

{

"op": "add",

"path": "/HTTP-body/new\_element",

"value": "value1"

}

]

}

}

The complete request with change history as will arrive at the hSEPP

{

"requestHistory": [

{

"integrityProtectedPartRequest": "protectedHeader.protectedPayloadIsPartRequestFromVSEPP.signature"

},

{

"integrityProtectedPartRequests": "protectedHeader.protectedPayloadIsPartRequestFromFirstIntermediary.signature"

},

{

"integrityProtectedPartRequests": "protectedHeader.protectedPayloadIsPartRequestFromFirstIntermediary.signature"

}

]

}

##### 6.2.2.2.2 Authorization of modifications based on JSON patch

The receiving SEPP requires a policy S which elements may be changed by the first IPX provider and a policy R which elements may be changed by the second IPX provider.

The sending SEPP informs the receiving SEPP of policy S either out of band or by including the policy (or link thereto) in the message itself. The receiving SEPP applies the policy that policies cannot be modified by intermediate IPX providers. Policy R is local to the receiving SEPP.

Each policy consists of a list of paths with the allowed operations. Below is an example:

"allowed-operations": [

{

"op": "replace",

"path": "/HTTP-headers/Host"

},

{

"op": "replace",

"path": "/HTTP-headers/Content-Length"

},

{

"op": "add",

"path": "/HTTP-body/new\_element"

}

]

The receiving SEPP verifies the modifications proposed by the first IPX in the incoming message against policy S. If a policy violation occurred, the receiving SEPP informs the sending SEPP of the policy violation in an error message with the appropriate HTTP error code and enough information for the sending SEPP to pinpoint the policy violation. The receiving SEPP discards the incoming message. The SEPP sending the original message (i.e. the one receiving the error message) applies the policy that policy violation messages cannot be modified by intermediate IPX providers.

The receiving SEPP verifies the modifications proposed by the second IPX in the incoming message against policy R. If a policy violation occurred, the receiving SEPP informs the second IPX provider out of band. The receiving SEPP also informs the sending SEPP of the fact that a policy violation occurred in an error message with the appropriate HTTP error code, and discard the incoming message. The SEPP sending the original message (i.e. receiving the error message) applies the policy that policy violation messages cannot be modified by intermediate IPX providers.

NOTE 1: What the sending SEPP will do when receiving an error code is not addressed in the present document.

##### 6.2.2.2.3 Authentication of intermediaries

Each intermediary has its own certificate infrastructure. The sending SEPP includes the root CA of the first IPX intermediary in its policy. The sending SEPP signs its policy.

##### 6.2.2.2.4 Rewriting of HTTP message into JSON-object

The solution "Integrity protection based on JSON patch" described in clause 6.2.2.2.1 also contains a solution for rewriting the HTTP message into a JSON object. Once the HTTP message has been rewritten in this way, it becomes more straight-forward to apply JOSE protection to selected elements of the message. Hence the rewriting process is of importance even for a solution without standardized modifications of intermediaries.

It thus seems reasonable to consider the following steps as a separate solution:

**Rewriting of HTTP-message into JSON-object:**

The vSEPP encapsulates the HTTP request into a JSON object encapsulatedRequest consisting of three JSON objects:

- the request line is put into an element called requestLine containing an element each for the method, the URI, and the protocol of the request received in step 1.

- the header of the request received in step 1 is put in into an element called httpHeaders, with one element per header of the original request.

- the body of the request received in step 1 is put into an element called http body.

NOTE 1: It is not addressed in the present document whether including the identity of the vSEPP in the JSON object is necessary.

#### 6.2.2.3 Void

### 6.2.3 Evaluation

Void

## 6.3 Solution #3: NF service registration process

### 6.3.1 Void

### 6.3.2 Solution Details

During initial provisioning and configuration of NF, NRF is configured with NF's public key and other information. And NF is configured with public key of NRF and other information.



Figure 6.3.2-1: Authentication of NF service registration

1) NF service consumer sends Nnrf\_NFManagement\_NFRegister Request message to NRF, signed by NF's private key and encrypted using public key of NRF. Registration request includes a nonce for replay protection.

2) NRF sends Registration response signed by NRF private key. Registration response includes NF certificate and other parameters.

3) Upon receipt if registration response, NF service consumer checks the integrity the Nnrf\_NFRegister\_Response by using public key of NRF decrypts the payload by NF Service consumer's private key.

### 6.3.3 Evaluation

Void

## 6.4 Solution #4: Authorization of NF service access

### 6.4.1 Introduction

During initial provisioning and configuration of NF, NRF is configured with NF's public key and other information. And NF is configured with public key of NRF and other information. During service registration, NF obtains certificate from NRF for its public key.

Service request and response uses TLS to establish a secure session between NF Service Consumer and NF Service Producer using their corresponding certificates. Upon successful Service request and response, a secure association is established between NF service consumer and NF service producer which provides secure session between the two.

Service request and response can function within same PLMN or across PLMNs. Subsequent clauses describe the detailed flow for each case.

### 6.4.2 Solution details

#### 6.4.2.1 Authorization of NF service access in the same PLMN



Figure 6.4.2.1-1: Authorization of NF service request in the same PLMN

1. The NF Consumer sends an NF Service request to NF producer. It contains a self-signed client ID. Service request also includes a client TLS [client\_hello] message for the NF Producer. The contents of TLS client\_hello are defined in the TLS specification.

2.

a. The NF Producer forwards the Signed Client ID as a payload to IsAuthorized message to NRF.

b. NRF verifies client ID signature. If the NF Consumer ID is successfully verified, NRF checks the stored NF profile information to determine whether the access can be permitted. If the service can be provided, NRF sends the verification result back to NF Service Producer. If verification is unsuccessful, NF Service producer does not proceed.

NOTE 1: IsAuthorized request and response message and VerifyCertificate messages as well as its format and parameters are not specified in the present document.

3. The NF Producer replies to the NF Consumer with TLS[server\_hello], which further includes information elements such as server\_hello, NF\_P\_Certificate, server\_key\_exchange, certificate\_request, server\_hello\_done. These information elements are defined in the RFCs for the TLS.

4. Upon receiving the TLS[server\_hello] message NF consumer forwards the message to its NRF through VerifyCertificate message. NRF verifies the NF Producer certificate received in TLS [server\_hello].

5. Upon successful verification of NF producer certificate, NF Consumer replies with TLS [client key exchange], which further contains information element such as client\_certificate (NF\_C\_Certificate), client\_key\_exchange, client\_certificate\_verify, change\_cipher\_spec, client\_finished, etc.

6. After receiving the TLS [client\_certificate] message NF consumer forwards the message to its NRF through VerifyCertificate message. NRF verifies the NF Consumer certificate received in TLS [client\_certificate] by NRF's public key.

7. NF producer sends Nrf\_Nf\_Service Response with TLS [Server\_finished] with change\_cipher\_spec to the NF Consumer.

8. Session Key (KSESSION\_C\_P) is used to secure further communication between NF consumer and producer.

#### 6.4.2.2 Authorization of NF service access in different PLMNs



Figure 6.4.2.2-1: Authorization of NF service access across PLMNs

1. The NF Consumer sends an NF Service request to NF producer in the home PLMN. It contains a self-signed client ID. Service request also includes a client TLS [client\_hello] message for the NF Producer. The contents of TLS client\_hello are defined in the TLS specification.

2. The NF Producer forwards the Signed Client ID as a payload to IsAuthorized message to NRF in home PLMN. hNRF acts proxy for NRF in serving PLMN and forwards the signed payload to it. Serving NRF verifies the Client ID signature. If the NF Consumer ID is successfully verified, NRF checks the stored NF profile information to determine whether the access can be permitted. If the service can be provided, NRF sends the verification result back to NF Service Produce through hNRF proxy. If verification is unsuccessful, NF Service producer does not proceed.

NOTE 1: IsAuthorized request and response message and VerifyCertificate messages as well as its format and parameters are not specified in the present document.

3. The NF Producer replies to the NF Consumer with TLS[server\_hello], which further includes information elements such as server\_hello, NF\_P\_Certificate, server\_key\_exchange, certificate\_request, server\_hello\_done. These information elements are defined in the RFCs for the TLS.

4. NF Service producer's certificate is sent to NRF in HPLMN for verification through the VerifyCertificate message. Serving NRF acts as a proxy and just transfer the payload to Home NRF. The NRF in HPLMN verifies the NF producer's certificate received in TLS [server\_hello].

5. NF Consumer replies with TLS [client key exchange], which further contains information element such as client\_certificate (NF\_C\_Certificate), client\_key\_exchange, client\_certificate\_verify, change\_cipher\_spec, client\_finished etc.

6. NF Service consumer certificate is sent to NRF in SPLMN for verification through the VerifyCertificate message. HPLMN NRF acts as a proxy and just transfer the payload to Serving NRF. The NRF in Serving PLMN verifies the NF Consumers certificate received in TLS [client\_certificate].

7. NF producer sends Nrf\_Nf\_Service Response with TLS [Server\_finished] with change\_cipher\_spec to the NF Consumer.

8. Session Key (KSESSION\_C\_P) is used to secure further communication between NF consumer and producer.

### 6.4.1 Evaluation

Void

## 6.5 Solution #5: Using mediation services with end-to-end encryption

### 6.5.1 Generic

The scenario that is depicted in the figure below is a scenario with two MNOs, MNO A and MNO B and two IPX providers, IPX A and IPX B. The IPX provider A provides mediation services for MNO A and IPX provider B provides mediation services for MNO B. Both MNOs have one network function (NF), which is left unnamed. This solution provides two possible implementations, one where two SEPPs communicate securely with each other via HTTPS or TLS, and one where JOSE is used for the protection of the messages between two SEPPs.

### 6.5.2 End-to-end encryption using HTTPS or TLS

In this version of the solution, it is assumed that the SEPPs themselves use HTTPS for providing end-to-end security. In this case, the solution works as follows:

1. The SEPP A receives a HTTP(S) Request from NF A as usual.

2. In case this request contains sensitive information according to clause 9.1.3.3, the SEPP A performs an action to hide these fields for the mediation service. This action is not to be standardized. Some examples are:

a. Replacing the values of these with some other values, e.g. a hash of the value. The SEPP A stores the hash of the value and the corresponding value temporarily.

b. Entirely removing the fields from the message and storing bot the header and the value temporarily.

c. Encrypting the fields using some proprietary mechanism.

3. The SEPP A invokes the Mediate service running at the IPX A by sending a MediateAndReturn Request message to the IPX provider. The MediateAndReturn Request contains the message that was received from the NF A and has its sensitive information removed or hidden according to step 2.

4. The Mediation services performs its mediation.

5. The mediation service sends the MediateAndReturn Response message, which contains the mediated message, to the SEPP A.

6. Upon reception, the SEPP A reinserts the sensitive information. This action depends on how the SEPP A has removed or hidden the sensitive information and can be entirely proprietary.

7. The SEPP A then sends the mediated version of the original NF A's request to the SEPP B over HTTPS. So the request would look like a request that came from NF A apart from the mediated fields.

8. The SEPP B receives the request, and if mediation is deemed necessary, the SEPP B also removes or hides the sensitive fields from the message.

9. The SEPP B then invokes the Mediate service running on IPX B by sending a MediateAndReturn Request message to IPX B.

10. The mediation service performs its mediation.

11. The mediation service sends the MediateAndReturn Response message, which contains the mediated message.

12. The SEPP B re-inserts the sensitive information.

13. And finally, SEPP B sends the request to NF B.

In short, the solution relies on standard HTTP and HTTPS. In between the steps 2 and 6, the SEPP A will either have to keep state or use an encryption / decryption mechanism. In between the steps 8 and 12, the SEPP B has a similar task. In case IPX provider hosts the SEPP (e.g. for small operators), the steps 2-6 would probably be left out altogether.

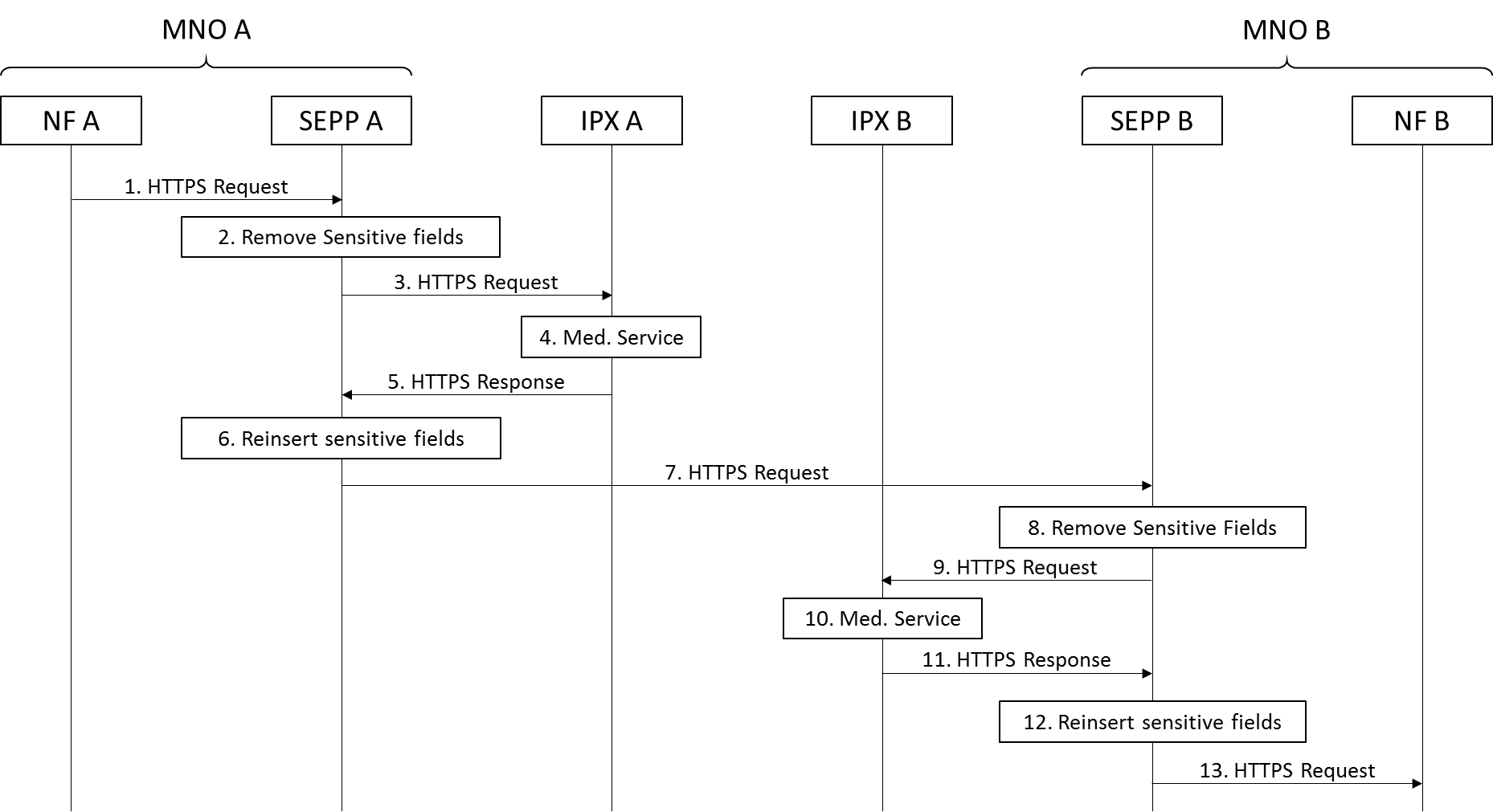


Figure 6.5.2-1: Mediation service using HTTPS

### 6.5.3 End-to-end security using JOSE

In this version of the solution, it is assumed that the SEPPs themselves use HTTP request with an encrypted JOSE payload for providing end-to-end security. In this case, the solution works as follows:

1. The SEPP A receives a HTTP(S) Request from NF A as usual.

2. The SEPP A takes the request and wraps the whole request into a JSON format. So, the request headers go into a field called 'HTTPRequestHeader', a binary blob goes into a field called 'BinaryBlob' and the session cookie goes into a field called 'SessionCookie'. Then, the SEPP determines whether the message contains sensitive information according to clause 9.1.3.3 and performs an action to hide these fields for the mediation service. This action is not to be standardized. Some examples are:

a. Replacing the values of these with some other values, e.g. a hash of the value. The SEPP A stores the hash of the value and the corresponding value temporarily.

b. Entirely removing the fields from the message and storing bot the header and the value temporarily.

c. Encrypting the fields using some proprietary mechanism.

3. The SEPP A invokes the Mediate service running at the IPX A by sending a MediateAndReturn Request message to the IPX provider. The MediateAndReturn Request contains the message that was received from the NF A and has its sensitive information removed or hidden according to step 2.

4. The Mediation services performs its mediation

5. The mediation service sends the MediateAndReturn Response message, which contains the mediated message, to the SEPP A.

6. Upon reception, the SEPP A reinserts the sensitive information. This action depends on how the SEPP A has removed or hidden the sensitive information and can be entirely proprietary. The SEPP A encrypts the message using standard JOSE using the target SEPP's public key.

7. The SEPP A then sends the mediated version of the original NF A's request to the SEPP B over HTTP.

8. The SEPP B receives the request, decrypts the request, and if mediation is deemed necessary, the SEPP B also removes or hides the sensitive fields from the message.

9. The SEPP B then invokes the Mediate service running on IPX B by sending a MediateAndReturn Request message to IPX B.

10. The mediation service performs its mediation.

11. The mediation service sends the MediateAndReturn Response message, which contains the mediated message.

12. The SEPP B re-inserts the sensitive information.

13. And finally, SEPP B reconstructs the HTTP Request from the JSON fields and sends the HTTP Request to the NF B.

In short, the solution relies on standard HTTP and JOSE. A complicating factor is that the SEPPs will have to convert the **entire** HTTP Request into a JSON object, which in itself will be contained in another HTTP request. The receiving SEPP will have to do the reverse conversion. Like in the solution based on HTTPS, in between the steps 2 and 6, the SEPP A will either have to keep state or use an encryption / decryption mechanism. In between the steps 8 and 12, the SEPP B has a similar task. In case IPX provider hosts the SEPP (e.g. for small operators), the steps 2-6 would probably be left out altogether.

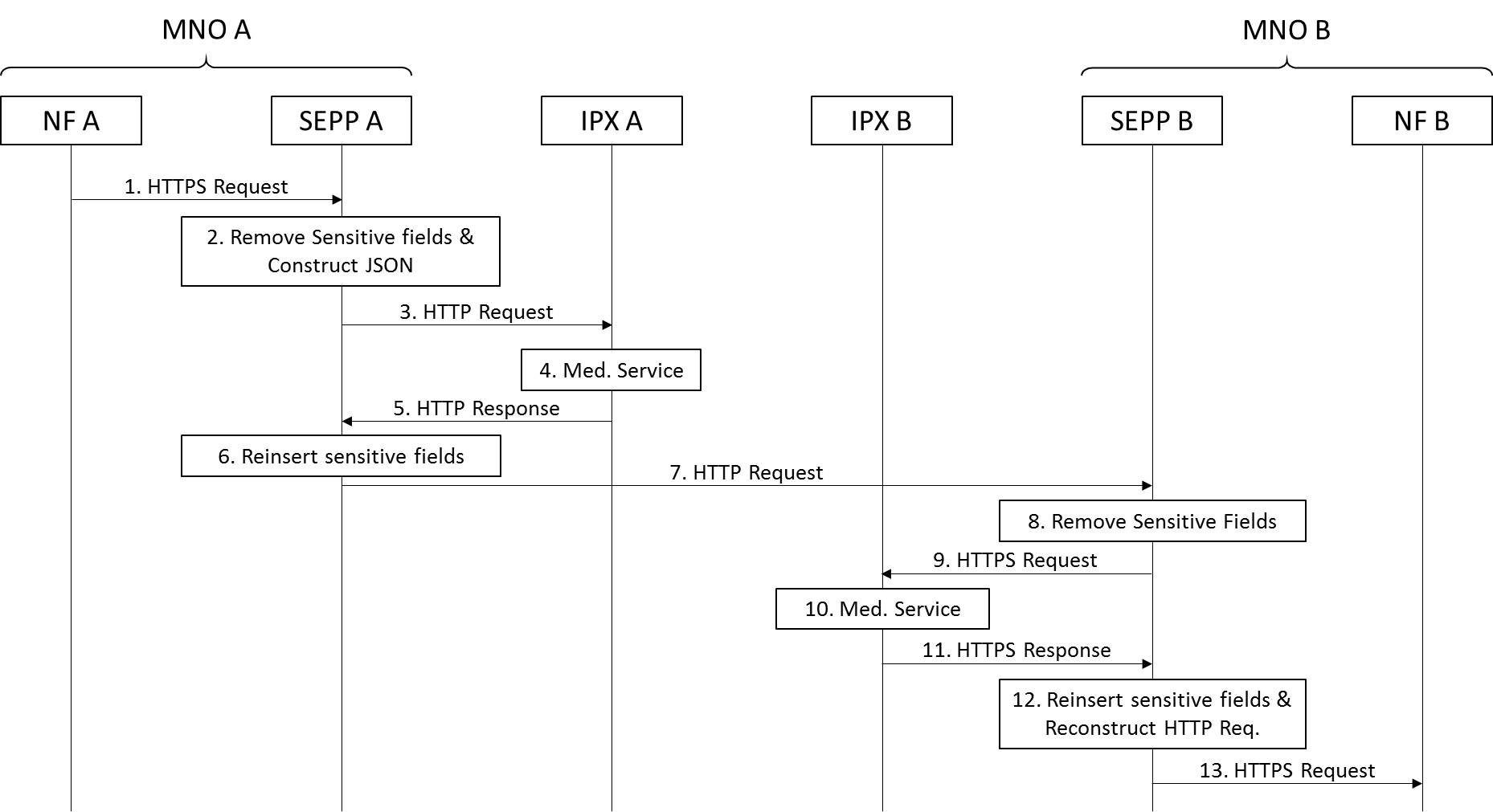


Figure 6.5.3-1: Mediation service using JOSE end-to-end encryption

### 6.5.4 Migration paths after accepting this solution

One possible drawback of the solution is that it will 'stick' even past its due date. The solution provides therefore proposes to name the mediation service in 3GPP specs. By standardizing a name, it becomes possible to migrate to a newer service by using a service under the newer name and migration remains under control of 3GPP. Investments in existing services and SEPPs will not be affected and IPX providers can distinguish themselves by operating the newer service.

### 6.5.5 Possible deployments

In this solution, it is always assumed that the SEPP is located in the MNO domain and the mediation service is located in the IPX domain. As a drawback, there is an additional message exchange between the SEPP in the MNO domain and the mediation service in the IPX domain. There are two possible deployments that alleviate this problem:

- Colocation of the SEPP in the IPX domain: This is a likely deployment scenario for smaller operators, but should not be the standard preferred option.

- Colocation of the mediation service in the MNO domain: In this deployment, the mediation service is run as a service on premise for the MNO. This is a deployment scenario for larger operators, but will depend on the willingness of IPX providers to run their service offsite.

### 6.5.6 Evaluation

This solution has a number of drawbacks:

- It introduces additional messages (in total 4 if mediation is used twice);

- The SEPP needs to either keep state for removing / reinserting the sensitive fields;

- TLS handshake for HTTP request will take time and messages;

- MNO needs to operate both a connection to IPX Mediation Service and a 'direct' connection to MNO peers (both can be over the same IPX network, but does not have to go there).

On the other hand:

- It mostly reuses standard HTTP, etc. making it relatively easy to implement;

- Works with end-to-end security, also if different from what is presented here;

- Offers a migration path;

- Does not expose the sensitive information to the IPX provider, while making mediation services possible;

- Can be specified within the timeframe available;

- Allows IPX providers to continue to offer their services, even if end-to-end security is used.

## 6.6 Solution #6: Policies for protection on the N32 interface

### 6.6.1 Void

### 6.6.2 Solution details

A message protection policy determines which part of a certain message is integrity protected, which part of a certain message is confidentiality protected, and which part of a certain message is modifiable by IPX providers. For application layer protection of messages on the N32 interface, the SEPP applies message protection policies.

NOTE 1: The specification of the protection policy and whether 'per subscription' is relevant is not addressed in the present document.

If the SEPP neither has nor obtains a policy applicable for a specific message, the SEPP applies a default policy.

NOTE 2: Which IEs are protected according to the default policy is not addressed in the present document.

For the protection of a specific message, an NF may include a message protection policy applicable for that specific message into the message.

The SEPP retrieves a message protection policy from the NRF, if operator configuration requires, e.g. when the SEPP has no message protection policy available for a message to be sent on N32.

NOTE 3: It is not addressed in the present document whether the procedure is a service offered by the NRF.

The SEPP also supports local configuration of message protection policy, e.g., by OA&M system. Configuration may occur during initial provisioning of SEPP or through dynamic updates any time the policy needs an update e.g., due to network configuration change.

The SEPP sends message protection policy error messages to NFs or the NRF if operator configuration requires, e.g. for the case that the SEPP has no policy applicable for a specific message.

It is up to operator configuration how the SEPP behaves if more than one policy applicable for a specific message are available to the SEPP.

### 6.6.3 Evaluation

Void

## 6.7 Solution #7: Signaling based provisioning of message protection policy in partner SEPPs

### 6.7.1 Void

### 6.7.2 Solution details

The signaling based provisioning and update of the message protection policy in a roaming partner SEPP allow the two SEPPs to share each other network's protection policy information.

This scheme is useful in scenarios where a local SEPP obtains its message protection policy information through an out of band mechanism such as via the OA&M interface or from a central repository, and not via in-band scheme such as for e.g. embedded in HTTP messages from Network functions themselves.

When the local SEPP in a network gets its initial copy of the message protection policy or if there is an update in the network that resulted in an update to its copy of the message protection policy the local SEPP initiates a handshake with each of its remote partner SEPPs in different networks. It provides its version of the protection policy to each of them. In the response, the SEPPs in the remote network may decide to provide the latest version of its message protection policy.

A mutually authenticated TLS connection is used for protecting SEPP to SEPP signaling messages over N32. TLS is e2e between two SEPPs with no intermediaries in between.

In the following illustration, Registration Request message flow from clause 13.5 in TS 33.501[11] is reused.



Figure 6.7.2-1: Signaling based provisioning of message protection policy in partner SEPPs

1. The SEPP which initiated the TLS connection sends a Registration Request message to the responding SEPP including the its message protection policy for protecting the NF service messages belonging to its network.

2. The responding SEPP stores the received message protection policy for network A.

3. The responding SEPP sends a Registration Response message to initiating SEPP including its message protection selected security mechanism for protecting the NF service messages belonging to its network.

4. The initiating SEPP stores the message protection policy for network B.

### 6.7.3 Evaluation

Void

## 6.8 Solution #8: Inter PLMN routing and TLS: Solution Options

### 6.8.1 Introduction

This solution addresses Key Issue #18: Inter-PLMN routing and TLS.6.8.2 Bump in the TLS

In this solution option, TLS seems to be end-to-end from an NF point of view, but is forced to be terminated in the SEPPs instead. This is made possible by providing the SEPPs with certificates representing the remote PLMN, signed by their own CA. This either requires a large number of pre-provisioned certificates, certificate creation on the fly, multilevel wild card certificates or the certificates would have to include the actual IP of SEPP.

**Pros:** No changes to current specifications.

**Cons:** Unorthodox solution, should work technically but may introduce implementation issues. The solution needs to be repeated for possible SEPP-IPX, IPX-IPX TLS usage.

### 6.8.3 TLS tunnel or VPN from NF to SEPP

In this solution, the inter-PLMN service request would be in http plain text but sent to the SEPP over TLS (stunnel, etc.).

Alternatively, the transport layer protection does not need to be a TLS tunnel but could be any "VPN" connection that can be authenticated and that provides sufficient security.

**Pros:** Solves TLS issues in SEPP, same approach could be used between PLMN and IPX.

**Cons:** Adds requirements for NFs as a separate setup for the NF-SEPP tunnel is needed. It would not work if QUIC is introduced in a future release.

### 6.8.4 Using local SEPP FQDN in request URI

In this solution option, the URI would always point to the next-hop and hence TLS could be terminated in an ordinary way. The actual target NF FQDN could be carried in some other place in the header or body of the message.



Figure 6.8.4-1: Message routing in roaming scenarios

NOTE: IPXs not depicted for the sake of clarity.

The message flow for this solution is the following, see also Figure 6.8.4-1:

1) The consumer NF initiates the service request towards the local vSEPP URL, target PLMN FQDN is carried in message header or body.

2) The vSEPP forwards the service request towards the next-hop URL. The next-hop URL may be either an IPX proxy or the SEPP of the roaming partner.

3) The hSEPP forwards the service request to the URL of the target producer NF.

4) The producer NF sends the response back to the URL of the local hSEPP.

5) The local hSEPP forwards the service request towards the next-hop URL. The next-hop URL may be either an IPX proxy or the SEPP of the roaming partner.

6) The vSEPP forwards the response to the consumer NF

**Pros:** Solves TLS termination issues.

**Cons**: Requires changes to current SA2 and CT4 specifications.

### 6.8.5 Mapped FQDN in request URI

This solution introduces added functionality to SEPP.

During NF service discovery in inter-PLMN communication, the involved elements perform the following actions:

1. The consumer NF in the serving PLMN requests the service discovery from its own NRF for a producer NF in the home PLMN.

2. The NRF sends a discovery request to the home NRF. Inter-PLMN message exchange between NRFs happens through cSEPP and pSEPP via N32.

3. The home NRF returns the producer NF's FQDN, e.g. "pNF.foreign\_network.tld", and sends the response back to the requesting NRF via the pSEPP.

4. Depending on operator policy, the pSEPP may rewrite the producer NF's FQDN inside the discovery response if necessary, for topology hiding, e.g. "rewritten\_pNF.foreign\_network.tld".

5. The pSEPP transfers the discovery response via N32 to the cSEPP.

6. The cSEPP, of which the internal interface has FQDN "cSEPP.own\_network.tld", rewrites the producer NF's FQDN to a new unique FQDN consisting of a unique label as first element of FQDN and the cSEPP FQDN as suffix: "label.cSEPP.own\_network.tld". The "label" can be any unique label that the cSEPP can translate back to original producer NF's FQDN. This unique label in the cSEPP that maps to original producer NF's FQDN can be called a Rendezvous Point at the cSEPP.

7. The cSEPP sends the discovery response including the new FQDN to its own NRF, which will in turn forward this information to the consumer NF.

8. The consumer NF resolves the received FQDN to the IP address of the cSEPP.

9. The consumer NF sets up a TLS connection to the IP of cSEPP. The hostname is the FQDN received from local NRF.

10. The cSEPP authenticates towards the consumer NF using a wildcard certificate for "\*.cSEPP.own\_network.tld"

11. The cSEPP rewrites the hostname in the HTTP request, e.g. to "rewritten\_pNF.foreign\_network.tld"

12. The pSEPP rewrites the hostname in the HTTP request, e.g. to "pNF.foreign\_network.tld", establishes a TLS connection to producer NF unless it already exists and sends the message to the corresponding producer NF.

Similarly whenever a callback URI is provided during subscription for notification(s), or other scenarios (e.g. PDU session creation at H-SMF by a V-SMF where V-SMF provides a callback URI of the PDU session representation at the V-SMF), the SEPP at the receiving side rewrites the authority part of the URI with a new unique FQDN consisting of a unique label and cSEPP FQDN as suffix.

**Pros:** Solves TLS termination issues without the need for multilevel wildcard certificates

**Cons:** Additional functionality to SEPP. Alsorequires either a FQDN mapping table to be kept in SEPP or a new encoding scheme for flattening the original remote FQDN to an acceptable DNS label [6].

### 6.8.6 Evaluation

For Key issue #18: Inter-PLMN routing and TLS, solution option "Using local SEPP FQDN in request URI" of Solution #8 is SA3's recommendation for normative work.

## 6.9 Solution #9: N32 message anti-spoofing within the SEPP

### 6.9.1 Void

### 6.9.2 Solution Details

As the first point of contact for incoming signalling messages on N32, it is the responsibility of the Security Edge Protection Proxy to protect the PLMN and its NFs from malicious messages. Such messages might contain spoofed JSON content within the HTTP body in order to obtain unauthorized service access or obtain information about the topology of a given PLMN. Therefore, SEPP is able to perform anti-spoofing on incoming messages, enforcing the following plausibility checks:

- Matching of MNC and MCC: If MCC and MNC or PLMN-ID is contained in an incoming message on N32, the receiving SEPP verifies that the combination of MCC and MNC is valid.

- Validation of the originating SEPP's certificate: The receiving SEPP validates the TLS certificate of the originating SEPP. This includes matching of the originating FQDN with the one that the certificate was issues for.

- Matching of MNC and MCC and SEPP FQDN: If MCC and MNC or PLMN-ID is contained in an incoming message on N32, the receiving SEPP verifies that the originating SEPP's FQDN matches the one expected for the contained PLMN-ID.

- Matching of SUPI and expected SUPI-range: If the SUPI is contained in an incoming message on N32, the receiving SEPP verifies that it is either within the operators own SUPI-range or the pre-configured SUPI-range of an associated roaming partner.

In case any of the above-mentioned checks fail, the SEPP discards the incoming message.

### 6.9.3 Evaluation

Void

## 6.10 Solution #10: Mitigation against fraudulent registration attack between SEPPs

### 6.10.1 Introduction

This solution addresses Key Issue #3: Fraudulent registration message over N32 interface.

### 6.10.2 Solution Details

To mitigate this attack, the VSEPP generates a secret based on the certificate which negotiated between SEPPs and the PLMN ID (MCC and MNC in NF ID), or the VSEPP generates a signature by using the PLMN ID and its private key. The VSEPP sends the secret or signature together with the PLMN ID to the HSEPP through the N32 message. The HSEPP verifies the secret based on the certificate and the PLMN ID, or verify the signature based on the serving network's public key and the PLMN ID. The HSEPP sends a response to the VSEPP through the N32 message.

### 6.10.3 Evaluation

Void

## 6.11 Solution #11: Security policy provisioning for SEPP

### 6.11.1 Void

### 6.11.2 Solution Details



Figure 6.11.2-1: Security policy provisioning for SEPP

1. The cSEPP (the SEPP in the consumer PLMN) sends a service discovery request to the pSEPP (the SEPP in the producer PLMN). The service discovery request message includes the name of the required service (service 1) and the PLMN-ID of the producer PLMN.

2. The pSEPP sends the discovery request message to the pNRF. The pNRF generates the authorization token of the service 1, and send it to the pSEPP.

3. The pSEPP sends a security policy request message to the pPCF. The pPCF returns the security policy of the service 1 and the security policy of service authorization to the pSEPP.

4. The pSEPP sends the security policy of service 1, the security policy of service authorization and the protected (e.g. encrypted) token in the service discovery response message. The pSEPP protects the token based on the security policy of service authorization.

4a. Upon receipt of the service discovery response message from the pSEPP, the cSEPP verifies the token in the response message based on the security policy of service authorization. In addition, the cSEPP may send a security policy notification message based on its own configuration, which includes cSEPP-supported security policies.

### 6.11.3 Evaluation

Void

## 6.12 Solution #12: End-to-end data protection in hop-by-hop network communication links

### 6.12.1 Introduction

NOTE 1: Entities generating and distributing the public/private keys mentioned in this solution, and call flows for the proposed solution are not addressed in the present document.

Consider a hop-by-hop network communication link, e.g., in a 5G or LTE IPX network, whose nodes correspond to IPX entities. In particular, such a link corresponds to N32 interface in 5G. Assume that data is arranged in a signaling message as a sequence of information elements (IEs), e.g., as a sequence of AVPs in the LTE Diameter protocol. For example, IEs can be implemented as JSON elements.

Signaling messages go from source to destination via specified intermediate peers which can be authorized to read or modify (change or delete) the IEs or can add the new ones. The communication links thus have an intrinsic hop-by-hop nature and, as such, can be protected in the hop-by-hop manner (e.g., by TLS tunnels over http in 5G or by IPSec tunnels in LTE). However, such hop-by-hop protection does not ensure end-to-end integrity protection with non-repudiation and traceability of changes. Neither does it ensure that only authorized nodes should perform changes in a signaling message. Neither does it ensure that only authorized nodes should have read access to sensitive IEs.

### 6.12.2 Integrity protection with non-repudiation and traceability of changes

The solution described in this clause ensures end-to-end integrity protection with non-repudiation by using hash functions and digital signatures. Each node receives only the last signaling message meant to be received by that node, after all the changes performed by previous nodes along the link, along with some auxiliary information.

The signaling message received by any node along the link is verified as authentic if and only if all included digital signatures are verified as valid. In that case, the receiving node also learns and verifies as authentic all the change operations performed by previous nodes in the respective received signaling messages. It also verifies as authentic any information about the nodes (e.g., their identity attributes) associated with the respective digital signatures. Non-repudiation is ensured by digital signatures, with respect to this associated information. Digital signatures are performed only by the nodes adding or modifying the IEs in a signaling message.

The solution is defined as follows:

1. If a node adds a new IE to the sequence, then it associates to it an index that is different from the indexes of other IEs in the sequence, before sending the new IE to the next node. In particular, this relates to the source node.

2. If a node modifies a received IE, by changing or deleting its value, then it associates to the modified IE a hash0 value of its original value, without modifying its index, before sending it to the next node. Here, the hash0 function needs to be collision-resistant, e.g., a cryptographic hash function, or, if IE is very short, an identity function, which is not one-way.

3. If a node does not modify a received IE, then it forwards it to the next node in the same form.

4. If a node neither adds new IEs nor modifies the existing IEs in a received signaling message, then it forwards the received signaling message as a whole to the next node.

5. If a node adds or modifies at least one IE in a signaling message, then it computes a hash value of the concatenation of the hash0 values of all added or modified IEs including their indexes. Then, it computes a digital signature on the resulting hash value, by applying the respective private key, and adds a new IE containing the digital signature together with the indexes of the added or modified IEs. Here, the hash function needs to be a cryptographic hash function, which is both collision-resistant and one-way.

6. Each computed digital signature should include anti-replay protection mechanisms (e.g., based on nonces).

7. Upon receiving a signaling message, each receiving node verifies all the digital signatures included in the signaling message, by iteratively exploiting the associated hash0 values of the original values of modified IEs and by applying the respective public keys for verification.

8. The method can be applied to all or to only selected IEs in a signaling message, where the selection should be performed by the nodes adding new IEs.

### 6.12.3 Integrity protection with non-repudiation, traceability of changes, and authorization

In the solution described in clause 6.12.2, each receiving node can locally store the authorizations of all previous nodes for performing the changes in a signaling message and can then verify the consistency of the traced operations by comparing them with the stored authorizations. However, the local storage and update of authorizations can be impractical, especially if nodes belong to different domains. If classical digital signatures are used, then an inter-operator public-key infrastructure (PKI) is required, which may be impractical.

A more effective and efficient method, using attribute-based cryptosystems is described in the following:

- Authorization rights of a node for performing the changes in a signaling message are expressed by an access policy in terms of the node attributes (e.g., their identity or domain attributes).

- Such an access policy is (dynamically) embedded in a digital signature of a node by using attribute-based signatures (ABS) or identity-based signatures (IBS) [3]. The node attributes are embedded in the node private key for signing.

- In ABS, there is a common public key for signature verification and a multiplicity of private signing keys.

- An ABS signature can be verified as valid if and only if the embedded node attributes satisfy the embedded access policy and the signed information is authentic.

- Such write authorization rights are then verified by verifying a digital signature and by checking if the access policy embedded in the digital signature is compliant with the write access policy associated with an IE (e.g., as an integral part of IE value).

### 6.12.4 Confidentiality protection with authorization

Confidentiality of sensitive IEs can be protected by using encryption. The objective is that the source node, or any intermediate node adding new sensitive IEs to a signaling message, should encrypt these IEs in such a way that only the further nodes along the link that are authorized to read these IEs (including the destination node) are in possession of the respective private decryption key. Classical solutions are not satisfactory due to impractical key management.

A more effective and efficient method, using attribute-based cryptosystems is described in the following:

- Confidentiality of selected IEs with authorized access to decryption keys is achieved by applying attribute-based encryption (ABE) or identity-based encryption (IBE), where the relevant read access policy is (dynamically) embedded in ciphertext and the node attributes in its decryption key (ciphertext-policy ABE – CP-ABE) [9]. Preferably, both CP-ABE and ABS should use the same public and private keys (ABES) [10]. Alternatively, the relevant node attributes is embedded in ciphertext and the access policy in the node decryption key (key-policy ABE – KP-ABE) [8].

- In ABE, there is a common public key used for encryption and a multiplicity of private decryption keys.

- In ABE, the decryption can work if and only if the embedded node attributes satisfy the embedded access policy. This means that the read authorization rights are thus ensured automatically.

- ABE should be used for establishing a common shared key for a symmetric-key encryption/decryption. The same key is automatically shared by the encryption node all authorized decryption nodes. Moreover, this key can be used as static, together with a key-derivation function in order to generate dynamic session keys for encryption.

- The integrity protection of modified and re-encrypted IEs should be performed on ciphertexts, in order to enable for the nodes that are not authorized to decrypt/read encrypted IEs to verify the integrity of these IEs.

## 6.13 Solution #13: Content and structure of protection policies

### 6.13.1 Introduction

A protection policy determines which part of a certain message is integrity protected, which part of a certain message is confidentiality protected, and which part of a certain message is modifiable by IPX providers. For application layer protection of messages on the N32 interface, the SEPP applies message protection policies.

In this solution, the following protection policies are introduced:

- Data-type encryption policy that specifies which data types need to be confidentiality protected;

- A modification policy that specifies which IEs are modifiable by intermediaries.

In addition, there is a mapping between the data-types in the data-type encryption policy and the IEs in NF API descriptions which is given in a NF-API data-type placement mapping. For each message, the resulting policy is the combination of the data-type protection policy with the date-type mapping and the data field modification policy. The resulting policy applies to the message after rewriting by the SEPP.

In this solution, it is not specified how the data-type mapping gets to the SEPP.

### 6.13.2 Data-type encryption policy

The SEPP contains an operator controlled protection policy that specifies which types of data is encrypted. The data-types defined at this moment are the following:

- Data of the type 'SUPI'

- Data of the type 'location data'

- Data of the type 'key material'

- Data of the type 'authentication token'

- Data of the type 'other data requiring encryption'

This policy is on a per roaming partner basis.

The policy contains an identifier that identifies the policy.

### 6.13.3 NF API data-type placement mapping

Each NF API data-type placement mapping contains the following:

- Which IEs contain data of the type 'SUPI'

- Which IEs contain data of the type 'location data'

- Which IEs contain data of the type 'key material'

- Which IEs contain data of the type 'other data requiring encryption'

Where the location of the IEs refers to the location of the IEs after the SEPP has rewritten the message for transmission over N32.

An NF API data-type placement mapping furthermore contains data that identifies the NF API, namely:

- The name of the NF

- The version

- An identifier

NOTE: Larger networks can contain multiple NFs with the same API, e.g. three AMFs. The NF API policy applies to all NFs with the same API.

The NF API data-type placement mapping resides in the SEPP.

### 6.13.4 Modification policy

The modification policy specifies which IEs can be modified by an IPX provider of the sending SEPP. The IEs refer to the IEs after the SEPP has rewritten the policy.

This policy is specific per roaming partner and per IPX provider that is used for the specific roaming partner.

This policy resides at the SEPP.

### 6.13.5 Evaluation

This solution achieves the following:

- The ability to configure the usage of encryption by the operator; and

A mechanism to activate and deactivate NF policies in the SEPP.

## 6.14 Solution #14: Provisioning and negotiation of protection policies

### 6.14.1 Introduction

In order for the SEPP to apply the protection policies, it needs to be provisioned with the:

- Data-type encryption policy;

- NF API Data-type placement mapping;

- The modification policy.

This solution proposes to manually configure the SEPP.

### 6.14.2 Provisioning of the policies in the SEPP

The SEPP contains an interface that the operator can use to manually configure the protection policies in the SEPP.

The SEPP is able to store and process the following policies for outgoing messages:

- A generic data-type encryption policy;

- Roaming partner specific encryption policies that will take precedence over a generic data-type encryption policy if present;

- One NF API Data-type placement mapping;

- Multiple modification policies, to handle modifications that are specific per IPX provider and modification policies that are specific per IPX provider and roaming partner.

The SEPP is also able to store and process the following policies for incoming messages:

- Roaming partner specific encryption policies;

- A modification policies per roaming partner that specifies which fields can be modified by which IPX providers.

### 6.14.3 Negotiation of protection policies

In addition to statically configuring the protection policies between roaming partners, two SEPPs can also exchange their modification policies in the initial handshake. In that case, both SEPPs include their modification policies in the initial handshake and store the received policies.

### 6.14.4 Evaluation

This solution describes how the protection policies are provisioned in the SEPP.

## 6.15 Solution #15: Service access authorization in the delegated "Subscribe-Notify" interaction scenarios

### 6.15.1 Introduction

This solution addresses key issue #28: Service access authorization in the delegated "Subscribe-Notify" interaction scenarios.

### 6.15.2 Solution details

This authorization scheme is useful in the delegated "Subscribe-Notify" interaction scenarios, which is a NF\_A subscribes to NF Service offered by NF\_B on behalf of NF\_C.



Figure 6.15.2-1: Service access authorization in the delegated "Subscribe-Notify" interaction scenarios for non-roaming.

0. As precondition, the NF\_A and the NF\_C perform mutual authentication and determines that the NF\_A subscribes services on behalf of the NF\_C, and provides its own instance id and Notification URI to the peer.

1. The NF\_A sends the Nnrf\_AccessToken\_Get Request to the NRF. The token request contains the instance ID of the NF\_A, the Notification URI of the NF\_A, the Notification URI of the NF\_C and an indication (Expressed as "Indication\_delegated") that indicates the token request is for the delegated "Subscribe-Notify" interaction scenarios which is the NF\_A subscribes services on behalf of the NF\_C.

2. Upon reception of the Nnrf\_AccessToken\_Get Request, the NRF obtains the NF Instance ID of the Service Consumer (NF\_A), the Notification URI of the Service Consumer (NF\_A and NF\_C) and the indication\_delegated from the NF\_A. The NRF then determines whether the NF\_A is authorized to subscribe services on behalf of the NF\_C, and whether the NF\_C is authorized to receive the NF service provided by the NF\_B, based on the locally configured policies or authorization information. If the authorization is successful, the NRF generates a token which includes the instance ID of NF\_A, the Notification URI of the NF\_A, the Notification URI of the NF\_C, and the indication\_delegated. The token is used to indicate that the NF\_A is authorized to subscribe services on behalf of the NF\_C, and is used to indicate that the NF\_C is authorized to receive the NF service provided by the NF\_B.

3. The NRF sends the Nnrf\_AccessToken\_Get Response to the NF\_A with the token.

4. The NF\_A sends a subscription request, which contains the token and the associated notification endpoint of the NF\_C and the NF\_A (i.e. the Notification URI of the NF\_C and the NF\_A), to the NF\_B.

5. Upon reception of the subscription request, the NF\_B verifies the integrity of the token and verify the claims in the token. If successful and determine that whether the NF\_C is authorized to receive the NF\_B service notification based on the instance ID and the Notification URI of the NF\_C contained in the token claims.

6. If the verification of the token is successful, the NF\_B sends an authorization notification to the NF\_C, which indicates the NF\_C is authorized to receive the NF service provided by the NF\_B. The authorization notification contains the token.

7. The NF\_C verifies the integrity of the token also. If the token verification is successful, the NF\_C shall send an authorization reply back to the NF\_B.

8. The NF\_B sends the subscription response to the NF\_A after certain conditions are met.

9. Optionally, the NF\_B may send a notification to the NF\_A after certain conditions are met.

10. The NF\_B detects the monitored event occurs and sends the event report notification to the NF\_C, and the NF\_C decides whether to deal with the notification sent by NF\_B according to the saved token.

NOTE : Step 6 and 7 are optional and allow the consumer to verify whether producer is authorized to send the notification. They are only performed if the producer authorization is required.



Figure 6.15.2-2: Service access authorization in the delegated "Subscribe-Notify" interaction scenarios for roaming.

0. As precondition, the NF\_A and the NF\_C perform mutual authentication and determines that the NF\_A subscribes services on behalf of the NF\_C, and provides its own instance id and Notification URI to the peer.

1. The NF\_A sends the Nnrf\_AccessToken\_Get Request to the cNRF (the NRF in the PLMN of Consumer). The token request contains the instance ID of the NF\_A, the Notification URI of the NF\_A, the Notification URI of the NF\_C and an indication (Expressed as "Indication\_delegated") that indicates the token request is for the delegated "Subscribe-Notify" interaction scenarios which is the NF\_A subscribes services on behalf of the NF\_C.

2. The cNRF identify the pNRF (the NRF in the PLMN of Producer) based on the home PLMN ID, and request an access token from the pNRF.

3. Upon reception of the Nnrf\_AccessToken\_Get Request, the pNRF obtains the NF Instance ID of the Service Consumer (NF\_A), the Notification URI of the Service Consumer (NF\_A and NF\_C) and the indication\_delegated from the NF\_A. The pNRF then determines whether the NF\_A is authorized to subscribe services on behalf of the NF\_C, and whether the NF\_C is authorized to receive the NF service provided by the NF\_B, based on the locally configured policies or authorization information. If the authorization is successful, the pNRF generates a token which includes the instance ID of NF\_A, the Notification URI of the NF\_A, the Notification URI of the NF\_C, and the indication\_delegated. The token is used to indicate that the NF\_A is authorized to subscribe services on behalf of the NF\_C, and is used to indicate that the NF\_C is authorized to receive the NF service provided by the NF\_B.

4. The pNRF sends the Nnrf\_AccessToken\_Get Response to the cNRF with the token.

5. The cNRF forwards the Nnrf\_AccessToken\_Get Response to the NF\_A with the token.

6. The NF\_A sends a subscription request, which contains the token and the associated notification endpoint of the NF\_C and the NF\_A (i.e. the Notification URI of the NF\_C and the NF\_A), to the NF\_B.

7. Upon reception of the subscription request, the NF\_B verifies the integrity of the token and verify the claims in the token. If successful and determine that whether the NF\_C is authorized to receive the NF\_B service notification based on the instance ID and the Notification URI of the NF\_C contained in the token claims.

8. If the verification of the token is successful, the NF\_B sends an authorization notification to the NF\_C, which indicates the NF\_C is authorized to receive the NF service provided by the NF\_B. The authorization notification contain the token.

9. The NF\_C verify the integrity of the token also. If the token verification is successful, the NF\_C sends an authorization reply back to the NF\_B.

10. The NF\_B sends the subscription response to the NF\_A after certain conditions are met.

11. Optionally, the NF\_B may send a notification to the NF\_A after certain conditions are met.

12. The NF\_B detects the monitored event occurs and sends the event report notification to the NF\_C, and the NF\_C decides whether to deal with the notification sent by NF\_B according to the saved token.

NOTE : Step 8 and 9 are optional and allow the consumer to verify whether producer is authorized to send the notification. They are only performed if the producer authorization is required.

### 6.15.3 Evaluation

The solution extends the existing token-based authorization method by including the instance ID of the NF\_A and the instance ID of the NF\_C in the token claims. The token is used to indicate that the NF\_A is only authorized to subscribe services on behalf of the NF\_C (i.e. the NF\_A is NOT authorized to subscribe services on behalf of any other NFs except the NF\_C), and is also used to indicate that besides the NF\_A, the NF\_C is also authorized to receive the NF service provided by the NF\_B (i.e. any other unauthorized NFs except the NF\_A and the NF\_C is not be able to use the service of the NF\_B).

The solution fulfils the potential security requirements from Key Issue #28: "The 5G system shall support an authorization mechanism for the delegated "Subscribe-Notify" scenarios, in which NF\_A subscribes the service of NF\_B on behalf of NF\_C."

The solution assures that the NF\_B can detect that whether NF\_C is authorized to use the subscribe service and whether the NF\_B is authorized to do this delegated service request. Otherwise, NF\_B can subscribe services from NF\_C on behalf of any NFs without any authorization.

The solution has impact on the access token generation by adding the Notification URI of the NF\_A, the Notification URI of delegated NF\_C and the instance ID of delegated NF\_C into the access token, and token verification.

The NRF can verify the notification URI of the consumer by examining its NFprofile. Specifically, the NRF checks if the host information contained in the notification URI matches that (e.g. IP address or the FQDN) in the NFprofile..

## 6.16 Solution #16: OAuth 2.0 based authorization for Indirect communication without Delegated Discovery (Model C)

### 6.16.1 Introduction

The following figure, reproduced from clause 4.17.11 of TS 23.502, depicts the call flow for Model C.



Figure 6.16.1-1: Procedure for Indirect Communication without delegated discovery

When Indirect Communication Model without delegated discovery (Model C) is used for Network Functions to interact with each other, the Service Communication Proxy (SeCoP) is mainly used for message forwarding and routing between two Network Functions.

### 6.16.2 Solution details

The procedure for OAuth based service access authorization, defined in clause 13.4 of TS 33.501 [11], can be used for authorizing the NF consumer for service access in Model C scenarios. The OAuth 2.0 access token is obtained by the NF service Consumer and verified by the NF Service Producer. The SeCoP, which is forwarding and routing messages between the consumer and the producer, is not involved in OAuth-related procedures.

NOTE 1: It is not addressed in the present document whether the NFs can offload OAuth 2.0 functionality to Service Communication Proxy (SeCoP), including executing the required OAuth procedure to obtain the access token on the consumer side and verification of the access token on the producer side.

### 6.16.3 Evaluation

This solution addresses all requirements of key issue #22.

When Model C based indirect communication is used for communication between NFs, based service access authorization is used for authorizing NF consumer access to the services of the target NF producer.

The NF consumer is responsible for discovering and selecting the target NF producer. The NF consumer then uses OAuth access token request procedure to obtain an access token from the NRF. The NF producer uses access token verification procedure to verify and validate the access token.

## 6.17 Solution #17: Protection of SeCoP interfaces

### 6.17.1 Introduction

A new core network entity Service Communication Proxy (SeCoP) is used to enable indirect communications between Network Functions.



Figure 6.17.1-1: SeCoP Interfaces

As shown above, an SeCoP has interfaces with Network functions, the NRF and peer SeCoPs within the PLMN.

### 6.17.2 Solution details

The SeCoP is used in indirect communication for message forwarding and routing between various NFs. The SeCoP interfaces carry signalling data as well as privacy sensitive material and other parameters such as security keys. Therefore, confidentiality and integrity protection are required on these interfaces. Protection at the network or transport layer, as specified in Clause 13.1 in TS 33.501[11], applies to all SeCoP interfaces within a PLMN.

The SeCoP and the other endpoint of its interface, such as NFs including NRF and peer SeCoP, mutually authenticate each other before service layer messages can be exchanged on that interface.

- If the PLMN uses protection at the transport layer as described in clause 13.1 of TS 33.501[11], authentication provided by the transport layer protection solution is used for mutual authentication of the SeCoP and the other endpoint of its interface.

- If the PLMN does not use protection at the transport layer, mutual authentication of SeCoP and other endpoints may be implicit by NDS/IP or physical security (see clause 13.1 of TS 33.501[11]).

NOTE: In deployments where service-mesh technology is used and SeCoP endpoints are co-located with the 5GC functionality (e.g. an NF Instance), mutual authentication between SeCoP and the 5GC entity is implicit based on physical security. Explicit protection based on TLS or NDS/IP is optional in such deployments.

NOTE 1: It is not addressed in the present document whether the SeCoP may further perform authorization of a requesting NF before granting access to its services.

### 6.17.3 Evaluation

The above solution addresses all requirements of key issue #20.

## 6.18 Solution #18: Support NDS/IP on the inter-PLMN N9 interface

### 6.18.1 Introduction

For the protection of the non-SBA 5GC internal interfaces, such as N4 and N9, TS33.501[11] clause 9.9 already requires the use of NDS/IP. For confidentiality, integrity, and replay protection of inter-PLMN UP, such as N9, NDS/IP is used as specified in TS 33.210[20]. This can be simply achieved by minor modifications of TS 33.501[11] clause 9.9, as shown in the highlighted text below.

### 6.18.2 Solution details

The following clause is a direct copy from TS 33.501[11]; with text in bold indicating the proposed changes.

*"9.9 Security mechanisms for non-SBA interfaces internal to the 5GC* ***and between PLMNs***

*Interfaces internal to the 5G Core* ***such as N4 and N9 and roaming interfaces between PLMNs such as N9****, can be used to transport signalling data as well as privacy sensitive material, such as user and subscription data, or other parameters, such as security keys. Therefore, confidentiality and integrity protection are required.*

*For the protection of the non-SBA 5GC internal****and inter-PLMN*** *interfaces, NDS/IP* ***shall*** *be used as specified in [3], unless security is provided by other means, e.g. physical security."*

Creation of NDS/IP interfaces commensurate with requirements of network slices and transport of traffic from different slices on NDS/IP protected interfaces is up to operator policies.

### 6.18.3 Evaluation

The above solution addresses all requirements of key issue #26.

## 6.19 Solution #19: Service access authorization based on NF Set in non-roaming scenario

### 6.19.1 Introduction

This solution addresses key issue #24: Service access authorization based on NF Set in the non-roaming scenarios.

### 6.19.2 Solution details

#### 6.19.2.0 General

This authorization scheme is useful in the indirect communication mode, and the service producer within a NF Set. It is assumed that the NF\_A is the Service Consumer, and the NF\_B and the NF\_C are located in the same NF Set as the Service Producer. When the NF\_B and the NF\_C are registered to the NRF, it is assumed that the NF Set ID of the NF Set where the NF Producer is located is sent to the NRF as the NF profile.

NOTE 1: Not all the deployment scenarios are addressed in the present document.

#### 6.19.2.1 Service access authorization for NF producers within a NF set (Model B)

The following procedure is used when a NF service consumer needs an access token to obtain services from any of the NF service producers within the NF set.

As a pre-requisite to this procedure, the NF service consumer has obtained a list of NF producers within a set by executing the Discovery request procedure with the NRF.



Figure 6.19.2.1-1: OAuth based service access authorization for NF Sets

1. The NF service consumer sends the Nnrf\_AccessToken\_Get Request to the NRF. The Nnrf\_AccessToken\_Get Request contains the NF Instance ID of the Service Consumer (NF\_A), expected NF service name(s), and the NF Set Id of the producer.

2. The NRF authorizes the client.

3. The NRF generates an access token. It populates the "audience" claim in the access token with the NF Set Id.

4. The NRF sends the access token to NF service consumer.

5-6. The NF service consumer selects a NF producer within the NF Set, and issues a service request. It includes the access token obtained in step 4 in the service request.

6-7 The NF service consumer may reselect a new NF producer instance within the NF set for subsequent requests. It includes the same access obtained in step 4 above in the service request.

#### 6.19.2.2 Service access authorization based on NF Set by verifying the token on the service producer (Model C)



Figure 6.19.2.2-1: Service access authorization based on of NF Set.

1. The NF\_A sends the Nnrf\_AccessToken\_Get Request to the NRF. The Nnrf\_AccessToken\_Get Request contains the NF Instance ID of the Service Consumer (NF\_A), expected NF service name(s), NF Set ID of the Service Producer.

2. The NRF performs authorization. If the authorization is successful, the NRF generates a token. The Audience Claim in the token contains the NF Set ID of the Service Producer.

3. The NRF sends the token to the NF\_A through the Nnrf\_AccessToken\_Get Response.

4. The NF\_A sends a service request to the SeCoP. The service request contains the token.

5. The SeCoP selects a NF as Service Producer from the NF Set, such as selecting the NF\_C as the Service Producer.

6. The SeCoP sends the service request, which contains the token, to the NF\_C.

7. The NF\_C verifies the token integrity, and then verify whether the NF Set ID of the Producer in the Audience Claim is the same as the NF Set ID of the NF\_C.

8. If the token verification in the NF\_C is success, the NF\_C replies the service response to the SeCoP with requested service(s).

If the token verification in the NF\_C is fails, the NF\_C replies service response to the SeCoP with an error code indicating this mismatch.

9. The SeCoP sends the service response to the NF\_A.

### 6.19.3 Evaluation

The solution extends the existing token-based authorization method by including the NF Set ID of the NF producer in the token claims for Model B and Model C in the non-roaming scenario. The OAuth 2.0 based authorization method is applicable to the authorization based on NF Set. With this solution, the NF consumer is able to obtain services from a NF set using a token on NF Set granularity.

The solution fulfils the potential security requirements from Key Issue #24: "The 5GS shall support service access authorization based on NF Set".

## 6.20 Solution #20: UP Gateway function on the inter-PLMN N9 interface

### 6.20.1 Introduction

This solution provides a solution for key issue #27.

The SEPP-U is a gateway function used for filtering GTP-U traffic on the N9 interface. The SEPP-U filters GTP-U messages in a way that only genuine GTP-U packets, that correspond to active PDU sessions established through the N32 interface, can transit through the gateway. All other GTP-U packets are discarded and logged. This ensures that no unwanted GTP-U packets enter or leave the mobile network.

The SEPP-U function may be deployed either at the edge of the operator network or collocated with the UPF. It monitors incoming/outgoing GTP-U traffic on the N9 interface and executes GTP-U checks on every GTP-U packet on the N9 interface.

SEPP-U interacts with SMF over the Nx interface to obtain local and remote tunnelInfo information (TEID and tunnel IP address).

SEPP-U operates as a transparent gateway, which sits on the IP route, examines each packet and decides to either pass it or drop it.

In the following figure, SEPP-U is shown as a separate function in front of UPF to only forward GTP-U traffic, belonging to successfully established PDU sessions. The SEPP-U interfaces with the SMF over the Nx interface to obtain the required session (tunnelInfo) information.



Figure 6.20.1-1: SEPP-U, a UP gateway Function for the inter-PLMN N9 interface

### 6.20.2 Solution details

In the ingress direction (i.e. entering the network), the SEPP-U function intercepts all incoming GTP-U traffic on the N9 interface and forwards valid GTP-U traffic to the concerned UPF inside the network for further processing. This ensures that only valid GTP-U traffic is received at the UPF.

In the egress direction (i.e. exiting the network), the SEPP-U function intercepts all outgoing GTP-U traffic from UPFs, and forwards valid GTP-U traffic towards the other network.

#### 6.20.2.1 Interface between SEPP-U and Core Network control plane entity

A new interface is proposed between the SEPP-U and a Core Network control plane entity. This interface is used for communication between the core network control plane entity and SEPP-U.

The SMF, which has access to the TunnelInfo information of both GTP-U endpoints, is the Core Network control plane that interacts with the SEPP-U function.

The SEPP-U receives GTP Tunnel Info from SMF and executes the required operations.

The protocol between the SMF and the SEPP-U function may be based on the existing N4 interface and Packet Forwarding Control Protocol (PFCP) (see TS 29.244[18]).

In the following figure, SMF is the Control Plane entity that supplies SEPP-U with remote GTP\_U tunnel information including TEID and IP address:



Figure 6.20.2.1-1: Interface between SEPP-U and SMF

In Step 3, the SMF pushes the local TunnelInfo information of the GTP-U tunnel endpoint in its network and optionally the peer network TunnelInfo information of the peer GTP-U tunnel endpoint obtained from the other network to SEPP-U. This allows SEPP-U to identify and verify whether the incoming GTP-U traffic targets a valid GTP-U end-point in the network receiving the GTP-U packet and/or that it is from a valid network or not.

In addition, the SMF also indicates which operation to perform in SEPP-U for the TunnelInfo information. These operations may include add, modify or remove valid GTP-U session information in the SEPP-U, request to only check target destination IP address and TEID, or also check source IP address of the GTP-U packet.

In deployments where SEPP-U is collocated with the UPF, the existing N4 interface and Packet Forwarding Control Protocol (PFCP) (see 3GPP TS 29.244 [18]) between the SMF and the UPF may be used by the SMF to push GTP-U TunnelInfo to the UPF.

#### 6.20.2.2 Interface between UPFs and SEPP-U

In deployments where the SEPP-U function is centralized, for e.g. sitting at the perimeter configured to perform GTP-U firewall function on a traffic destined to a set of UPFs, SEPP-U operates as a transparent proxy.

The SEPP-U function looks for a specific pattern in the GTP-U packet (basically the GTP header and IP address in the IP Header) for validity checks. The UPFs may not be aware that SEPP-U exists at the perimeter of the network to monitor incoming GTP-U traffic.

### 6.20.3 Evaluation

The above solution addresses all requirements of key issue #27.

## 6.21 Solution #21: OAuth 2.0 based authorization for Indirect communication with Delegated Discovery (Model D)

### 6.21.1 Introduction

The Service Communication Proxy (SeCoP) implements OAuth 2.0 based Service access authorization of NF consumer when Option D architectural option is used to communicate between two Network Functions.

OAuth 2.0 roles, as defined in clause 1.1 of RFC 6749 [19], are as follows:

- Network Resource Function (NRF) is the OAuth 2.0 Authorization server

- Service Communication Proxy (SeCoP) on the consumer side is the OAuth 2.0 client

- Service Communication Proxy (SeCoP) on the producer side is the OAuth 2.0 resource server

Following figure illustrates the OAuth 2.0 architecture for indirect communications for deployments where the NFs and the SeCoPs are in different deployment units.



Figure 6.21.1-1: OAuth 2.0 Architecture for Indirect communication w/ Delegated discovery (Model D)

Rel-16 allows NRF to be co-located or combined with SCP. In such a scenario, the SCP could also include the functionality of OAuth 2.0 Authorization server.

Following figure illustrates the OAuth 2.0 architecture for indirect communications for deployments where the NFs and the SeCoPs are in co-located in the same deployment unit and SeCoP is setup as a service mesh proxy.



Figure 6.21.1-2: OAuth 2.0 Architecture for Indirect communication w/ Delegated discovery– service mesh deployment

NOTE: In deployments where a common SeCoP is used to connect NF Service Consumer and NF Service Producer, OAuth 2.0 based Authorization of NF access is not required.

### 6.21.2 Solution details

#### 6.21.2.1 SeCoP obtaining access token on behalf of the NF consumer

The following procedure describes how the SeCoP connected to the NF consumer obtains an access token before forwarding the service access to the selected NF service producer.



Figure 6.21.2.1-1: SeCoPc obtaining access token before service access

1. The NFc invokes the API requesting specific service towards the SeCoPc. The request may include discovery and selection parameters necessary to discover and select a NF service producer instance.

2. The SeCoP performs service discovery and selection of target NFps if required and selects the target NF service producer.

3. The SeCoP obtains an access token for service access authorization on behalf of the NF consumer. The message include the NF instance Id of the selected producer, expected NF service name(s), NF type of the expected NF producer instance and NF consumer.

4. The NRF checks its internal database to authorize the NF consumer. The NRF determines the scope of access based on what the NF consumer is authorized to access.

5. The NRF generates an access token with appropriate claims included. The NRF digitally sign the generated access token based on a shared secret or private key as described in RFC 7515 [45].

The claims in the token includes the NF Instance Id of NRF (issuer), NF Instance Id of the NF Service consumer (subject), NF Instance Id of the NF Service producer (audience), expected service name(s) (scope) and expiration time (expiration).

6. The NRF sends the access token along with the expiration time to the SeCoP.

The SeCoPc forwards the request to the selected NF service producer along with the access token. It also stores the access token till it receives service response from the NF service producer.

#### 6.21.2.2 SeCoP authorizing NF consumer based on token verification

The following procedure describes how authorization is performed by SeCoP before forwarding the service request to the NF service producer.



Figure 6.21.2.2-1: SeCoPp validating the access token and authorizing NFc

1. The SeCoP connected to the NF Producer receives the service request along with the access token.

2-3 The SeCoPp verifies the integrity of the access token, validates all the claims in the access token and authorizes NF consumer to obtain service from the NF producer.

4. The SeCoPp forwards the service request to the NF producer. The SeCoPp may optionally include the access token and an indication to NF producer.

#### 6.21.2.4 SeCoP includes access token in the Service Response message

When SeCoPc receives the Service Response message from the NF producer, it inserts in the message the access token that it has stored for this transaction and information on the discovered NF producer, and forwards the Service Response message to the NF consumer.

The NF consumer may use this token in scenarios where it directly talks to the NF producer.

### 6.21.3 Evaluation

The above solution addresses all requirements of key issue #22.

When Model D based indirect communication is used for communication between NFs, the SeCoP handles all OAuth related authorization procedures in addition to discovering and selecting the target NF producer.

On the consumer side, the SeCoP obtains an access token specific to the selected NF producer. On the producer side, the SeCoP verifies that the access token is applicable to the target NF producer before forwarding the service request to it.

The SeCoP implements HTTP transaction-level statefulness, that enables it to include the access token in the HTTP response message when forwarding the response from the NF producer, to the NF consumer. NF consumer may use the same access token for direct communication with the NF producer.

## 6.22 Solution #22: Authentication and authorization between Network Functions for Indirect Communication models

### 6.22.1 Introduction

NOTE 1: Authentication and Authorization of NFs in different deployment scenarios is not addressed in the present document.

This solution addresses key issue # 23.

When Indirect Communication via SeCoP is used by the NF Service consumer to communicate to the target NF Service producer, authentication between the NFs is achieved implicitly based on hop-by-hop authentication.

Authorization between NFs is either direct based on clause 13.3.2 in TS 33.501[11] or implicit by SeCoP authorizing NF Service consumer on behalf of the NF Service producer.

### 6.22.2 Solution details

In Indirect Communication, the NF Service consumer communicates with the target NF Service producer via a SeCoP. Authentication between NF Service consumer and NF Service producer is implicitly achieved as follows:

- When the communicating NFs are connected to a common SeCoP, authentication between NFs is implicit by authentication between NF and SeCoP.

- When the communicating NFs are connected to different SeCoPs within a PLMN, authentication between NFs is implicit by authentication between two SeCoPs, and between NF and SeCoP.

- When the communicating NFs are in different PLMNs, authentication between NFs is implicit by authentication between two SEPPs, SEPP-SeCoP and NF-SeCoP.

Depending on which model is used by the NF Service consumer for Indirect Communication, authorization is performed in one of the following ways:

- If Indirect communication without delegated discovery (Model C) is used, authorization is performed as specified in clause 13.3.2 of TS 33.501[11].

- If Indirect communication with delegated discovery (Model D) is used, authorization is delegated to SeCoP. Therefore, authorization is implicit and there is no direct authorization between NFs.

### 6.22.3 Evaluation

The above solution addresses all requirements of key issue #23.

## 6.23 Solution #23: Token-based authorization for Scenario D using stateless SeCoP

### 6.23.1 Introduction

This solution addresses Key Issue #22 (Authorization of NF service access in Indirect Communication). To be more specific, this solution addresses the scenario of indirect communication with delegated discovery (Scenario D).

One main idea of this solution is that the SeCoP should be stateless and not store any tokens. Instead it is the NF service consumer itself that stores the tokens. Statelessness in this context refers to the tokens, i.e. a stateless SeCoP is a SeCoP that does not store authorization tokens.

Another main idea of this solution is that existing methods in TS 33.501 [11] should be reused as much as possible. Indeed, this solution reuses only existing methods in TS 33.501[11] and does not need any additional normative specification.

### 6.23.2 Solution Description

#### 6.23.2.1 General

In the following, the abbreviations cNF and pNF for NF service consumer and producer, respectively, will be used.

#### 6.23.2.2 Assumptions on authentication and interface protection

A1. The SeCoP has authenticated the cNF (e.g. using TLS with server certificates) and the interface between them is confidentiality, integrity and replay protected (e.g. using TLS). The cNF may have authenticated the SeCoP.

A2. The SeCoP has authenticated the pNF (e.g. using TLS with server certificates) and the interface between them is confidentiality, integrity and replay protected (e.g. using TLS). The pNF may have authenticated the SeCoP.

The implications of the assumptions will be further discussed in clause 6.23.2.4 on the trust model below.

#### 6.23.2.3 Authorization and service invocation procedure



Figure 6.23.2.3-1: Authorization and service invocation procedure

**cNF authorization:**

1. The cNF sends an access token request to the NRF. The request contains the NF type of the pNF and potentially slice information, but not a specific NF instance ID of the pNF. The request also contains the NF instance ID of the cNF.

2. The NRF sends the access token response to the cNF including the signed token.

**Service request:**

3. The cNF uses delegated discovery and selection and sends a service request for the specific service to the SeCoP. The service request includes the access token for the pNF providing the service as received in step 2.

4. If no cached data is available the SeCoP discovers the pNF.

5. The SeCoP selects a pNF instance, performs the API root modifications and forwards the received request to the selected pNF instance. The request contains the token as received in 3. and valid for the cNF.

6. To authorize the access the pNF validates the token by verifying the signature and checking if the requested service is part of the token's scope. If the checks are ok the pNF processes the request and provides a response.

7. The SeCoP performs revers API root modifications and forwards the response.

#### 6.23.2.4 Trust model

According to the assumptions on authentication and interface protection, there is hop-by-hop server-side authentication and interface protection on the cNF – SeCoP – pNF link. Using only server side certificates is a possible minor optimization and mutual authentication may also be used. The SeCoP needs to be trusted by the cNF, NRF and pNF:

- Both the NRF and the cNF trust the SeCoP that it forwards tokens only to pNFs whose services the cNF requests.

- The pNF trusts the SeCoP that it only forwards service requests of the cNFs and does not impersonate cNFs using tokens received during the cNFs' token requests via the SeCoP.

Because all interfaces where tokens are sent are protected, there is no other entity except NRF, SeCoP, the authorized cNF and the pNF whose services are requested that receive the token. This also means that the pNF does not need to authenticate the SeCoP and the SeCoP does not need to authenticate the cNF, because the received token serves as authentication of the cNF requesting the service.

### 6.23.3 Solution Evaluation

This solution addresses Key Issue #22 Authorization of NF service access in Indirect Communication for Scenario D (indirect communication with delegated discovery). It describes how token-based authorization can be used in this scenario. No normative changes to TS 33.501 [11] are needed, and no tokens are stored in the SeCoP. This solution is based on direct authentication between consumer and NRF. It does not consider hierarchical NRFs.

## 6.24 Solution #24: Token-based authorization for Scenario C using stateless SeCoP

### 6.24.1 Introduction

This solution addresses Key Issue #22 Authorization of NF service access in Indirect Communication. To be more specific, this solution addresses the scenario of indirect communication without delegated discovery (Scenario C).

The main idea of this solution is that the SeCoP should be stateless and not store any tokens. Instead it is the NF service consumer itself that stores the tokens. Statelessness in this context refers to the tokens, i.e. a stateless SeCoP is a SeCoP that does not store authorization tokens.

Another main idea of this solution is that existing methods in TS 33.501 [11] should be reused as much as possible. Indeed, this solution reuses only existing methods in TS 33.501[11] and does not need any additional normative specification.

### 6.24.2 Solution Description

#### 6.24.2.1 General

In the following, the abbreviations cNF and pNF for NF service consumer and producer, respectively, will be used.

#### 6.24.2.2 Assumptions on authentication and interface protection

A1. The SeCoP has authenticated the cNF (e.g. using TLS with server certificates) and the interface between them is confidentiality, integrity and replay protected (e.g. using TLS). The cNF may have authenticated the SeCoP. A2. The SeCoP has authenticated the pNF (e.g. using TLS with server certificates) and the interface between them is confidentiality, integrity and replay protected (e.g. using TLS). The pNF may have authenticated the SeCoP.

The implications of the assumptions will be further discussed in clause 6.24.2.4 on the trust model below.

#### 6.24.2.3 Authorization and service invocation procedure



Figure 6.24.2.3-1: Authorization and service invocation procedure

**Discovery of the NF service producer:**

0. Optionally, the NF service consumer may discover the NF service producer before requesting authorization to invoke the services of the NF service producer.

**NF service consumer authorization:**

1. The cNF sends an access token request to the NRF. The request also contains the NF instance ID of the cNF.

2. The NRF sends the access token response to the cNF including the signed token.

**Service request:**

3. If no cached data is available, the NF service consumer discovers the NF service producer via the SeCoP.

4. The cNF sends a service request for the specific service to the SeCoP. The service request includes the access token for the pNF providing the service as received in step 5.

5. The SeCoP selects a pNF instance, performs the API root modifications and forwards the received request to the selected pNF instance. The request contains the token as received in 6. and valid for the cNF.

6. To authorize the access the pNF validates the token by verifying the signature and checking if the requested service is part of the token's scope. If the checks are ok the pNF processes the request and provides a response.

7. The SeCoP performs revers API root modifications and forwards the response.

#### 6.24.2.4 Trust model

According to the assumptions on authentication and interface protection, there is hop-by-hop server-side authentication and interface protection on the cNF – SeCoP – pNF link. Using only server side certificates is a possible minor optimization. Client certificates may also be used. The SeCoP needs to be trusted by the cNF, NRF and pNF:

- Both the NRF and the cNF trust the SeCoP that it forwards tokens only to pNFs whose services the cNF requests.

- The pNF trusts the SeCoP that it only forwards service requests of the cNFs and does not impersonate cNFs using tokens received during the cNFs' token requests.

Because all interfaces where tokens are sent are protected, there is no other entity except NRF, SeCoP, the authorized cNF and the pNF whose services are requested that receive the token. This also means that the pNF does not need to authenticate the SeCoP and the SeCoP does not need to authenticate the cNF, because the received token serves as authentication of the cNF requesting the service.

### 6.24.3 Solution Evaluation

This solution addresses Key Issue #22 Authorization of NF service access in Indirect Communication for Scenario C (indirect communication without delegated discovery). It describes how token-based authorization can be used in this scenario. No normative changes to TS 33.501 [11] are needed, and no tokens are stored in the SeCoP. This solution is based on direct authentication between consumer and NRF. It does not consider hierarchical NRFs.

## 6.25 Solution #25: NF service consumer verification during service access authorization in the direct communication scenario

### 6.25.1 Introduction

This solution addresses key issue # 5, and proposes a new mechanism allowing the NF service producer to verify the identity of the NF service consumer securely based on the certificate information in the direct communication scenario.

### 6.25.2 Solution details for the non-roaming scenario

#### 6.25.2.0 General

There are two parts in this scenario:

a) Including NF service consumer's certificate in the access token generated by the NRF.

b) NF service producer verifies whether the certificate in the access token is same as the certificate received during mutual authentication procedure with NF service consumer.

#### 6.25.2.1 Access token generation with the certificate of the NF service consumer



Figure 6.25.2.1-1: Including NFc's certificate in the access token – in non-roaming scenarios

In this procedure, the NRF includes NF service consumer's certificate (denoted by NFc's certificate) as one of the claims in the access token. The above figure illustrates the procedure in the non-roaming scenarios.

Compared with the procedure specified in clause 13.4.1.1 of TS 33.501[11], the enhancement for the access token generation procedure includes the following two parts.

- In step 1. After the mutual authentication, NRF stores the NFc's certificate, which is received after the successful mutual authentication, and used by the NRF to verify the identity of the NF service consumer.

- In step 3. After receiving the Nnrf\_AccessToken\_Get request, NRF generates the access token, in which the hashedNFc's certificate is included as a new parameter.

#### 6.25.2.2 NF service Producer authenticates NF consumer



Figure 6.25.2.2-1: NF service producer authenticates NF consumer

Compared with the procedure specified in clause 13.4.1.1 of TS 33.501[11], the enhancement for the access token verification procedure includes the following two parts.

- In step 1. After the mutual authentication, NF service producer stores the NFc's certificate, which is received during mutual authentication, and used by the NF service producer to verify the identity of the NF service consumer.

- In step 3. After receiving the NF service request, besides the current token verification, NF service producer verifies whether the hashed NFc's certificate stored in step 1, is the same as the hashed certificate included in the access token. If the certificate verification is failure, the NF service producer replies the service response to the NF service consumer with an error code indicating this mismatch

### 6.25.3 Solution details for the roaming scenario

For the roaming scenario, there is no end to end mutual authentication between the NF service consumer and the hNRF. One option would that the hashed NFc's certificate can be embedded into the access token request sent by NF service consumer to the hNRF via vNRF. Then the hNRF generates the access token, by including the hashed NFc's certificate, then sends the access token to the NF service consumer via vNRF



Figure 6.25.3-1: cSEPP authenticates NF consumer

Compared with the procedure specified in clause 13.4.1.1 of TS 33.501[11], the enhancement for the access token verification procedure includes the following parts.

- In step 1. After the mutual authentication, cSEPP stores the NFc's certificate, which is received during mutual authentication.

- In step 3. After receiving the NF service request, cSEPP verifies whether the hashed NFc's certificate stored in step 1, is the same as the hashed certificate included in the access token. If the certificate verification is failure, the cSEPP replies the service response to the NF service consumer with an error code indicating this mismatch. If the verification successes, the cSEPP sends the access token to the pSEPP.

- In step 6. NF service producer verifies the claims of access token except the hashed NFc's certificates.

### 6.25.4 Evaluation

The above solution addresses the requirements of key issue #5 in the direct communication mode.

For this requirement, the root CA inside the PLMN may be used for the NF service consumer verification. For instance, after receiving the registration request message including the instance ID of NF service consumer, the NRF as a root CA generates a signature based on the instance ID of NF service consumer and NFc's certificate, which is regarded as a proof for the instance ID and certificate combination. Then in the next service request procedure, the NF service producer can verify the instance ID of NF service consumer based on the signature.

Technically, this option is overlap with access token mechanism, since the access token is also generated based on the private key of NRF. On the other hand, there is no need to introduce a new root CA for this signature besides the root CA for access token. Hence, introducing a new claim for the hashed NFc's certificate in this solution would be better.

This solution allows the NF service producer to verify whether the NF service consumer requesting the service is the same one who is authorized by the NRF, by adding the NF service consumer certificate information into the access token during the access token generation procedure. Since the certificate of the NF service consumer is both authenticated by the NRF and NF service producer, the instance ID of NF service consumer can be validated based on the combination of certificate and instance ID in the access token. This solution assures that the access token can only be used by the NF service consumer who retrieved this access token using its certificate.

The solution has impact on the access token generation by adding the certificate information of NF service consumer, and token verification.

## 6.26 Solution #26: OAuth 2.0 based resource level authorization of NF service consumers

### 6.26.1 Introduction

This solution addresses KI #29: Resource level authorization of NF consumers.

### 6.26.2 Solution Description

The basic idea is to enhance OAuth 2.0 procedures to convey additional information in the JSON Web Token, that enables the NF producer or SeCoP (in Indirect communications) to verify whether the requesting NF is authorized to access resources/datasets managed by the NF producer.



Figure 6.26.2-1: Access token request for accessing resources within the NF

1. The NF service consumer requests an access token from the NRF. The request includes the NF Instance Id(s) of the NF service consumer, expected NF service name(s), **expected NF resource(s),** NF type of the expected NF producer instance and NF consumer.

NOTE 1: Expected resource name is also included when the access token request is for a particular NF producer instance.

2. The NRF authorizes the NF service consumer. It then generates an access token with appropriate claims included.

3. The claims in the token includes the NF Instance Id of NRF (issuer), NF Instance Id of the NF Service consumer (subject), NF type of the NF Service producer (audience), expected service name(s) (scope), allowed resources (allowedResources) and expiration time (expiration).

NOTE 2: The claim "allowedResources" contains the resource URI(s) of the data set(s)/resource(s) the consumer is allowed to access. If this claim is absent, it means that the NF service consumer is free to access all the resources within the NF.

4. The NRF sends the access token with the allowed resources to the NF service consumer.

The NF producer/SCP checks if the "allowedResources" claim is present, and accordingly grants access to only those resources present in this claim.

### 6.26.3 Solution Evaluation

This solution meets all the requirements of KI #29.

The access token is enhanced to include a new claim that restricts the scope of access to one or more resources within the NF service. The target NF producer instance will use this claim to regulate access to sensitive information within the corresponding NF Service.

## 6.27 Solution #27: Policy based authorization for Indirect communication between Network functions

### 6.27.1 Introduction

This solution addresses KI #22 - Authorization of NF service access in Indirect Communication.

The solution proposes policy-based authorization of NF consumer requests in the SeCoP associated with the NF producer.

NOTE 1: It is not addressed in the present document whether this solution can be standardized.

A set of policies are provisioned in the SeCoP which allow the SeCoP to recognise an incoming Service Request from a NF consumer and determine whether to allow the request and set of services that can be allowed for the requesting NF.

NOTE 2: The solution is based on static set of policy files that are provisioned by the operator using OAM infrastructure.

The following figure gives a pictorial description of the proposal:



Figure 6.27.1-1: Policy based service access authorization of NF consumer

1) The NF consumer (NFc) of a certain Network function type (NF type) is invoking an API request on the selected target NF producer (NFp). The message is routed via an SeCoPc.

2) The SeCoPc routes the message to a peer SeCoP (SeCoPp) that is proxying on behalf of the NF producer.

3) The SeCoP associated with the producer (SeCoPp) checks if the NF type to which the NF consumer belongs, is authorized to obtain services from the target NF producer (NFp).

4) If the NFc is authorized, the SeCoPp forwards the API request to NFp.

5) NFp provides service to the NFc via the SeCoP.

### 6.27.2 Solution Description

#### 6.27.2.1 Policy files

Two sets of policy files are required.

a) **Permissions** – defines how resources within an NF service can be accessed. Essentially this refers to the resources managed by a service and set of operations that can be performed on them (such as Create, Delete etc).

b) **Permissions to NF type binding**– binds NF consumers to Permissions. This policy file maps NF type (attribute) to set of permissions. Essentially this policy file dictates for each NF type:

- which resource can a NF of a that NF type access within the NF service, and

- applicable operations that can be performed.

The combination of **Permissions** and **Permissions to NF type Binding** specifies: **which** NF consumer is allowed to do **what** in an NF producer.

#### 6.27.2.2 Procedure

The following figure illustrates the concept in Model C of Indirect communications.



Figure 6.27.2.2-1: Policy based service access authorization (e.g. in model C)

1. The NF Service Consumer sends a POST request to a particular resource identified by its resource URI.

NOTE 1: In model D, the AMF sends POST request without selecting the NF instance.

2. The SeCoP connected to the NFc (SeCoPc) routes the message to the SeCoP proxying the NFp (SeCoPp).

NOTE 2: In model D, the SeCoPc first discovers appropriate NF instances that can service the request from consumer and selects one NF instance. It then routes the message SeCoPp proxying for the selected NF instance.

3. The SeCoPp refers to the **Permissions to NF type binding** policy file to check if the NFc is authorized to perform POST on the resource.

4-5. If service request from NFc is allowed, SeCoPp will forward the POST method to NFp.

6. Service between the NFc and NFp takes place.

### 6.27.3 Solution Evaluation

Void

## 6.28 Solution #28: Authorization between Network Functions in Scenario D

### 6.28.1 Introduction

This solution addresses key issue #23 NF to NF authentication and authorization in Indirect communication, in Scenario D (indirect communication with delegated discovery).

### 6.28.2 Solution details

If indirect communication with delegated discovery (Model D) is used, authorization is performed as specified in clause 13.3.2 of TS 33.501[11].

### 6.28.3 Evaluation

The above solution addresses the requirements of key issue #23.

## 6.29 Solution #29: Telescopic FQDN for the SeCoP

### 6.29.1 Introduction

This solution addresses Key Issue #20: Protection of SeCoP interfaces and Key Issue #21: Secure message transport via the SeCoP.

The solution is based on Solution #17: Protection of SeCoP interfaces and Solution #22: Authentication and authorization between Network Functions for Indirect Communication models but adds details on the TLS and routing issues described in Key Issue #18: Inter-PLMN routing and TLS.

In Scenario C (Indirect communication without delegated discovery), the SeCoP needs to terminate TLS in order to perform its tasks. The FQDN in the Request URIs contain the FQDN of the service consumer. In order to terminate TLS, the SeCoP needs to provide a certificate on behalf of remote consumer.

### 6.29.2 Solution Description

This solution proposes to reuse the "telescopic FQDN" solution that is described for the SEPP in clause 13.1 of TS 33.501 [11].

NOTE 1: Relations to TS 29.500 on how to handle routing are not addressed in the present document.

NOTE 2: Further details of the solution are not addressed in the present document.

### 6.29.3 Solution Evaluation

Void

## 6.30 Solution #30: Token-based authorization for NF Sets / NF Service Sets by existing methods

### 6.30.1 Introduction

This solution addresses Key Issue #24: Service access authorization within a NF Set or a NF Service Set.

### 6.30.2 Solution Description

Token-based authorization as described in TS 33.501 [11] already provides means to authorize service access with high granularity. Specifically, authorization according to slice information is already possible.

It may be possible that there will be no difference from an authorization point of view between NF Sets / NF Service Sets of producers of the same type and sub-type belonging to the same slice. In that case, the existing methods in TS 33.501 [11] are already sufficient.

### 6.30.3 Solution Evaluation

The solution addresses Key Issue #24 without any additional standardization effort.

## 6.31 Solution #31: Service access authorization based on of a NF Set in roaming scenario

### 6.31.1 Introduction

This solution addresses key issue #24: Service access authorization based on NF Set in the roaming scenarios.

### 6.31.2 Solution details

#### 6.31.2.0 General

This authorization scheme is useful in the indirect communication mode, and the service producer within a NF Set. It is assumed that the NF\_A is the Service Consumer, and the NF\_B and the NF\_C are located in the same NF Set as the Service Producer. When the NF\_B and the NF\_C are registered to the NRF, it is assumed that the NF Set ID of the NF Set where the NF Producer is located is sent to the NRF as the NF profile.

#### 6.31.2.1 Service access authorization for NF producers within a NF set (Model C)



Figure 6.31.2.1-1: Service access authorization based on NF Set in roaming scenarios

1. The NF\_A sends the Nnrf\_AccessToken\_Get Request to the vNRF. The Nnrf\_AccessToken\_Get Request contains the NF Instance ID of the Service Consumer (NF\_A and NF\_B), expected NF service name(s), and the NF Instance ID of the Service Producer (NF\_B), NF Set ID of the Service Producer.

2. The vNRF sends the Nnrf\_AccessToken\_Get Request to the hNRF.

3. The hNRF performs authorization. If the authorization is successful, the NRF generates a token. The Audience Claim in the token contains the NF Set ID of the Service Producer.

4-5. The hNRF sends the token to the NF\_A through the vNRF in Nnrf\_AccessToken\_Get Response.

6-7. The NF\_A sends a service request to the hSeCoP through the vSeCoP. The service request contains the token.

8. The hSeCoP selects a NF as Service Producer from the NF Set, such as selecting the NF\_C as the Service Producer.

9. The hSeCoP sends the service request, which contains the token, to the NF\_C.

10. The NF\_C verifies the token integrity, and then verify whether the NF Set ID of the Producer in the Audience Claim is the same as the NF Set ID of the NF\_C.

11. If the token verification in the NF\_C is success, the NF\_C replies the service response to the hSeCoP with requested service(s).

If the token verification in the NF\_C is fails, the NF\_C replies service response to the hSeCoP with an error code indicating this mismatch.

12-13. The hSeCoP send the service response to the NF\_A through the vSeCoP.

### 6.31.3 Solution Evaluation

The solution extends the existing token-based authorization method by including the NF Set ID of the NF producer in the token claims for Model C in the roaming scenario. The OAuth 2.0 based authorization method is applicable to the service authorization based on NF Set. With this solution, the NF consumer is able to obtain services from a NF set using a token on NF set granularity.

The solution fulfils the potential security requirements from Key Issue #24: "The 5GS shall support service access authorization based on NF Set".

## 6.32 Solution #32: OAuth 2.0 based resource level authorization of NF service consumers

### 6.32.1 Introduction

This solution addresses KI #29: Resource level authorization of NF consumers.

### 6.32.2 Solution Description

#### 6.32.2.0 General

The basic principle of this solution is to enhance OAuth 2.0 procedures to convey "additional scope" information (within the existing "scope" claim) in the JSON Web Token, that enables the NF producer to verify whether the requesting NF is authorized to perform the requested operation (read/write) over the resources/datasets managed by the NF producer.

The procedure requires the NF service producer to register in NRF the allowed "additional scope" information for each type/sub-type of NF consumer.

NOTE: Alternatively, the "additional scope" information for each type/sub-type of NF consumer may be also locally configured in the NRF. In that case, the registration of the NF service producer in the NRF as OAuth 2.0 resource server can be omitted.

If needed, it also requires the service consumer to register the NF sub-type so that authorization can be based on both NF type and NF sub-type (e.g. UDM vs UDM-ARPF NF sub-type, IMS-AS NF type with Multimedia-Telephony NF sub-type).

The following clauses define the phases and steps of the proposed procedure using the description and information flows in clause 13.4 of TS 33.501 [11]. The additional steps proposed by this solution are in bold.

#### 6.32.2.1 NF OAuth 2.0 client (NF service consumer) registration with the OAuth 2.0 authorization server (NRF)

The NF service registration procedure, as defined in clause 4.17.1 of TS 23.502 [8], is used to register the OAuth 2.0 client (NF service consumer) with the OAuth 2.0 Authorization server (NRF), as described in clause 2.0 of RFC 6749 [43]. The client id, used during OAuth 2.0 registration, is the NF Instance Id of the NF. I**n addition to the NF type, the NF service consumer may include the NF sub-type (which is related to the scope to be later requested by the client and accepted by the NRF) as part of its NF profile.**



Figure 6.32.2.1-1: NF service consumer registers in NRF

1) The NF service consumer registers as OAuth 2.0 client in the NRF. "NF sub-type" parameter is included as part of NF profile configuration data by the NF service consumer. This parameter is associated to the "additional scope" information allowed to be requested and granted to the client.

2) After storing the NF Profile, NRF responds successfully.

NOTE 1: The "NF sub-type" can be associated with "additional scope" information in the NF service producer (resource server).

NOTE 2: Alignment of NF subtype during the registration procedure with TS 23.502 and TS 29.510 is not addressed in the present document.

NOTE 3: Impact if the additional scope for the NF producer is sent to the NRF during registration is not addressed in the present document.

#### 6.32.2.2 NF OAuth 2.0 resource server (NF service producer) registration with the OAuth 2.0 authorization server (NRF)

The solution defined in this clause also proposes that the NF service registration procedure, as defined in clause 4.17.1 of TS 23.502 [8], is also used to register the OAuth 2.0 resource server (NF service producer) with the OAuth 2.0 Authorization server (NRF), as described in clause 2.0 of RFC 6749 [43]. The NF service producer, as part of its NF profile, includes "additional scope" information related to the allowed service operations and resources per NF consumer type and optionally NF consumer sub-type.



Figure 6.32.2.2-1: NF service producer registers in NRF

1) The NF service producer registers as OAuth 2.0 resource server in the NRF. The "additional scope" information is included as part of NF profile configuration data by the NF service producer. This information indicates the additional scope(s) allowed to be requested and granted per NF type and optionally per NF sub-type.

2) After storing the NF Profile, NRF responds successfully.

NOTE: The "additional scope" information can be associated to NF type only (e.g. UDM, IMS-AS) or to NF type and NF sub-type (e.g. UDM-ARPF, IMS-AS-MMtel-Service).

Alternatively, the "additional scope" information for some services and NF consumer type/sub-type can be locally configured at the NRF as a default extension of the NF profiles registered by some NF service producers. For example, NRF can be configured with an additional scope for Nudm\_SDM\_Get requests for am-data from AMFs. In this case, it is required that all NF service producer instances registering in the NRF which expose services for which "additional scope" info is locally configured at the NRF, are capable to authorize the corresponding service requests based on access tokens that include additional scope(s).

#### 6.32.2.3 NF Access token request before service access

The following procedure describes how the NF service consumer obtains an access token before service access to NF service producers of a specific NF type.

Pre-requisite:

a) The NF Service consumer (OAuth2.0 client) is registered with the NRF (Authorization Server) with its NF type and optionally NF sub-type (if configured in the NF profile)

b) The NF Service producer (OAuth2.0 resource server) is registered with the NRF (Authorization Server) with "additional scope" information per NF type and optionally NF sub-type (if configured in the NF profile)

NOTE 1: Alternatively, the "additional scope" information for each type/sub-type of NF consumer could be also locally configured in the NRF. In that case, the registration of the NF service producer in the NRF as OAuth 2.0 resource server can be omitted.

c. The NRF and NF service producer share the required credentials.

d. The NRF and NF service consumer have mutually authenticated each other.



Figure 6.32.2.3-1: NF service consumer obtaining access token before NF service access

1. The NF service consumer requests an access token from the NRF in the same PLMN using the Nnrf\_AccessToken\_Get request operation. The message includes the NF Instance Id(s) of the NF service consumer, the requested "scope" including the expected NF service name(s), NF type of the expected NF producer instance and NF consumer. **Additionally, the NF service consumer may also include in the authorization token request "additional scope" information requested to be authorized within the "scope" parameter.**

2. The NRF may optionally authorize the NF service consumer. It then generates an access token with appropriate claims included. **The NRF generates the access token based on the information included in the authorization token request (i.e. "scope" including expected service names and "additional scope" information to be authorized, if any), the information registered in the NRF by the NF service consumer (e.g. "NF sub-type", if any) and the "additional scope" information per NF type and NF sub-type registered by the NF service producer or locally configured in NRF, if any)**.

The NRF digitally signs the generated access token based on a shared secret or private key as described in RFC 7515 [45].

The claims in the token include the NF Instance Id of NRF (issuer), NF Instance Id of the NF Service consumer (subject), NF type of the NF Service producer (audience), the expected service name(s) **and optional "additional scope" information (scope**) and expiration time (expiration).

NOTE 2: The "additional scope" information is part of the "scope" parameter using a space-delimited strings as described in IETF RFC 6749 [5], clause 3.3.

3. If the authorization is successful, the NRF sends access token to the NF service consumer in the Nnrf\_AccessToken\_Get response operation, otherwise it replies based on Oauth 2.0 error response defined in RFC 6749 [43]. The other parameters (e.g., the expiration time) sent by NRF in addition to the access token are described in TS 29.510 [68].

The NF service consumer may store the received token(s). Stored tokens may be re-used for accessing service(s) from producer NF type listed in claims (scope, audience) during their validity time.

#### 6.32.2.4 NF Service access request based on token verification

The following figure and procedure describe how authorization is performed during Service request of the NF service consumer. **Prior to the request, the NF service consumer may perform Nnrf\_NFDiscovery\_Request operation with the requested additional scopes to select a suitable NF service producer (authorization server) which is able to authorize the Service Access request.**



Figure 6.32.2.3-1: NF service consumer requesting service access with an access token

Pre-requisite: The NF service consumer is in possession of a valid access token before requesting service access from the NF Service producer.

1. The NF Service consumer requests a service from the NF service producer. The NF Service Consumer includes the access token.

The NF Service consumer and NF service producer authenticates each other following clause 13.3.

2. The NF Service producer verify the token as follows:

- The NF Service producer ensures the integrity of the token by verifying the signature using NRF's public key or checking the MAC value using the shared secret. If integrity check is successful, the NF Service producer verifies the claims in the token as follows:

- It checks that the audience claim in the access token matches its own identity or the type of NF service producer.

**- If scope is present, it checks that the scope matches the requested service operation.**

**- If scope contains "additional scope" information, it checks that the additional scope matches the requested service operation.**

**NOTE: The "additional scope" information can identify an operation over a resource or a set of operations over a set of resources. The set of operations over a set of resources associated to the "additional scope" information is according to a locally configured list on the service producer. The "additional scope" information can identify an authorization profile in the NF service producer which defines the set of allowed operations over different allowed resources and/or datasets. This enables to extend the list of operations and resources allowed without enlarging the "scope" parameter every time a new resource is added to the service.**

**The additional scope(s) included within the access token add additional security checks at the NF service producer that authorizes the services operations, resources and NF consumer type/sub-types related to the additional scope(s).**

- It checks that the access token has not expired by verifying the expiration time in the access token against the current data/time.

3. If the verification is successful, the NF Service producer executes the requested service and responds back to the NF Service consumer. Otherwise it replies based on Oauth 2.0 error response defined in RFC 6749 [43]. The NF service consumer may store the received token(s). Stored tokens may be re-used for accessing service(s) from producer NF type listed in claims (scope, audience) during their validity time.

### 6.32.3 Solution Evaluation

Void

## 6.33 Solution #33: NF service consumer verification during service access authorization in indirect communication scenario

### 6.33.1 Introduction

This solution addresses key issue # 22, and proposes a new mechanism allowing the SeCoP to verify the identity of the NF service consumer securely based on the certificate information in the indirect communication scenario.

### 6.33.2 Solution Description

#### 6.33.2.1 General

This solution addresses the NF service consumer verification indirect communication scenarios, including the scenario C and scenario D, in which the SeCoP as the proxy verifies the instance ID of NF service consumer during the service access procedure.

#### 6.33.2.2 Solution details for the scenario C

##### 6.33.2.2.0 General

There are two parts in this scenario:

a) Including NF service consumer's certificate in the access token generated by the NRF.

b) SeCoP verifies whether the certificate in the access token is same as the certificate received during mutual authentication procedure with NF service consumer.

##### 6.33.2.2.1 Access token generation with the certificate of the NF service consumer



Figure 6.33.2.2.1-1: Including NFc's certificate in the access token – in non-roaming scenarios

In this procedure, the NRF includes NF service consumer's certificate (denoted by NFc's certificate) as one of the claims in the access token. The above figure illustrates the procedure in the non-roaming scenarios.

Compared with the procedure specified in clause 13.4.1.1 of TS 33.501[11], the enhancement for the access token generation procedure includes the following two parts.

- In step 1. After the mutual authentication, NRF stores the NFc's certificate, which is received after the successful mutual authentication, and used by the NRF to verify the identity of the NF service consumer.

- In step 3. After receiving the Nnrf\_AccessToken\_Get request, NRF generates the access token, in which the hashedNFc's certificate is included as a new parameter.

##### 6.33.2.2.2 SeCoP authenticates NF consumer



Figure 6.33.2.2.2-1: NF service producer authenticates NF consumer

Compared with the procedure specified in clause 13.4.1.1 of TS 33.501[11], the enhancement for the access token verification procedure includes the following two parts.

- In step 1. After the mutual authentication, SeCoP stores the NFc's certificate, which is received during mutual authentication.

- In step 3. After receiving the NF service request, SeCoP verifies whether the hashed NFc's certificate stored in step 1, is the same as the hashed certificate included in the access token. If the certificate verification is failure, the SeCoP replies the service response to the NF service consumer with an error code indicating this mismatch. If the verification successes, the SeCoP sends the access token to the NF service producer.

- In step 5. NF service producer verifies the claims of access token except the hashed NFc's certificates.

#### 6.33.2.3 Solution details for the scenario D

##### 6.33.2.3.0 General

There are two parts in this scenario:

a) Including NF service consumer's certificate sent by SeCoP in the access token generated by the NRF.

b) SeCoP verifies whether the certificate in the access token is same as the certificate received during mutual authentication procedure with NF service consumer.

##### 6.33.2.3.1 Access token generation with the certificate of the NF service consumer



Figure 6.33.2.3.1-1: Including NFc's certificate in the access token – in non-roaming scenarios

In this procedure, the NRF includes NF service consumer's certificate (denoted by NFc's certificate) as one of the claims in the access token. The above figure illustrates the procedure in the non-roaming scenarios.

Compared with the procedure specified in clause 13.4.1.1 of TS 33.501[11], the enhancement for the access token generation procedure includes the following two parts.

- In step 1. After the mutual authentication, SeCoP stores the NFc's certificate, which is received after the successful mutual authentication.

- In step3. SeCoP sends the hashed NFc's certificate to NRF, and also forwards the received Nnrf\_AccessToken\_Get Request to the NRF.

- In step 4. NRF generates the access token, in which the hashed NFc's certificate is included as a new parameter.

##### 6.33.2.3.2 SeCoP authenticates NF consumer



Figure 6.33.2.3.2-1: NF service producer authenticates NF consumer

Compared with the procedure specified in clause 13.4.1.1 of TS 33.501[11], the enhancement for the access token verification procedure includes the following two parts.

- In step 1. After the mutual authentication, SeCoP stores the NFc's certificate, which is received during mutual authentication.

- In step 3. After receiving the NF service request, SeCoP verifies whether the hashed NFc's certificate stored in step 1, is the same as the hashed certificate included in the access token. If the certificate verification is failure, the SeCoP replies the service response to the NF service consumer with an error code indicating this mismatch. If the verification successes, the SeCoP sends the access token to the NF service producer.

- In step 5. NF service producer verifies the claims of access token except the hashed NFc's certificates.

#### 6.33.2.4 Solution details for the roaming scenario

For the roaming scenario, there is no end to end mutual authentication between the NF service consumer and the hNRF. One option would that the hashed NFc's certificate can be embedded into the access token request sent by NF service consumer to the hNRF via vNRF and SeCoP. Then the hNRF generates the access token, by including the hashed NFc's certificate, then sends the access token to the NF service consumer via vNRF.



Figure 6.25.3-1: SeCoP authenticates NF consumer

Compared with the procedure specified in clause 13.4.1.1 of TS 33.501[11], the enhancement for the access token verification procedure includes the following parts.

- In step 1. After the mutual authentication, SeCoP stores the NFc's certificate, which is received during mutual authentication.

- In step 3. After receiving the NF service request, SeCoP verifies whether the hashed NFc's certificate stored in step 1, is the same as the hashed certificate included in the access token. If the certificate verification is failure, the SeCoP replies the service response to the NF service consumer with an error code indicating this mismatch. If the verification successes, the SeCoP sends the access token to the cSEPP.

- In step 6. NF service producer verifies the claims of access token except the hashed NFc's certificates.

### 6.33.3 Evaluation

NOTE 1: Addressing the scenarios where the solution does not work is not addressed in the present document.

## 6.34 Solution #34: Security of indirect communication in roaming scenarios

### 6.34.1 Introduction

This solution addresses Key Issue #25 "Indirect communication in roaming scenarios".

When indirect communication is used in roaming scenarios, the interface between SECOP and SEPP needs to be secured. More specifically, authentication and transport security between SECOP and SEPP is necessary.

Authentication between network functions in different PLMNs is implicit by authentication between NF-SECOP, SECOP-SEPP, SEPP-SEPP and SEPP-SECOP and SECOP-NF.

In should be noted that in a roaming scenario it is assumed that a SECOP does not span multiple PLMNs and that the interface between a SECOP and a SEPP is within a single PLMN.

### 6.34.2 Solution Description

Authentication between SECOP and SEPP is performed in the same way as authentication between SEPP and network functions, as described in TS 33.501 [11], clause 13.3.3.

Transport protection between SECOP and SEPP is performed in the same way as transport protection between SEPP and network functions, as described in TS 33.501 [11], clause 13.1.

Details of communication between SECOP and SEPP will be handled during normative work.

### 6.34.3 Solution Evaluation

The solution satisfies the potential requirements of Key Issue #25.

## 6.35 Solution #35: Service access authorization in the non-delegated "Subscribe-Notify" interaction scenarios

### 6.35.1 Introduction

This solution addresses key issue #30: Service access authorization for non-delegated subscribe-notify.

### 6.35.2 Solution details

This authorization scheme is useful in the non-delegated "Subscribe-Notify" interaction scenarios, which is a NF\_A subscribes to NF Service offered by NF\_B for itself.



Figure 6.35.2-1: Service access authorization in the non-delegated "Subscribe-Notify" interaction scenarios for non-roaming.

1. The NF\_A sends the Nnrf\_AccessToken\_Get Request to the NRF. The token request contains the instance ID of the NF\_A, the Notification URI of the NF\_A, an indication (Expressed as "Indication\_non-delegated") that indicates the token request is for the non-delegated "Subscribe-Notify" interaction scenarios which is the NF\_A subscribes NF services offered by the NF\_B for itself.

2. Upon reception of the Nnrf\_AccessToken\_Get Request, the NRF obtains the NF Instance ID of the Service Consumer (NF\_A), the Notification URI of the Service Consumer (NF\_A) and the indication\_non-delegated from the NF\_A. The NRF then determines whether the NF\_A is authorized to subscribe the services provided by the NF\_B, based on the locally configured policies or authorization information. If the authorization is successful, the NRF generates a token which includes the instance ID of NF\_A, the Notification URI of the NF\_A, and the indication\_non-delegated.

3. The NRF sends the Nnrf\_AccessToken\_Get Response to the NF\_A with the token.

4. The NF\_A sends a subscription request, which contains the token and the associated notification endpoint of the NF\_A (i.e. the Notification URI of the NF\_A), to the NF\_B.

5. Upon reception of the subscription request, the NF\_B verifies the integrity of the token and verify the claims in the token.

6. If the verification of the token is successful, the NF\_B sends the subscription response to the NF\_A.

7. Optionally, the NF\_B may send a notification to the NF\_A after certain conditions are met.



Figure 6.35.2-2: Service access authorization in the non-delegated "Subscribe-Notify" interaction scenarios for roaming.

1. The NF\_A send the Nnrf\_AccessToken\_Get Request to the cNRF (the NRF in the PLMN of Consumer). The token request contains the instance ID of the NF\_A, the Notification URI of the NF\_A, and an indication (Expressed as "Indication\_non-delegated") that indicates the token request is for the non-delegated "Subscribe-Notify" interaction scenarios which is the NF\_A subscribes services offered by the NF\_B for itself.

2. The cNRF identify the pNRF (the NRF in the PLMN of Producer) based on the home PLMN ID, and request an access token from the pNRF.

3. Upon reception of the Nnrf\_AccessToken\_Get Request, the pNRF obtain the NF Instance ID of the Service Consumer (NF\_A), the Notification URI of the Service Consumer (NF\_A) and the indication\_non-delegated from the NF\_A. The pNRF then determine whether the NF\_A is authorized to subscribe services provided by the NF\_B, based on the locally configured policies or authorization information. If the authorization is successful, the pNRF generate a token which includes the instance ID of NF\_A, the Notification URI of the NF\_A, and the indication\_non-delegated.

4. The pNRF send the Nnrf\_AccessToken\_Get Response to the cNRF with the token.

5. The cNRF forward the Nnrf\_AccessToken\_Get Response to the NF\_A with the token.

6. The NF\_A send a subscription request, which contains the token and the associated notification endpoint of the NF\_A (i.e. the Notification URI of the NF\_A), to the NF\_B.

7. Upon reception of the subscription request, the NF\_B verify the integrity of the token and verify the claims in the token.

8. If the verification of the token is successful, the NF\_B send the subscription response to the NF\_A.

9. Optionally, the NF\_B may send a notification to the NF\_A after certain conditions are met.

### 6.35.3 Evaluation

The solution extends the existing token-based authorization method by including the Notification URI of the NF\_A and the indication\_non-delegated in the token claims. The token is used to indicate that the NF\_A is authorized to subscribe services offered by NF\_B.

The solution fulfils the potential security requirements from Key Issue #30: "The 5G system support an authorization mechanism for the "Subscribe-Notify" scenarios for the scenario that NF\_A subscribes the service of NF\_B for itself." In this solution, only NF\_A, NF\_B, and NRF are involved in the non-delegated scenario, without the NF consumer of NF\_C. Hence, the NRF could determine whether the NF\_A is authorized or not according the NF\_A's profile stored in the NRF.

The impact is that the indication\_non-delegated is required to inform the NRF, and URI of NF\_A be added into the token.

# 7 Conclusions

## 7.1 Conclusion on KI #20

The KI #20 is about protection of SeCoP interfaces to other 5GC Network functions.

Solution #17 is recommended to be used for normative specification.

## 7.2 Conclusions on Key Issue #21: Secure message transport via the SeCoP

The potential requirement "The SeCoP provide confidentiality, integrity and replay protection for its internal communication over SeCoP internal network interfaces" is basis for normative requirements. Its solution is not specified and left to implementation.

## 7.3 Conclusions on Key issue #22: Authorization of NF service access in indirect communication

For Scenario C (Indirect communication without delegated discovery), no normative changes to TS 33.501 [11] are needed, as described in Solution #16: (OAuth 2.0 based authorization for Indirect communication without Delegated Discovery (Model C)) and Solution #24 (Token-based authorization for Scenario C using stateless SeCoP).

NOTE 1: The conclusion for Scenario D (Indirect communication with delegated discovery) is not addressed in the present document.

## 7.4 Conclusion on KI #23

The KI #23 is about authentication and authorization between two NFs in indirect communication scenarios.

Solution #22 is recommended for normative work, with the exception of service access authorization in Scenario D where there is no conclusion.

Token-based authorization between NF consumer and NF producer in indirect communication scenarios is addressed by the conclusions on Key Issue #22 "Authorization of NF service access in Indirect Communication" in clause 7.3.

## 7.5 Conclusion on KI #24

The KI #24 is about service access authorization based on NF Set.

It is concluded that NF Set granularity for token-based authorization will be included in normative work.

## 7.6 Conclusions on Key issue #25: Indirect communication in roaming scenarios

Solution #34 is recommended as basis for normative work. Details of communication between SECOP and SEPP will be handled during normative work.

## 7.7 Conclusion on KI #26

The KI #26 is about protection of Inter-PLMN user traffic on the N9 interface.

Solution #18 provides an NDS/IP based solution for protection of user traffic on inter-PLMN N9 interface. This is recommended to update TS 33.501[11] clause 9.9 during normative phase.

## 7.8 Conclusion on KI #27

The KI #27 is about support of a UP gateway function on the N9 interface.

Solution #20 provides a solution for filtering GTP-U traffic on the inter-PLMN N9 interface.

Following aspects of the solution are recommended for normative specification:

- A new function - UP Gateway Function (UPGF), is introduced for filtering GTP-U traffic on the inter-PLMN N9 interface.

NOTE: Whether or not a UPGF is deployed on the inter-PLMN N9 interface is based on operator policy.

- UPGF will interface with SMF to obtain GTP Tunnel Information.

- Protocol between SMF and the UPGF uses the existing PFCP protocol.

- UPGF may be deployed as a collocated function within the UPF or as a separate function handling one or more UPFs.

## 7.9 Conclusion on KI #29

The KI #29 is about support for resource level authorization of NF consumers.

It is concluded that resource-based authorization will be addressed during normative work.

Annex A :  
Void

Annex B:  
Options for integrity protection on the N32 interface

The JSON framework offers three cryptographic mechanisms for integrity protection: keyed MACs, digital signatures and authenticated encryption with additional data (AEAD). Keyed MACs and AEAD are symmetric mechanisms while digital signatures are asymmetric.

JSON Web Signatures (JWS) [11] provide integrity protection for arbitrary data using MACs or digital signatures. JSON Web Encryption (JWE) [2] represents encrypted content using JSON-based data structures. All content encryption algorithms in JWE are authenticated encryption algorithms, meaning that these algorithms provide integrity protection of the data, as well as confidentiality protection. All content encryption algorithms in JWE permit the inclusion of Additional Authenticated Data (AAD). This is data which is integrity protected but not encrypted. Therefore, JWE can be used when confidentiality protection is only required for certain IEs. Whether JWE can be used when no encryption is required is undefined.

Table B-1: Comparison of options for integrity protection on N32 interface

|  |  |  |  |
| --- | --- | --- | --- |
|  | MAC | Digital signature | Authenticated encryption |
| Confidentiality and integrity protection achievable within JOSE framework | Yes – JWE encapsulating JWS or JWS including JWE | Yes – JWE encapsulating JWS or JWS including JWE | Yes – JWE only |
| Allows integrity protection only | Yes | Yes | Undefined |
| Signature/MAC size(s) | JWS: 256 bits – 512 bits  JWE: 128 bits | JWS: 512 bits – 2048 bits  JWE: 128 bits | JWE: 128 bits |
| Non-repudiation | No | Yes | No |
| Confidentiality protection for specific IEs | Yes | Yes | Yes |
| Separate keys for integrity protection and encryption | Yes | Yes | Maybe (algorithm dependent) |
| Allows IPX modifications | Maybe – Different keys for endpoints and IPXs preferred. | Yes - IPXs need own certificates for signing message changes. Multiple signatures can be added to one message. | Maybe – Different keys for endpoints and IPXs required. |

When both encryption and integrity protection are required the simplest mechanism for integrity protection for SEPP-SEPP communications is authenticated encryption, which is provided by all JWE encryption algorithms. This requires the least overhead of all options, both in terms of bandwidth and processing. Using an authenticated encryption mechanism reduces complexity, making it less likely that mistakes will be made in securing messages. Authenticated encryption algorithms also reduce the possibility of combining integrity and encryption algorithms in an insecure manner.

When integrity protection alone is required the behaviour of JWE is undefined, so JWS is a more appropriate mechanism for messages which require no encryption. MACs are preferable to signatures in this scenario due to their reduced overhead. Alternatively, JWE could be used, with a defined "null" value for the JWE plaintext.

An IPX might not have a relationship with every operator to whom it routes a message, hence agreeing shared keys might be difficult. Therefore, digital signatures are the most appropriate integrity protection mechanism for IPX modifications. The disadvantage of using digital signatures is that they add an overhead to communications in terms of bandwidth and a cryptographic overhead for signing and verification. Therefore, addition of digital signatures to every modified message could significantly increase the IPX's processing requirements.

Annex C:  
Deployment options for the UP gateway

# C.1 Deployment option 1: UP Gateway per slice

NOTE 1: The scenario where N9 terminates in UPGW is not addressed in the present document.

NOTE 2: Not all the scenarios are addressed in the current document. Alignment with TS 23.501 is not addressed in the current document.

In such a deployment each network slice has own UP Gateway instances. In such case, as shown in Figure C.1-1, the N? control interface between SMF and UP Gateway as well as the N9 interface between UPF and UP Gateway remain network slice internal.

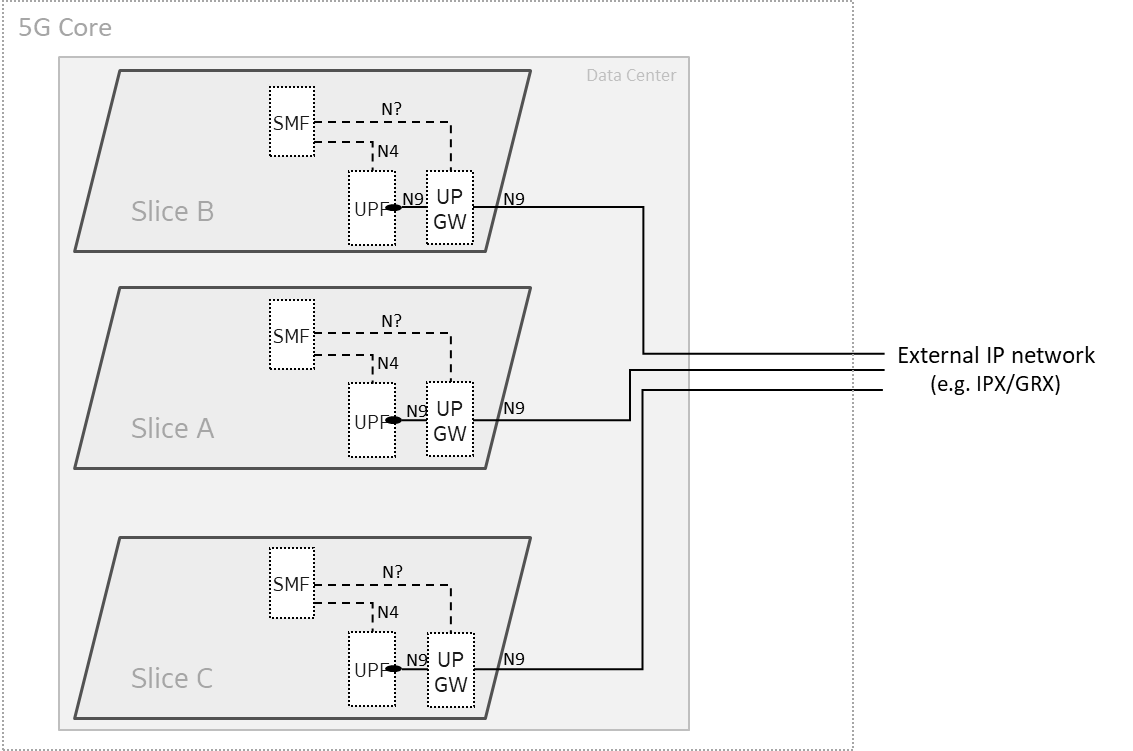


Figure C.1-1: UP Gateway deployed per network slice

# C.2 Deployment option 2: UP Gateway as shared appliance

In this case, as shown in Figure C.2-1, the UP Gateway is a shared function potentially even as a stand-alone appliance. In such deployments there will be numerous CP and UP interfaces from different network slices connected to the UP Gateway.

However, in this case UP Gateway needs to ensure that security of slices is not compromised. That means for cases where UP Gateway handles traffic of multiple network slices the UP Gateway is responsible to ensure slice security is not compromised.

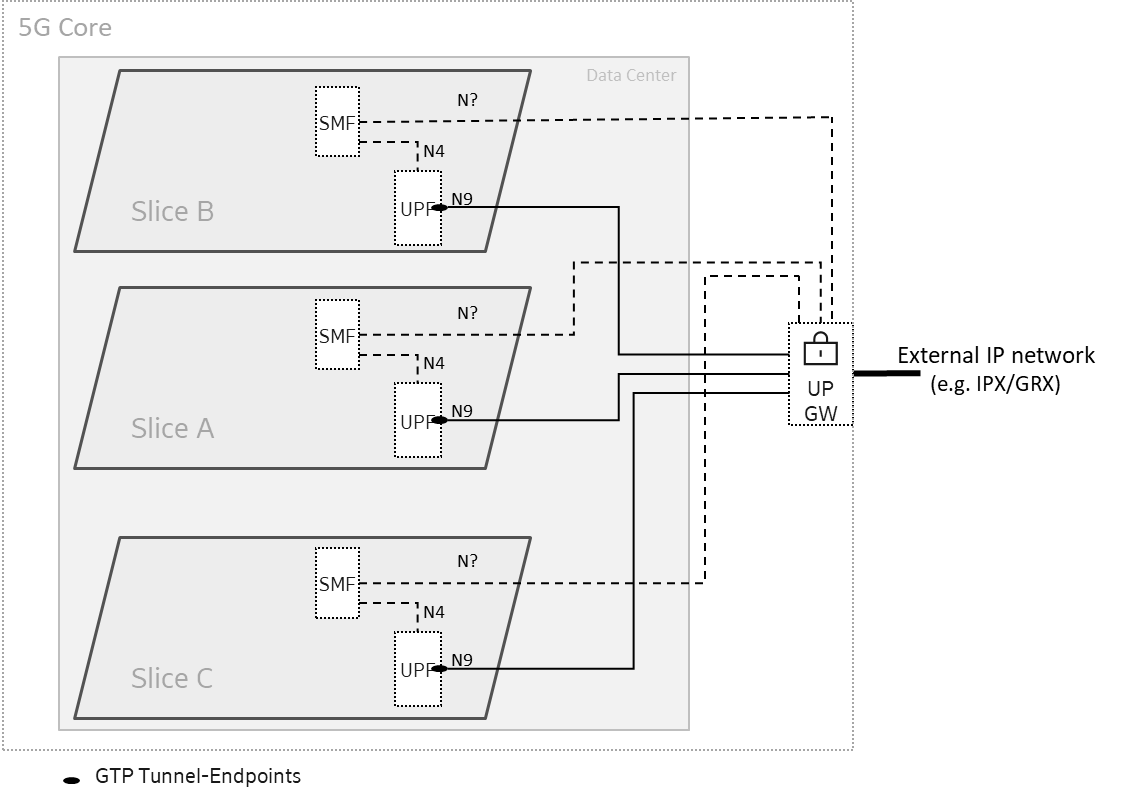


Figure C.2-1: UP Gateway deployment as shared appliance

Annex D:  
Change history

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
| 05/2018 | SA3#91bis | S3-181964 | - | - | - | First draft based on living document S3-181715 | 0.1.0 |
| 05/2018 | SA3#91bis | S3-182090 | - | - | - | Updates based on new living document S3-181945 | 0.2.0 |
| 2018-06 | SA#80 | SP-180456 |  |  |  | Presented for information | 1.0.0 |
| 2018-07 |  |  |  |  |  | Editorial fixes (EditHelp + MCC) | 1.0.1 |
| 2018-08 | SA3#92 | S3-182683 |  |  |  | Version after SA3#92 incorporating changes from S3-182503 | 1.1.0 |
| 2018-09 | SA3#92bis | S3-183058 |  |  |  | Version after SA3#92bis incorporating changes from S3-183093, S3-183094, S3-183095, S3-183096  Changes Release to 16, as study is not concluded in Rel-15 timeframe | 1.2.0 |
| 2018-11 | SA3#93 | S3-183724 |  |  |  | Version after SA3#93 incorporating changes from S3-183723 | 1.3.0 |
| 2019-03 | SA3#94-AH | S3-191031 |  |  |  | Version after SA3#94-AH incorporating changes from S3-190964, S3-190965, S3-190967, S3-190968, S3-190980, S3-190981, S3-190982, S3-190983, S3-190984 | 1.4.0 |
| 2019-05 | SA3#95 | S3-191667 |  |  |  | Version after SA3#95 incorporating changes from S3-191175, S3-191404, S3-191490, S3-191521, S3-191661, S3-191668, S3-191669, S3-191671, S3-191672, S3-191673, S3-191674 | 1.5.0 |
| 2019-06 | SA3#95bis | S3-192438 |  |  |  | Version after SA3#95bis incorporating changes from S3-192439, S3-192440, S3-192035, S3-192441, S3-192412, S3-192249, S3-192413, S3-192252, S3-192443, S3-192444, S3-192442, S3-192258 | 1.6.0 |
| 2019-08 | SA3#96 | S3-193065 |  |  |  | Version after SA3#96 incorporating changes from S3-193064, S3-102606, S3-192607, S3-192802, S3-193066, S3-193067, S3-193068, S3-193069, S3-193070, S3-192815, S3-193174, S3-193175, S3-193077, S3-193078, S3-192688, S3-192689, S3-193079, S3-192811, S3-193081, S3-193082, S3-193095, S3-193097, S3-192608, S3-193098, S3-193099, S3-193100 | 1.7.0 |
| 2019-10 | SA3#96adhoc | S3-193729 |  |  |  | Version after SA3#96adhoc incorporating changes from S3-193725, S3-193726, S3-193728, S3-193615, S3-193730 | 1.8.0 |
| 2019-11 | SA3#97 | S3-194516 |  |  |  | Version after SA3#97 incorporating changes from S3-194505, S3-194506, S3-194507, S3-194508, S3-194509, S3-194510, S3-194511, S3-194515 | 1.9.0 |
| 2020-06 | SA#88-e | SP-200384 |  |  |  | EditHelp and MCC review  Presented for approval | 2.0.0 |
| 2020-07 | SA#88-e |  |  |  |  | Upgrade to change control version | 16.0.0 |
| 2020-09 | SA#89e | SP-200712 | 0001 | - | F | Clean-up, including removal of Editor's Notes | 16.1.0 |