Extended Subscribers Authentication Technical Specifications

ONGO-TS-1003

V4.1.0

2022-10-18

LEGAL NOTICES AND DISCLOSURES

OnGo Alliance
3855 SW 153rd Drive, Beaverton, OR 97003
www.ongoalliance.org
info@ongoalliance.org

Table of Contents

Table of Figures

Table of Tables

1 Introduction and Scope

This document defines UE and service provider network architecture and protocols for extended subscribers authentication methods: Certificate Based Subscribers Authentication (CBSA) and non-Certificate Based Subscribers Authentication. These extended authentication mechanisms are defined for both the NHN and the 3GPP-based (non-EPS-AKA) Access Modes for 4G/LTE. The Stage 2 and 3 Aspects are described in CBRS TS 1002 [3] and together provide extended mechanisms for subscribers authentication.

This document also specifies extended authentications for 5GC. 3GPP TS 33.501 [45] defines four authentication methods including 5G-AKA, EAP-AKA', EAP-TLS, and EAP-TTLS for UE accessing 5GC. This document specifies two additional deployment models of EAP-TTLS for 5GC.

1.1 Key Words

The key words "required", "shall", "shall not", "should", "should not", "recommended", "may", and "optional" in this document are to be interpreted as described in RFC-2119 [7]. In addition, the key word "conditional" shall be interpreted to mean that the definition is an absolute requirement of this specification only if the stated condition is met.

2 References

[1] CBRS Network Services Technical Report, CBRSA-TR-1001 v 1.0.0

[2] CBRS Network Services Requirements Technical Specification, CBRSA-TS-1001 v 3.0.0 (Release 3)

[3] CBRS Network Services Technical Specification Stage 2 and 3, CBRSA-TS-1002 v 3.0.0 (Release 3)

[4] The Internet Engineering Task Force (IETF). Improved Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA'), edited by J. Arkko, V. Lehtovira, and P. Eronen, May 2009. IETF RFC 5448, also available at https://tools.ietf.org/html/rfc5448.

[5] The Internet Engineering Task Force (IETF). The EAP-TLS Authentication Protocol, edited by D. Simon, B. Aboba, and R. Hurst, March 2008. IETF RFC 5216, also available at https://tools.ietf.org/html/rfc5216.

[6] MulteFire Alliance (MFA), Architecture for Neutral Host Network Access Mode Stage 2 (Release 1), MFA TS MF.202 V1.0.3 (2017-06).

[7] The Internet Engineering Task Force (IETF). Key words for use in RFCs to Indicate Requirement Levels, edited by S. Bradner, March 1997. IETF RFC 2119, also available at https://tools.ietf.org/html/rfc2119.

[8] The Internet Engineering Task Force (IETF). Extensible Authentication Protocol Tunnelled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0), edited by P. Funk and S. Blake-Wilson, August 2008. IETF RFC 5281, also available at https://tools.ietf.org/html/rfc5281

[9] The Internet Engineering Task Force (IETF). Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, edited by D. Cooper et al., May 2008. IETF RFC 5280, also available at https://tools.ietf.org/html/rfc5280.

[10] The Internet Engineering Task Force (IETF). Microsoft PPP CHAP Extensions, Version 2, edited by G. Zorn, January 2000. IETF RFC 2759, also available at https://tools.ietf.org/html/rfc2759.

[11] The Internet Engineering Task Force (IETF). X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP, edited by M. Myers et al., June 1999. IETF RFC 2560, also available at https://tools.ietf.org/html/rfc2560.

[12] The Internet Engineering Task Force (IETF). The Lightweight Online Certificate Status Protocol (OCSP) Profile for High-Volume Environments, edited by A. Deacon and R. Hurst, September 2007. IETF RFC 5019, also available at https://tools.ietf.org/html/rfc5019.

[13] The Internet Engineering Task Force (IETF). X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP, edited by S. Santesson et al., June 2013. IETF RFC 6960, also available at https://tools.ietf.org/html/rfc6960.

[14] The Internet Engineering Task Force (IETF). The Transport Layer Security (TLS) Multiple Certificate Status Request Extension, edited by Y. Pettersen, June 2013. IETF RFC 6961, also available at https://tools.ietf.org/html/rfc6961.

[15] The Internet Engineering Task Force (IETF). Extensible Authentication Protocol (EAP) Key Management Framework, edited by B. Aboba et Al., August 2008. IETF RFC 5247, also available at https://tools.ietf.org/html/rfc5247.

[16] 3GPP TS 23.401, "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access", v15.8.0, June 2019.

[17] 3GPP TS 33.401 “3GPP System Architecture Evolution: Security Architecture”. v15.8.0, June 2019.

[18] 3GPP TS 23.122, “Non-Access-Stratum (NAS) functions related to Mobile Station (MS) in idle mode”.

[19] 3GPP TS 36.304 v14.3.0, "Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) procedures in idle mode".

[20] 3GPP TS 36.331 v14.3.0, " Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification".

[21] 3GPP TS 33.402 v14.2.0, "Security aspects of non-3GPP accesses".

[22] 3GPP TS 37.340 v15.7.0, Evolved Universal Terrestrial Radio Access (E-UTRA) and NR".

[23] 3GPP TS 23.402 v14.4.0, " Architecture enhancements for non-3GPP accesses".

[24] 3GPP TS 24.301 v14.3.0, “Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS)”, http://www.3gpp.org/

[25] National Security Agency (NSA). Commercial National Security Algorithm Suite (CNSA), available at https://www.iad.gov/iad/programs/iad-initiatives/cnsa-suite.cfm

[26] The Internet Engineering Task Force (IETF). Extensible Authentication Protocol (EAP), edited by B. Aboba et al., June 2004. IETF RFC 3748, also available at https://tools.ietf.org/html/rfc3748.

[27] The Internet Engineering Task Force (IETF). The Transport Layer Security (TLS) Protocol Version 1.2, edited by T. Dierks et al., August 2008. IETF RFC 5246, also available at https://tools.ietf.org/html/rfc5246.

[28] The Internet Engineering Task Force (IETF). Tunnel Extensible Authentication Protocol (TEAP) Version 1, edited by H. Zhou et al., May 2014. IETF RFC 7170, also available at https://tools.ietf.org/html/rfc7170.

[29] The Internet Engineering Task Force (IETF). The Flexible Authentication via Secure Tunneling Extensible Authentication Protocol Method (EAP-FAST), edited by N. Cam-Winget et al., May 2007. IETF RFC 7170, also available at https://tools.ietf.org/html/rfc7170.

[30] Microsoft Corporation, "[MS-PEAP]: Protected Extensible Authentication Protocol (PEAP)", February 2014.

[31] The Internet Engineering Task Force (IETF). Requirements for a Tunnel-Based Extensible Authentication Protocol (EAP) Method, edited by K. Hoeper et al., July 2012. IETF RFC 6678, also available at https://tools.ietf.org/html/rfc6678.

[32] MulteFire Alliance (MFA), Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS) Stage 3 (Release 1), MFA TS 24.301 V1.0.2 (2017-02).

[33] The Internet Engineering Task Force (IETF). RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP), edited by B. Aboba and P. Calhoun, September 2003. IETF RFC 3579, also available at https://tools.ietf.org/html/rfc3579.

[34] The Internet Engineering Task Force (IETF). Diameter Base Protocol, edited by V. Fajardo, Ed. et al., October 2012. IETF RFC 6733, also available at https://tools.ietf.org/html/rfc6733.

[35] The Internet Engineering Task Force (IETF). Realm-Based Redirection In Diameter, edited by T. Tsou et al., November 2013. IETF RFC 7075, also available at https://tools.ietf.org/html/rfc7075.

[36] The Internet Engineering Task Force (IETF). The Network Access Identifier, edited by A. DeKoket, May 2015. IETF RFC 7542, also available at https://tools.ietf.org/html/rfc7542.

[37] The Internet Engineering Task Force (IETF). The Transport Layer Security (TLS) Protocol Version 1.1, edited by T. Dierks et al., April 2006. IETF RFC 4346, also available at https://tools.ietf.org/html/rfc4346.

[38] The Internet Engineering Task Force (IETF). The TLS Protocol Version 1.0, edited by T. Dierks et al., January 1999. IETF RFC 2246, also available at https://tools.ietf.org/html/rfc2246.

[39] Third Generation Partnership Project (3GPP). Numbering, addressing and identification. 3GPP TS 23.273.

[40] The Internet Engineering Task Force (IETF). Microsoft Vendor-specific RADIUS Attributes, edited by G. Zorn et al., March 1999. IETF RFC 2548, also available at https://tools.ietf.org/html/rfc2548.

[41] The Internet Engineering Task Force (IETF). Remote Authentication Dial-In User Service (RADIUS) Protocol Extensions, edited by A. DeKok and A. Lior, April 2013. IETF RFC 6929, also available at https://tools.ietf.org/html/rfc6929.

[42] The Internet Engineering Task Force (IETF). Credentials Provisioning and Management via EAP (EAP-CREDS), edited by M. Pala, April 2019. IETF I-D available at https://datatracker.ietf.org/doc/draft-pala-eap-creds/.

[43] The Internet Engineering Task Force (IETF). Cryptographic Message Syntax (CMS), edited by R. Housley, September 2009. IETF RFC 5652, also available at https://tools.ietf.org/html/rfc5652.

[44] 3GPP TS 23.003 v16.2.0, Numbering, addressing and identification.

[45] 3GPP TS 33.501 v17.5.0, Security Architecture and Procedures for 5G System.

3 Definitions and Abbreviations

3.1 Definitions

None

3.2 Abbreviations

Abbreviation	Explanation
5GC	5G Core Network
AAA	Authentication, authorization and accounting
ABBA	Anti-Bidding down Between Architectures
AKA	Authentication and Key Agreement
AUSF	Authentication Server Function
CBRS	Citizens Broadband Radio Service
CBSA	Certificate Based Subscribers Authentication
CA	Certification Authority
CRL	Certificate Revocation List
CSG	Closed Subscriber Group
CSR	Certificate Signing Request
DER	Distinguished Encoding Rules
EAP	Extensible Authentication Protocol
EE	End Entity
EN-DC	E-UTRAN New Radio – Dual Connectivity
EPS	Evolved Packet System
EST	Enrollment over Secure Transport
IMSI	International Mobile Subscriber Identity
MeNB	Master eNB
MNO	Mobile Network Operator
NAS	Non-Access Stratum
ngKSI	Key Set Identifier in 5G
NH	Neutral Host
NHN	Neutral Host Network
NPN	Non-Public Network

Abbreviation	Explanation
NSSAAF	Network Slice specific and SNPN Authentication and Authorization Function
NW	Network
OCSP	Online Certificate Status Protocol
PKI	Public Key Infrastructure
PSP	Participating Service Provider
PLMN	Public Land Mobile Network
RAN	Radio Access Network
SEAF	SEcurity Anchor Function
SIDF	Subscription Identity De-concealing Function
SIM	Subscriber Identity Module
SgNB	Secondary gNB
SNPN	Standalone Non-Public Network
SP	Service Provider
SUCI	Subscription Concealed Identifier
SUPI	Subscription Permanent Identifier
TLS	Transport Layer Security
TTLS	Tunneled Transport Layer Security
UDM	Unified Data Management
UE	User Equipment
USIM	Universal Subscriber Identity Module

4 Extended Subscribers Authentication

4.1 General

This document specifies extended authentication mechanisms for both NHN and 3GPP-based (non-EPS-AKA) Access Modes. In both cases, the call flow for EAP is depicted in Figure 4-1 where the extended authentication replaces the 3GPP EPS-AKA one (both in NHN and 3GPP-based Access Modes).

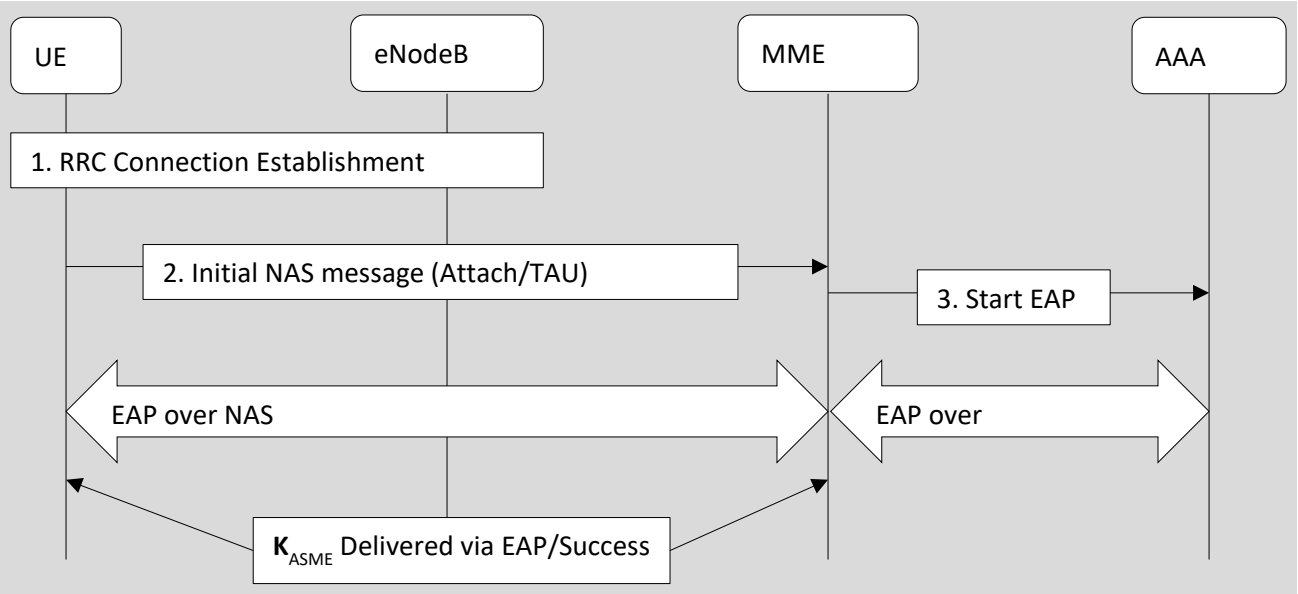

Figure 4-1: Overview of EAP-Based Authentication Procedure Message Flow

This call flow also applies to EN-DC [22]. In particular:

1. The UE establishes an RRC connection with the eNodeB (or the eNB in 3GPP-based Access Mode).
2. The UE sends an Initial NAS message (e.g. Attach/TAU Request) to the NH-MME (for NHN Access Mode) or to the SP's MME' (for 3GPP-based Access Mode [non-EPS-AKA]).
    a. The attach procedure continues as identified in 3GPP TS 23.401 Section 5.3.2.1 [16] up to (and including) the Identity Response from the UE.
3. The local MME initiates (NH-MME or SP MME') the EAP authentication process.
    a. When in NHN Access Mode the NH-MME initiates EAP authentication, and notifies the EAP authenticator function to initiate EAP authentication.
    b. When in 3GPP-based Access Mode (non-EPS-AKA), the SP's MME' initiates the EAP authentication and notifies the EAP authenticator function to initiate the EAP authentication based on the UE-provided IMSI value (i.e., the 3GPP-based Access Mode is selected if the presented IMSI is configured for extended authentication on the SP's MME).
4. The EAP authentication takes place over the NAS transport both in NHN and 3GPP-based Access Modes. In particular:
    a. a. In NHN Access Mode, the identity used by the UE in response to the EAP Identity Request packet shall be provided in the Network Access Identifier (NAI) form in the EAP payload sent by

the UE. The EAP packets exchange continues between the SP's non-3GPP AAA server through the Local AAA Proxy. The interface between the EAP Authenticator and AAA is based on RADIUS or Diameter. The interface between the Local AAA proxy and the SP's AAA is based on RADIUS or Diameter. The interface between the Local AAA proxy and the non-3GPP AAA is based on RADIUS or Diameter and follows specifications for the SWa interface for Untrusted access mode and STa interface for Trusted access mode, as defined in 3GPP TS 29.273 [39].

b. b.In 3GPP-based Access Mode (non-EPS-AKA), the identity used by the UE in response to the EAP Identity Request packet shall be provided in the IMSI form in the EAP payload sent by the UE. The EAP packets exchange continues between the SP's non-3GPP AAA server and the SP's MME'. Also in this case, the interface between the EAP Authenticator and AAA is based on RADIUS or Diameter.

Upon successful authentication, the UE and the NH-MME (for NHN Access Mode) or the SP's MME' (for 3GPP-based Access Mode) derive the KASME from the EAP keying material (MSK) as defined in Section 5.12.4 of MFA MF.202 TS [6]. In particular, the KASME is defined as the 256 MSB (i.e., 32 Bytes) of the MSK that is generated as part of the TLS negotiation between the UE and the AAA server.

After this point the MME (NH-MME for NHN Access Mode or SP's MME' for 3GPP-based Access Mode) indicates to the UE that the NAS security is activated by sending a Security Mode Command to the UE. Subsequent key derivation from KASME for protecting NAS, RRC, and UP are defined in [17]. In particular for EN-DC, the MeNB generates and sends S-KgNB to the SgNB from which KSgNB-UP-enc, KSgNB-RRC-int and KSgNB-RRC-enc are then derived. The same derivation of S-KgNB and the subsequent keys also occurs at the UE side.

The MME continues the NAS procedure. For example, for attach procedure Steps 17 to 24 as listed in Section 5.3.2.1 of 3GPP TS 23.401 [16] are performed.

4.1.1 EAP-based Subscribers Authentication

CBRS supports extended authentication mechanisms for both NHN and 3GPP-based (non-EPS-AKA) Access Modes via the EAP protocol. The use of EAP provides an extensible approach that allows to support multiple authentication methods without requiring modifications to the network architecture.

The selection of the EAP mechanism used for subscriber authentication happens at the SP AAA server in response of the EAP-RSP/Identity initial authentication request packet [15]. With the introduction of additional methods and the possibility to support multiple types of credentials at once (e.g. some UE might support multiple mechanisms and/or might have multiple type of credentials that might be used for the same subscription – e.g., USIM-based and X.509 certificate), the selection of the appropriate authentication method might require some additional application logic on the SP's AAA (for NHN Access Mode) or on the SP's MME (for 3GPP-based non-EPS-AKA Access Mode).

In the most common case, the SP will be able to pre-select the authentication mechanism specific for that subscriber's device by maintaining the association between the subscriber's identity (e.g., IMSI or NAI) and the type of credentials that have been issued during the subscriber's registration process. Thus, in practice, when responding to the *EAP-RSP/Identity* packet, the SP AAA can still pre-select the appropriate EAP mechanism directly without requiring the implementation of additional procedures. For example, if the subscriber registered its account with the SP by using username and password (e.g., via a web portal), the initial EAP response from the SP AAA server shall use the EAP-TTLS [8] Start packet and the authentication will proceed by using MS-CHAP-V2 [10] as the inner method of the EAP-TTLS authentication. For a subscriber whose equipment was provisioned with an X.509 certificate during the registration process, the SP AAA server will use

the EAP-TLS [5] Start packet instead. For USIM-based subscribers, the EAP-AKA' may be selected as the authentication mechanisms as usual. Notice that other EAP mechanisms may be selected in case the SP and the UE provide support for them.

4.1.2 Direct and Indirect Server Authentication

Depending on which EAP authentication mechanism is selected for subscriber authentication, the UE authenticates the AAA server indirectly (EAP-AKA') or directly (EAP-TLS or EAP-TTLS).

When EAP-AKA' is used for subscriber's authentication, the identity of the server is indirectly verified by using the shared key inside the USIM in the key derivation process. In this case, the provisioning of the trusted secret for authentication is achieved via the provisioning of the USIM that serves as the base of trust for the UE.

On the contrary, when EAP-TLS or EAP-TTLS methods are used for subscribers' authentication, the UE is required to directly authenticate the AAA Server by verifying that the server's certificate is trusted, not expired, and not revoked. In particular, for the UE to be able to verify the server's credentials, the Trust Anchor that provides the root of trust for the server's certificate must be securely stored in the UE. This trust anchor is usually installed together with the UE credentials during the subscriber's registration process.

4.1.3 EAP mechanisms negotiation

In case multiple types of credentials are associated with a subscriber's identity or if the EAP mechanism selected by the SP is not supported by the UE, the UE may negotiate a different mechanism with the AAA server.

In particular, when the UE authenticates to the network and it does not support the EAP mechanism pre-selected by the SP, the UE should follow the procedures described in RFC 3748 [26] to negotiate the appropriate EAP mechanism that is supported.

It is important to notice that enabling negotiation of EAP mechanisms might raise some security concerns. In particular, the negotiation process is vulnerable to downgrade attacks where an attacker with full network access can force the EAP endpoints to negotiate a less secure method. Therefore, in the context of CBRS NHN and 3GPP-based (non-EPS-AKA) network authentication, the supported EAP mechanisms shall have similar or higher security level than the ones described in this document to prevent granting any advantage to the attacker.

4.1.4 EAP Tunneling mechanisms

Support for EAP Tunneling methods is required when extended authentication methods (non-Certificate-Based) are used to protect the integrity and secrecy of the secrets. There are several existing tunnel-based EAP mechanism that use Transport Layer Security (TLS) [27] to establish the secure tunnel. EAP mechanisms supporting this include Protected EAP (PEAP) [30], EAP Tunneled Transport Layer Security (EAP-TTLS) [8], and EAP Flexible Authentication via Secure Tunneling (EAP-FAST) [29].[1]

The latest tunnel-based EAP mechanism that has been standardized at IETF is the Tunnel Extensible Authentication Protocol (TEAP) [28] which provides a generic approach to EAP tunneling which is compatible with the requirements for EAP tunneling protocols described in RFC 6678 [31]. Although EAP-TEAP provides

[1] It is to be noted that all these methods are either vendor-specific or informational, and the industry calls for a Standards Track tunnel-based EAP method.

additional features that might be interesting for deployment (e.g., simple certificate management via EST), its current level of adoption is quite limited.

This specification defines the procedures to deploy EAP-TTLS method for providing EAP tunneling capabilities, however operators and equipment providers may support additional ones.

4.2 Non-Certificate-Based Subscribers Authentication

SPs may support subscribers' authentication methods that use credentials other than EPS-AKA and X.509 Certificates. In this case, SPs must support the EAP-TTLS method as described in the rest of this section and may support other methods. EAP-TTLS comprises two phases: the TLS handshake phase (also called phase 1) and the TLS tunnel phase (also called phase 2).

During phase 1, TLS is used to authenticate the TTLS server and, optionally, the client to the server via optional client certificates request. During this phase, the selection of a cipher suite and its activation allows for the next phase to proceed securely by using the TLS record layer.

During phase 2, instead, the information exchanged between the client and the server (e.g., user authentication) is exchanged via attribute-value pairs (AVPs) that are encrypted by using the cipher selected during the TLS negotiation.

Other tunneled methods (e.g., EAP-PEAP or EAP-TEAP) supported by SPs and UEs are out of scope of this document.

4.2.1 Security Considerations

The use of EAP-TTLS allows for the UE to cryptographically verify the identity of the AAA server (direct verify) when the appropriate Trust Anchor is available in the UE. However, it is important to notice that the security of the subscriber's authentication heavily depends on the quality of the secret selected (i.e., the password) during the subscriber's registration. The SP shall ensure that best practices for password management are properly followed to provide an adequate level of security during the authentication process.

Another important consideration related to the use of username and password is the possibility for malicious actors to guess the user's credentials. Differently from the EAP-TLS case, the attacker might attempt to guess credentials by trying many different username and password combinations. SPs that support this authentication mechanism should provide sufficient protection (e.g., throttling, disabling of a subscriber's account, etc.) to mitigate this type of attack.

4.3 Certificate Based Subscribers Authentication (CBSA)

Certificate Based Subscribers Authentication (CBSA) and key agreement shall be performed using the Extensible Authentication Protocol (EAP) RFC 5247 [15]. In particular, the EAP-TLS method shall be used when the UE is already provisioned with a valid X.509 certificate for the subscriber.

4.3.1 Security Considerations

Although CBSA provides high level of security, it is important that the procedures for certificate validation and revocation processing are implemented according to standard specification and best practices. Moreover, SPs shall use standard schemes and algorithms for certificates, public keys and certificate signing. It is suggested

that SPs follow the Commercial National Security Algorithm Suite (CNSA Suite) in their implementation for their PKIs and crypto parameters wherever possible.

An important deployment consideration about the use of CBSA is that since the subscriber's secret (i.e., the private key) is never shared or stored in the SP system, its use removes the threat of an attacker stealing the subscribers' authentication credentials by attacking the SP's servers. This differs from the case where EAP-TTLS or EAP-AKA' are used since the use of symmetric secrets requires both parties to have access to them, thus requiring SPs to store the secrets in their systems.

4.3.2 Internal vs. External X.509 Credentials Validation

One of the key aspects of CBSA is the provisioning and management of X.509 certificates for the UE. Some SPs might decide not to offer certificate provisioning for UEs (i.e., they will not directly issue certificates for their subscribers), and still want to leverage CBSA to extend the range of services offered to their customers.

In this case, SPs might decide to accept credentials issued and provisioned by third parties. For example, this could include certificates issued to the UE in the context of WiFi registration or certificates issued on behalf of the SP by a Certificate Service Provider.

Whatever the choice by the SP might be, by using CBSA, the SP has the ability to combine all the above options by simply adding the required Trust Anchors (i.e., Root CAs certificates or Public Keys) to the list of trusted authorities in the SP's AAA infrastructure without requiring the sharing of credentials' databases with the third parties that provide and manage the UE's certificates.

4.3.3 Restricting Trust to a specific branch of a PKI

Sometimes PKIs can have complex connections and multiple SubCAs dedicated to specific use. It is common practice, for example, to have, in the same infrastructure, a single Trust Anchor that issues "scoped" SubCAs (e.g., a Device SubCA, a Server SubCA, and a Code Signing SubCA).

SPs can decide to restrict trust for subscribers' authentication to a specific subset of the SubCAs issued under a Trust Anchor (i.e., only certificates issued by the identified SubCAs can be used for CBSA).

In order to accommodate this requirement, the SP must, after verifying that the certificate chains up to one of the installed Trust Anchors (i.e., the Root CA), verify that the Issuer of the certificate to be validated during the EAP-TLS session is the one (or one of the allowed ones) the SP wants to restrict the trust to by checking that the Issuer of the subscriber's certificate is in the allowed set.

4.4 UE Credentials Storage

In order to avoid requiring the user to enter their credentials every time the UE is required to authenticate to the network (or in case the credentials are directly provisioned to the device without requiring the subscriber to directly provide them), the UE may store the credentials in a permanent store. In particular, it shall be possible to store the credentials securely on the device since their compromise of would allow for an attacker to impersonate the subscriber.

Although it is out of scope of this document to provide indication to UE manufacturers about how to implement security mechanisms and controls to protect the subscriber's credentials, in case the UE is equipped with USIM, secure storage, secure elements (e.g., cryptography-capable hardware), or similar hardware-protected storage, it shall be possible for the UE to leverage these components to protect the subscriber credentials and relevant configuration options.

5 AAA servers

5.1 AAA servers for NHN Access Mode

When NHN Access Mode is selected, the NH-MME interacts (a) with the home SP's non-3GPP AAA Server via the Local AAA proxy server when authenticating subscribers via non-USIM based credentials, or (b) with the home SP's 3GPP AAA Server via the Local AAA proxy server when authenticating subscribers via USIM-based credentials. In particular, since NHN Access Mode uses EAP authentication (instead of EPS-AKA):

- EAP packets are transported between the *UE and the NH-MME* by using extended NAS messages as defined in MFA TS 24.301 [32] Section 8.2.32MF1 and Section 8.2.32MF2.
- EAP packets are transported between the *Local AAA proxy and the SP's non-3GPP AAA* via RADIUS (see RFC 3579 [33]) or Diameter (see RFC 6733 [34] and RFC 7075 [35]).

During the extended authentication procedures (i.e., in response to the *EAP-REQ/Identity* packet issued by the NH-MME), the UE indicates its home SP by providing its identity and home SP domain in a form of NAI defined in RFC 7542 [36]. The realm portion of the NAI is used by the local AAA Proxy to route the EAP packets to the appropriate non-3GPP AAA Server.

5.2 AAA servers for 3GPP-based Access Mode (non-EPS-AKA)

When 3GPP-based Access Mode (non-EPS-AKA) is selected, the MME' interacts with the local non-3GPP AAA Server directly when authenticating subscribers. In particular, since this access mode uses EAP authentication (instead of EPS-AKA):

- EAP packets are transported between the *UE and the MME'* by using extended NAS messages as defined in MFA TS 24.301 [32] Section 8.2.32MF1 and Section 8.2.32MF2.
- EAP packets are transported between the *MME' the SP's non-3GPP AAA* via RADIUS (see RFC 3579 [33]) or Diameter (see RFC 6733 [34] and RFC 7075 [35]).
- In case the SP's non-3GPP AAA server supports proxy functionality, EAP packets between the *SP's non-3GPP AAA server and other SPs' non-3GPP AAA* are transported via RADIUS (see RFC 3579 [33]) or Diameter (see RFC 6733 [34] and RFC 7075 [35]).

It is required that each non-USIM credential is associated with a unique IMSI value that the device must use in all authentication and identity procedures. In particular, during the attach, authentication, and identity request procedures, the UE shall provide its identity by using the IMSI associated with the subscriber's credentials used for authentication. For IMSIs that are associated with extended authentication and that are not served by the local SP, the SP's non-3GPP AAA server may be configured to provide Proxy functionality to route the EAP packets to the appropriate non-3GPP AAA Server.

6 Extended Subscribers Authentication

This section provides the specifications for using Extended Authentication (EAP-based) for NHN and 3GPP-based (non-EPS-AKA) Access Modes. Operators that decide to support Extended Authentication shall support EAP-TTLS and EAP-TLS mechanisms and may support additional ones. UEs that support Extended Authentication shall support EAP-TTLS and EAP-TLS and may support additional ones.

6.1 TLS Parameters Selection for EAP mechanisms

Both EAP-TTLS and EAP-TLS use the TLS protocol in order to establish a secure and authenticated communication channel between the UE and the AAA Server. SPs and UEs that support extended subscribers' authentication shall use the following settings for the TLS negotiation for both EAP-TTLS[2] and EAP-TLS mechanisms, in particular:

- The TLS endpoints shall support TLS version 1.2 [27] and may support TLS version 1.1 RFC 4346 [37] and/or 1.0 RFC 2246 [38] for compatibility reasons with UE implementations.
- For SPs that support TLSv1.2, the AAA server shall support the following ciphers and shall pick the first one (from the top of the following ordered list) that is supported by the UE during TLS negotiation:

  TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

  TLS_DHE_RSA_WITH_AES_256_GCM_SHA384

  TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256

  TLS_DHE_RSA_WITH_AES_128_CBC_SHA256

  For UEs that support TLSv1.1 or TLSv1.0, they shall support at least one of the above ciphers. SPs and UEs may support additional ciphers for backward compatibility.
- For SPs that support TLSv1.1 or TLSv1.0, the EAP-TTLS Server (AAA server) shall support the following ciphers and shall pick the first one (from the top of the following ordered list) that is supported by the UE during TLS negotiation:

  TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA

  TLS_DH_anon_WITH_3DES_EDE_CBC_SHA

  TLS_RSA_WITH_3DES_EDE_CBC_SHA

  For UEs that support TLSv1.1 or TLSv1.0, they shall support at least one of the above ciphers. SPs and UEs may support additional ciphers for backward compatibility.

6.2 Extended Subscribers Authentication via EAP-TTLS

During the first phase, the UE and the AAA server establish a secure communication channel by performing a TLS negotiation, while during the second phase the subscriber's credentials are exchanged and verified.

During the second phase of the EAP-TTLS protocol, the data exchanged between the UE and the AAA Server is sent by using Attribute-Value Pairs (AVPs) as described in Section 10 of RFC 5281 [8]: both parties are required

[2] As described in RFC 5281 Section 7.7.

to encode the information in a sequence of AVPs that must be processed by the TLS record layer for encryption to ensure that the identity and credentials information exchanged within the tunnel is kept secure.

6.2.1 Phase One Call Flow

The call flow for phase one of the initial attach procedure is depicted in Figure 6-1:

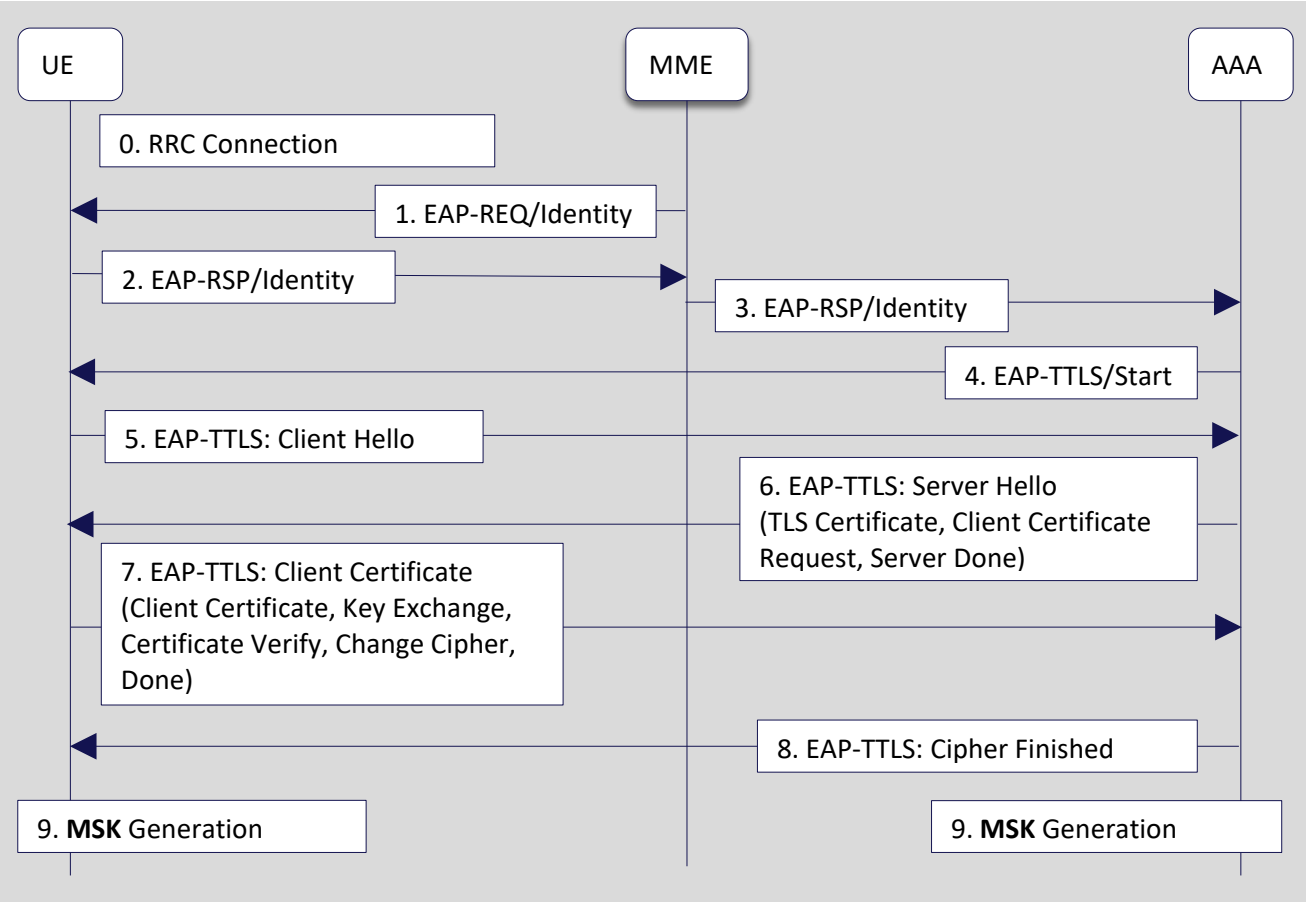

Figure 6-1: Phase-one call flow for initial attach with EAP-TTLS

The call flow for phase one is as follows:

1. After the initial RRC Connection Establishment the MME (when in NHN Access Mode) or the SP's MME' (when in 3GPP-based Access Mode [non-EPS-AKA]) initiates the EAP authentication procedure by sending the *EAP-REQ/Identity* packet to the UE.
2. The UE replies to the *EAP-REQ/Identity* with an *EAP-RSP/Identity* packet as described in Section 5.1 or Section 5.2 for NHN Access Mode and 3GPP-based Access Mode (non-EPS-AKA) respectively.
    - When in NHN Access Mode, the NH-MME uses the reported identity inside the *EAP-RSP/Identity* packet to route the packets to the appropriate AAA server via the local AAA Proxy.
    - When in 3GPP-base Access Mode (non-EPS-AKA), the SP's MME' uses the reported identity inside the *EAP-RSP/Identity* packet as the identity used in any subsequent procedure. Additionally, the MME' may decide to verify (depending on the SP's configured policy) that the

presented identity is the same as the one used in the Identity Response and may decide to end the authentication procedure (i.e., fail) when the two values do not match.

3. The *EAP-RSP/Identity* packet is forwarded to the appropriate AAA Server where the *EAP-TTLS* authentication method for the presented identity is selected. The considerations about the selection of the appropriate EAP method and support for EAP negotiation procedures are described in Section 4.1.1 and Section 4.1.3 respectively.
4. The AAA Server starts the selected EAP authentication mechanism by sending the *EAP-TTLS/Start* packet to the UE by setting the S (Start) bit in the packet as defined by RFC 5281 [8].
5. The UE starts the creation of the TLS tunnel by sending the *EAP-TTLS: Client Hello* packet to the AAA server with the initial parameters for TLS version selection and the supported list of ciphers.
6. The AAA server sends back the selected TLS version and selected cipher together with its own certificate (and certificate chain) in the *EAP-TTLS: Server Hello* packet. In addition, the server may include the request for an optional client certificate that may be used for device authentication during the establishment of the TLS channel.
    - This is a change in respect to Step 9 as defined in Section 5.12.3.4 of MFA MF.202 [6]. In particular, this step is modified in order to make client authentication during the TLS messages exchange optional as defined in RFC 5281 [8]. In case the UE has been provisioned with a device certificate, the UE shall include it in the Client Certificate response to the Server Hello message if the AAA Server included the request for client authentication.
7. The UE, after validating the server's certificate and certificate chain, replies to the *EAP-TTLS: Server Hello* packet by providing the selected cryptographic parameters (e.g., Client key exchange, Change Cipher, etc.) and, optionally, its own device certificate and the associated certificate chain.
    - If the UE has been provided with a unique device certificate, this shall be included in the *EAP-RSP/EAP-TTLS* that is transported to the MME over NAS and then forwarded to the home SP's non-3GPP AAA Server via the Local AAA proxy.
8. The AAA server proceeds with the validation of the client certificate (if provided by the UE) and sends the final packet to the UE with the indication of the successful TLS negotiation and final cipher selection.
    - The home SP's non-3GPP AAA Server shall not declare EAP failure if the UE does not provide a device certificate (i.e., in case an empty Client Certificate is sent in the response packet to the Server Hello) and shall not attempt to validate the client certificate in that case.
    - If a client certificate is provided by the UE, the AAA server shall attempt to validate it and shall fail for errors other than the fact that the certificate's chain does not anchor to any trusted Root CAs – i.e., the presented chain of certificates, together with the proof-of-possession of the private key associated with the presented client certificate, shall verify correctly up to the Root CA. Depending on the configured SP's authentication policy, the AAA server may fail the authentication procedure if the presented client certificate is not trusted.
9. The Master Session Key (MSK) and the Extended Master Session Key (EMSK) keying material is generated based on secret information developed during the TLS handshake between client and TTLS server as described in Section 8 of RFC 5281 [8].
    - The first 256 MSB (32 Bytes) of the MSK are used as the KASME to protect the communication layer at the end of the authentication procedure as described in Section 6.2.2 and in Section 5.12.4 of MFA MF.202 [6].

At this point, Phase One of the EAP-TTLS is successfully completed and the process continues with the inner EAP authentication mechanism for the subscriber's credentials.

6.2.2 Phase Two Call Flow

Phase Two is about authenticating the credentials associated with the identity reported by the UE in the initial *EAP-RSP/Identity* packet.

The complete call flow for phase two is depicted in Figure 6-2:

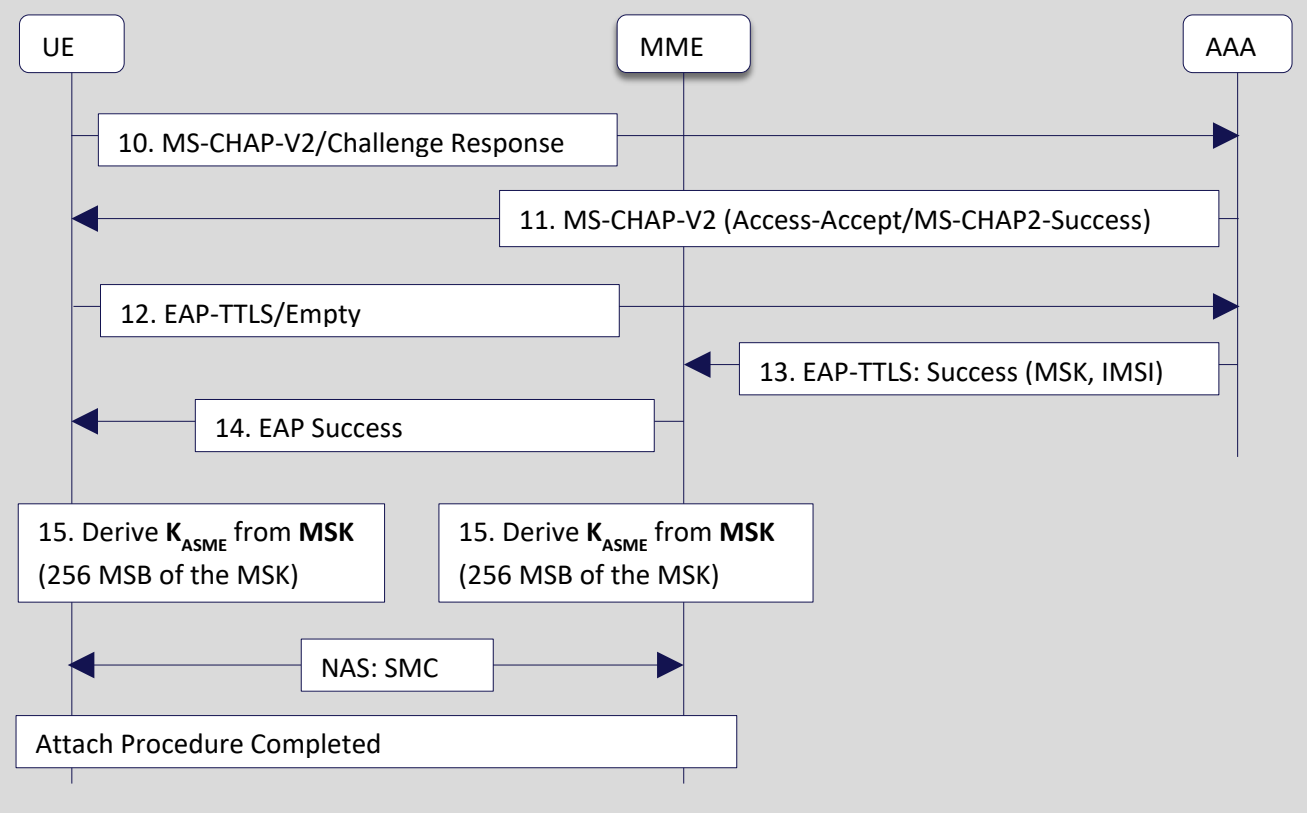

Figure 6-2: Phase-two call flow for initial attach with EAP-TTLS

The call flow for phase two is as follows:

10. The UE initiates the subscriber's credentials authentication by sending the initial *MS-CHAP-V2/Challenge Response* packet as described in Section 11.2.4 of RFC 5281 [8]. In particular, the packet sent to the AAA server includes the *User-Name*, the *MS-CHAP-Challenge*, and the *MS-CHAP2-Response* AVPs.
    - The *subscriber credential* shall be the *User-Name* and associated secret (password) used in the MS-CHAP-V2/Challenge Response.
    - The *MS-CHAP-Challenge* value is taken from the challenge material generated on the UE (17 Bytes).
    - The *MS-CHAP2-Response* consists of Ident (1 Byte from the challenge material), Flags (set to 0), Peer-Challenge (random value), and the Response (computed according to the MS-CHAP-V2 algorithm).

11. The AAA Server first verifies that the value of the *MS-CHAP2-Challenge AVP* and the value on the Ident in the client's *MS-CHAP2-Response AVP* are equal to the values generated as challenge material. If the authentication is successful, the AAA Server will respond with an *MS-CHAP2-Access-Accept AVP* with the *MS-CHAP2-Success AVP* (a 42-octet string that authenticates the AAA Server to the UE).
    - At this point, the authentication is not yet complete as the client must still accept the authentication response of the AAA Server.
12. The client authenticates the server based on the *MS-CHAP2-Success AVP* and the *MS-CHAP-Challenge AVP* generated in 10. If the authentication succeeds, the client sends an *EAP-TTLS/Empty* packet to the AAA server containing no data (that is, with a zero-length Data field).
13. Upon receipt of the empty *EAP-TTLS/Empty* packet from the UE, the AAA server considers the MS-CHAP-V2 authentication to have succeeded and issues an *EAP-TTLS/Success* packet to the MME which carries the MSK that was derived during phase one and the IMSI value associated with the credentials used for subscriber authentication
    - The *MPPE-Recv-Key* and *MPPE-Send-Key* attributes defined in RFC 2548 [40] are used to distribute the first 32 octets and second 32 octets of the MSK, respectively.
    - The *Extended-Type-1* attribute defined in RFC 6929 [41] is used to distribute the IMSI value associated to the credentials used for subscriber authentication.
14. The MME notifies the UE that the subscriber authentication was successful by sending an *EAP-TTLS/Success* packet to the UE.
    - The *EAP-TTLS/Success* packet exchanged between the MME and the UE does not contain the MSK nor the authenticated IMSI value as the communication between the two parties is not yet secured. Moreover, the UE has already derived the MSK from phase one and, therefore, the MSK does not need to be sent to the UE.
15. At this point the authentication is considered successful and both the UE and the MME derive the KASME by using the first 256 MSB (32 Bytes) of the MSK as described in Section 5.12.4 of MFA MF.202 [6].

6.2.3 EAP-TTLS Deployment for NHN Access Modes

The EAP authentication call flow shall follow the procedures described in the Section 6.2 with the following modifications:

- The MME in Section 6.2.1 and Section 6.2.2 is the NH-MME.
- The AAA Server in Section 6.2.1 and Section 6.2.2 is the SP's non-3GPP AAA Server.
- TLS cryptographic parameters shall follow the prescriptions in Section 6.1.

6.2.4 EAP-TTLS Deployment for 3GPP-based Access Mode (non-EPS-AKA)

The EAP authentication call flow shall follow the procedures described in the Section 6.3 with the following further specifications:

- The MME in Section 6.2.1 and Section 6.2.2 is the SP's MME.
- The AAA Server in Section 6.2.1 and Section 6.2.2 is the local SP's non-3GPP AAA Server.

- TLS cryptographic parameters shall follow the prescriptions in Section 6.1.

In order to prevent the UE from using subscriber credentials that are different from the identity provided in response to the first *EAP-REQ/Identity* packet from the MME', the use of *EAP-REQ/Identity* and *EAP-RSP/Identity* packets is prohibited after the successful completion of phase one.

When anonymous subscriber identities are used in the initial *EAP-RSP/Identity* packet from the UE as described in Section 2.1.4 of RFC 5216 [5], the AAA server must communicate the value of the IMSI associated with the credentials used for subscriber authentication to the MME' by including it in the *EAP-TLS/Success* packet via the *Extended-Type-1 AVP* as defined in Section 3.1 of RFC 6929 [41].

The MME' shall use the reported value for any subsequent operation involving the subscriber's identity, IMSI, and may decide to reject the connection in case the value reported by the AAA server does not match the value used in the initial Identity Request packet from Step 4 in Section 5.3.2.1 of 3GPP 23.401 [16].

6.3 Extended Subscribers Authentication via EAP-TLS

The complete Message Flow for EAP-TLS [8] is depicted in Figure 6-3:

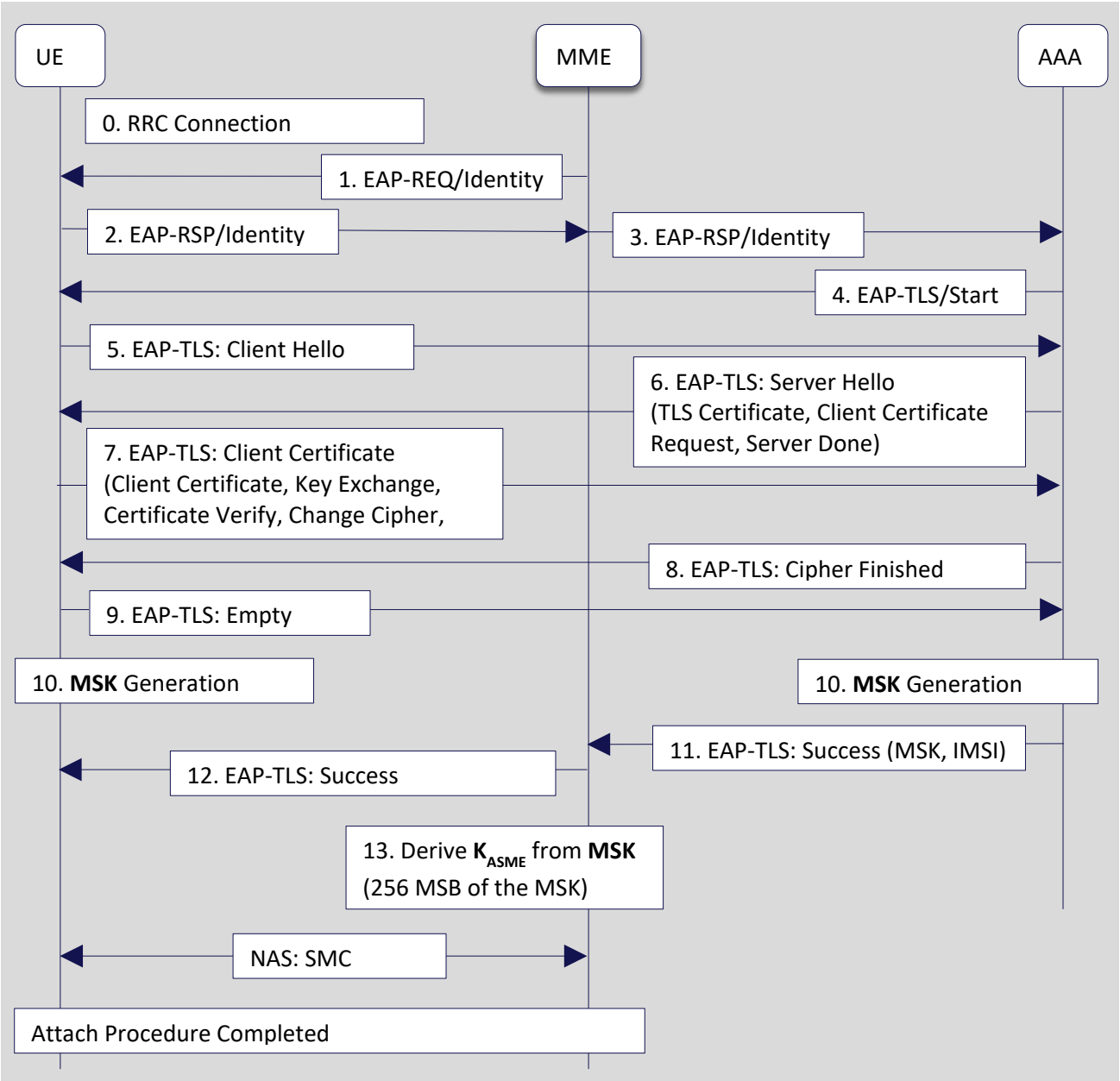

Figure 6-3: Complete call flow for initial attach with EAP-TLS

The EAP authentication shall follow the procedures described in Section 5.12.3.3 of MFA TS MF.202 [6] and support the modifications described in this section. In particular, the following messages shall be used:

1. After the initial RRC Connection Establishment the MME (when in NHN Access Mode) or the SP's MME' (when in 3GPP-based Access Mode [non-EPS-AKA]) initiates the EAP authentication procedure by sending the *EAP-REQ/Identity* packet to the UE.
2. The UE replies to the EAP-REQ/Identity with an *EAP-RSP/Identity* packet as described in Section 5.1 or Section 5.2 for NHN Access Mode and 3GPP-based Access Mode (non-EPS-AKA) respectively.
    - When in NHN Access Mode, the NH-MME uses the reported identity inside the *EAP-RSP/Identity* packet to route the packets to the appropriate AAA server via the local AAA Proxy.

- When in 3GPP-based Access Mode (non-EPS-AKA), the SP's MME' uses the reported identity inside the *EAP-RSP/Identity* packet as the identity used in any subsequent procedure. Additionally, the MME' may decide to verify (depending on the SP's configured policy) that the presented identity is the same as the one used in the Identity Response and may decide to end the authentication procedure (i.e., fail) when the two values do not match.

3. The *EAP-RSP/Identity* packet is forwarded to the appropriate AAA Server where the EAP-TTLS authentication method for the presented identity is selected. The considerations about the selection of the appropriate EAP method and support for EAP negotiation procedures are described in Section 4.1.1 and Section 4.1.3 respectively.
4. The AAA Server starts the selected EAP authentication mechanism by sending the *EAP-REQ/EAP-TLS*: Start packet to the UE by setting the S (Start) bit in the packet as defined by RFC 5216 [5].
5. The UE starts the creation of the TLS tunnel by sending the *EAP-RSP/EAP-TLS: Client Hello* packet to the AAA server with the initial parameters for TLS version selection and the supported list of ciphers.
6. The AAA server sends back the selected TLS version and selected cipher together with its own certificate (and certificate chain) in the EAP-REQ/*EAP-TLS: Server Hello* packet. In addition, the server includes the request for mandatory client certificate that will be used for subscriber authentication (mutual authentication).
7. The UE, after validating the server's certificate and certificate chain, replies to the EAP-RSP/*EAP-TLS: Server Hello* packet by providing the selected cryptographic parameters (e.g., Client key exchange, Change Cipher, etc.) together with the subscriber certificate and the associated certificate chain.
8. The AAA server proceeds with the validation of the client certificate and sends the final packet to the UE with the indication of the successful TLS negotiation and final cipher selection
    - The home SP's non-3GPP AAA Server must fail the EAP authentication if the UE does not provide a device certificate unless the server will request the certificate after the TLS finished message to protect the subscriber's identity as described in 6.3.1 and 6.3.2
    - The AAA server must attempt to validate the subscriber's certificate and shall fail in case the validation of the certificate shall fail for any reason
9. The UE sends an *EAP-TTLS/Empty* packet to the AAA server containing no data (that is, with a zero-length Data field) indicating the completion of the TLS negotiation.
10. On both the UE and the AAA server, the Master Session Key (MSK) and the Extended Master Session Key (EMSK) keying material is generated based on secret information developed during the TLS handshake between client and TTLS server as described in Section 8 of RFC 5281 [8].
11. Upon receipt of the empty *EAP-TLS/Empty* packet from the UE, the AAA server considers the EAP-TLS authentication to have succeeded and sends an *EAP-TLS/Success* packet to the MME which carries the MSK that was derived during phase one and the IMSI value associated with the credentials used for subscriber authentication.
    - The *MPPE-Recv-Key* and *MPPE-Send-Key* attributes defined in RFC 2548 [40] are used to distribute the first 32 octets and second 32 octets of the MSK, respectively.
    - The *Extended-Type-1* attribute defined in RFC 6929 [41] is used to distribute the IMSI value associated with the credentials used for subscriber authentication.

12. The MME notifies the UE that the subscriber authentication was successful by sending the *EAP-TTLS/Success* packet to the UE.
    - The *EAP-TTLS/Success* packet exchanged between the MME and the UE does not contain the MSK nor the authenticated IMSI value as the communication between the two parties is not yet secured. Moreover, the UE has already derived the MSK from phase one and, therefore, the MSK does not need to be sent to the UE.
13. On both the UE and the MME, the first 256 MSB (32 Bytes) of the MSK are utilized as the KASME that is used to protect the communication layer at the end of the authentication procedure as described in Section 5.12.4 of MFA MF.202 [6].

At this point, EAP-TLS is successfully completed. The EAP session can be successfully terminated or it can continue with the credentials management mechanism if supported by the operator.

6.3.1 EAP-TLS Deployment for NHN Access Modes

The EAP authentication call flow shall follow the procedures described in Section 6.3 and in Section 5.12.3.3 of MFA TS MF.202 with the following modifications:

- The MME in Section 6.3 is the NH-MME.
- The AAA Server in Section 6.3 is the local SP's non-3GPP AAA Server.
- TLS cryptographic parameters shall follow the prescriptions in Section 6.1.

As discussed in Section 5.12.3.3 of MFA MF.202 [6], the user identity may be protected as described in Section 2.1.4 of RFC 5216 [5]. In this case the actual identity is retrieved from the Client (subscriber) certificate that shall be sent after Step 8 (TLS Finished), i.e. under the TLS protection. In particular, the SP's non-3GPP AAA server continues the TLS handshake requesting the Client certificate again after Step 8 and performs the client certificate validation before sending the final *EAP-TLS/Success* packet.

In this case, the AAA may include the value of the IMSI associated with the credentials used for subscriber authentication in the *EAP-TLS/Success* packet via the *Extended-Type-1* AVP as defined in Section 3.1 of RFC 6929 [41].

6.3.2 EAP-TLS Deployment for 3GPP-based Access Mode (non-EPS-AKA)

The authentication flow is the same as depicted in 3GPP 33.401 Section 5.3.2 [17] up to, but excluding, 5a. In particular the User Authentication Request and the User Authentication Response messages are replaced by the call flow described in Section 6.3.

The EAP authentication call flow shall follow the procedures described in the Section 6.3 and in Section 5.12.3.3 of MFA TS MF.202 [6] with the following modifications:

- The MME in Section 6.3 is the SP's MME'.
- The AAA Server in Section 6.3 is the local SP's non-3GPP AAA Server.
- TLS supported parameters shall follow the prescriptions in Section 6.1.

In this Access Mode, the use of anonymous identities in the *EAP-RSP/Identity* packet, from the UE, in Step 2 (as described in Section 2.1.4 of RFC 5216 [5]) may be supported by the SP's MME' during the initial attach procedure. In this case, the AAA server must communicate the value of the IMSI associated with the

credentials used for subscriber authentication to the MME' by including it in the *EAP-TLS/Success* packet via the *Extended-Type-1* attribute as defined in Section 3.1 of RFC 6929 [41].

When real identities are used in the *EAP-RSP/Identity* packet from the UE, in order to make sure that the identity reported during the attach procedure is the actual one used during the authentication process, the AAA server must verify that the identity used in the *EAP-RSP/Identity* packet is the same as the one that is presented in the client certificate used for subscriber authentication. Also in this case, the AAA shall include the value of the IMSI associated with the credentials used for subscriber authentication in the *EAP-TLS/Success* packet by using the *Extended-Type-1* attribute as defined in Section 3.1 of RFC 6929 [41].

7 Subscribers Credentials Management

7.1 EAP-based Credentials Management Overview

The provisioning and management of UE credentials is a hard problem to solve for vendors and operators alike that often opt for long-lived credentials that cannot be easily revoked, replaced, or simply renewed.

CBRS networks address credentials management by leveraging the EAP [26] protocol to support provisioning, renewal, and removal functionality via EAP-CREDS [42]. In particular, the AAA server, after a successful EAP-TLS or EAP-TTLS session may continue the EAP session by sending a new EAP-Request/EAP-CREDS to the UE instead of sending the final EAP success/fail message. After that, the EAP-CREDS message flow, which is logically subdivided into three different phases (i.e., Initialization, Provisioning, and Validation), follows. During the Initialization phase, the AAA server and the UE exchange the list of credentials installed on the UE and the UE's capabilities (e.g., what encodings are supported, cryptographic algorithms, storage capabilities, etc.). After that, if all parameters required by the AAA server are supported, the server starts the Provisioning phase where the server can manage (renew, provision, or delete) the UE's credentials. If the AAA server does not need to perform credentials management, the server can terminate the EAP-CREDS section at the end of the Initialization phase.

Note: If the AAA server allows for registration of new UE's credentials, it may allow the use of EAP-TLS and/or EAP-TTLS without performing client authentication. In this case, the use of a token-based (or other bootstrap mechanism) is supported during the Initialization phase to provide registration information for the UE (e.g., a one-time token or symmetric secret) to get or register the access credentials for the UE.

Although EAP-CREDS supports the encapsulation of different standard protocols (e.g., CMP, CMC, etc.) and vendor-specific ones (See Appendix B), UEs and AAA servers should support, at minimum, the Simple Provisioning Protocol (SPP) that is integrated into EAP-CREDS and may support others. When the SPP protocol does not provide the support for specific features required by an operator, a vendor-specific protocol can be implemented by UEs and AAA servers to address specific use-cases.

Note: Adding support for new provisioning features can be easily implemented via software updates when access to the EAP layer is provided to the UE's credential subsystem.

Differently from other provisioning protocols where the client must know when a credential is to be renewed/managed to start the provisioning protocol, EAP-CREDS uses a server-driven approach. That means that the AAA server dictates the operations to be carried out by the UE (See Section 7.1.2) thus transferring most of the credentials management complexity to the AAA server (i.e., simplifies UE's logic).

Another important characteristic of EAP-CREDS is its use of Type-Length-Value (TLV) structures to carry different types of data. Throughout this document, we import the EAP-CREDS notation for TLVs as described in Table 7-1:

Table 7-1: EAP-CREDS notation

Symbol	Example	Usage
{ }	{TLV1}	Curly Brackets are used to indicate a set
[ ]	{[TLV2]}	Square Brackets are used to indicate that a field is optional

Symbol	Example	Usage
( )	{TLV1(=V)}	Round Squares are used to specify a value
+	{TLV_2+}	The Plus character indicates that one or more instances are allowed

The rest of this section provides a detailed description of EAP-CREDS method deployment in CBRS networks by including the detailed SPP message flows and parameters specifications for provisioning certificate-based and non-certificate-based credentials.

7.1.1 Security Requirements for Outer EAP Tunnel

EAP-CREDS requires that a secure association is in place between the UE and the AAA server in order to provide authentication and confidentiality of the messages exchanged via EAP-CREDS. Specifically, EAP-CREDS assumes that an appropriately encrypted and authenticated channel has been established to prevent the possibility to leak information or to allow man-in-the-middle attacks.

In CBRS networks, the use of EAP-CREDS must follow the use of one of the supported extended authentication methods (e.g., EAP-TLS or EAP-TTLS).

For networks that allow new UE registrations, the outer authentication layer (i.e., EAP-TLS or EAP-TTLS) must be configured to allow for optional client authentication to establish the secure channel before the registration/authorization information (i.e., a device certificate, a one-time token, a shared secret, etc.) is exchanged between the two parties.

7.1.2 EAP-CREDS' Simple Provisioning Protocol (SPP)

EAP-CREDS implements the Simple Provisioning Protocol (SPP) which comprises of a series of messages that enable the management not only of certificates, but also of other types of credentials like username/password pairs, asymmetric keys, and symmetric keys. Since CBRS networks support certificates and username/password types of extended credentials, EAP-CREDS focuses on these two specific authentication use cases.

Note: Since EAP-CREDS provides support for symmetric secrets management, it is possible for AAA servers to also manage AKA-based credentials when the UE's EAP layer is granted direct access to the master secret.

When no description of UE-installed credentials is provided by the UE in its Initialization phase message (see Step 2 in Section 7.3), the AAA server knows that the UE does not have valid credentials it wants to use to access the Network (or they have not been installed, yet). In this case, EAP-CREDS supports the use of Tokens to start the registration process. The type, format, or encoding of the Token is treated as a black-box field (i.e., EAP-CREDS does not require a specific format) thus allowing operators to integrate EAP-CREDS with their authorization system.

Note: During the Initialization phase, although the UE may include the token data in its Initialization message to provide the needed authorization to register a new set of credentials (see Section 7.3), AAA servers can still reject the operation when configured to deny the registration of new UEs or new credentials.

In the case where an authorization token is used, different usage patterns are supported that allow not only for bearer or one-time tokens, but also for tokens that support verifiable proof-of-possession data (i.e., the generation of challenge-response data based on a secret or a private key that is referenced in the token).

Note: Other methods to provide authorization information might be provided by the selected provisioning protocol: in this case, the AAA server may still enable the registration of new credentials when no authorization data is provided in the Initialization message from the UE and delegate the validation of the authorization data to the Provisioning phase.

7.2 Required Support for Provisioning Parameters

EAP-CREDS provides the possibility for the UE and the AAA server to specify different types of parameters (and values) that identify the supported credentials types, algorithms, datatypes, encoding types, etc. This subsection provides the required list of parameters and values for UEs and AAA servers.

7.2.1 Provisioning Protocols Values Requirements

UEs and AAA servers must support the SPP provisioning protocol (1) and may support others.

7.2.2 Credentials Types Values Requirements

UEs and AAA servers must support X.509 Certificates (1) and Username and Password (4) credentials types and may support others.

7.2.3 Credentials Algorithms Values Requirements

UEs and AAA servers must support None (0), RSA (1) and ECDSA (2) credentials algorithms and may support others.

7.2.4 Credentials Datatypes Values Requirements

UEs and AAA servers must support None (0), PKCS#8 (1), and PKCS#10 (2) credentials datatypes and may support others.

7.3 The Initialization Phase

EAP-CREDS starts with the Initialization phase. During this phase, the UE and the AAA Server exchange the details about the credentials installed on the UE, its internal status (e.g., the storage description), and its capabilities. The AAA server utilizes this information during the Provisioning phase of EAP-CREDS to identify which credentials need to be renewed, removed, or if new credentials need to be registered.

Note: Operators can decide to switch to a different type when renewing the UE's credentials. In this case, if the new type of credentials is not supported by the UE, the UE will respond with the appropriate error code, during the Provisioning phase, to the AAA Server's message where the description of the type of credentials is provided.

The complete message flow for the Initialization phase is provided in Figure 7-1.

The call flow for the Initialization phase comprises two messages:

1. **The AAA Server sends EAP-Request/EAP-CREDS(Type=Init).** After the establishment of the outer mechanism (e.g., EAP-TLS, EAP-TTLS, etc.), the server may start a credentials management session. In order to do that, instead of sending the final *Success* packet, the AAA Server sends an EAP-Request/EAP-CREDS(Type=Init) message to the UE with the 'S' bit set to '1' and the value of the phase field set to '1'

(thus indicating the beginning of the Initialization phase). The Server may optionally include one or more ('Version') TLVs to indicate the supported/enabled versions of EAP-CREDS. At minimum, the AAA server must support version one (1) of EAP-CREDS and may support others. Ultimately, the server may also include a ('Challenge') TLV when the registration of new UEs is enabled and it requires the use of a server-generated challenge to generate the authorization token data on the UE.

2. **The UE sends EAP-Response/EAP-CREDS(Type=Init).** The UE sends back its own Initialization message that carries one ('Version') TLV to indicate the selected version of EAP-CREDS (i.e. from the list provided by the server) (optional). If the client does not include the ('Version') TLV, the Server MUST use the most recent supported version of EAP-CREDS. The UE may also include one or more ('Protocol') TLVs to indicate the list of supported provisioning protocols, followed by one ('Credentials-Info') TLV for each of the installed credentials, i.e. if multiple credentials are configured on the UE for the current Network, then the UE must include one ('Credentials-Info') TLV for each of them.

Note: When no credentials are available on the UE, the UE may provide authorization information to the AAA server by including the proper TLVs. See Section 8 for a detailed description.

The list of supported Encodings and Formats is provided in one or more ('Supported-Encodings') and ('Supported-Formats') TLVs. The UE may optionally include the ('Storage-Status') TLV to provide the AAA Server with information about the its credentials storage.

Moreover, the UE may optionally add the ('Network-Usage') TLV to provide the Server with the indication of which network resources are needed by the Peer and what is its intended utilization pattern(s).

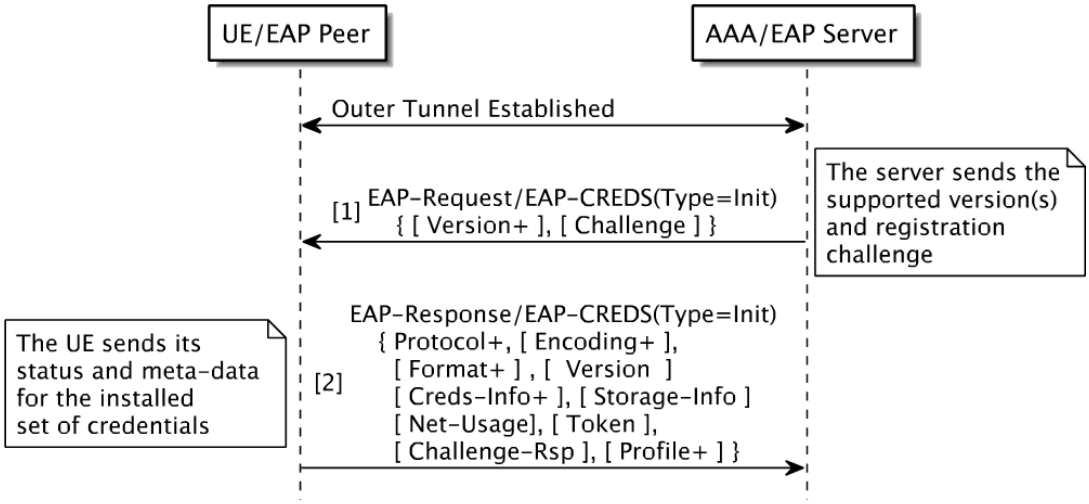

Figure 7-1: Complete Message flow for EAP-CREDS Initialization Phase

When the AAA server receives the UE's Initialization message, it checks for the supported protocols, version, and parameters and, if the server detects an error, it can proceed down two different paths. In particular, the server may (a) send a non-recoverable error message to the peer, notify the outer (tunneling) layer, and terminate the EAP-CREDS session, or (b) start phase one again by sending a new EAP-Request/EAP-CREDS(Type=Initialization) message again. The message must include the Error TLV that provides the UE with the reason the initial response was not acceptable. The AAA server and the UE can repeat phase one until they reach an agreement, or the session is terminated by the AAA server.

7.4 Non-Certificate-Based Credentials Management

CBRS-A allows for UE to be authenticated not only via SIM-based and Certificate-based credentials, but also via username and password combinations. This section describes how to manage such credentials via EAP-CREDS.

7.4.1 Provisioning Server-Side Generated secrets

When username-and-password based credentials are generated by the server, the complete message flow for EAP-CREDS/SPP is depicted in Figure 7-2.

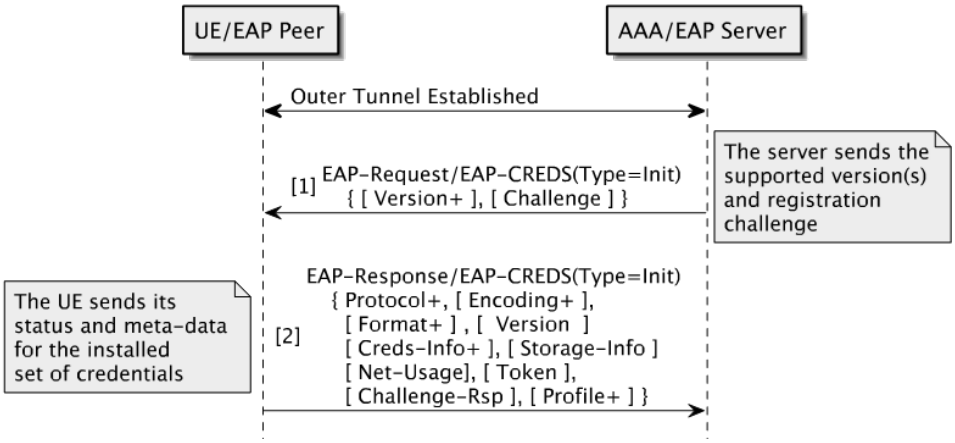

Figure 7-2: Server generated Username and Password provisioning message flow

The call flow for Username and Password provisioning with server-side generation is as follows:

1. ***The AAA Server sends the EAP-Request/EAP-CREDS(Type=Provisioning) packet.*** The AAA server must include the following TLVs:
   - The ('Protocol') TLV with the value set to the SPP protocol (0x1).
   - The ('Action') TLV with the value set to Registration (0x0) or Renewal (0x1).
   - The ('Credentials-Info') TLV where the fields must identify the specific credential to be Renewed. In particular, for username and password type of credentials, the following values should be used:
     - The *CredType* field must carry a value of (3) to indicate the use of username and password credentials.
     - The *ProtoID* field must carry a value of (0) to indicate that there is no special algorithm associated with the credentials.
     - The *IssuedOn* field must carry the issuance (or creation) timestamp of the credential.
     - The *ExpiresOn* field must carry the expiration date in GMT format for the credentials in a null-terminated string. If no expiration is set for the credential, the value of field shall contain only the null-termination octet (0x0).
     - The *CredentialsLength* field must be set to the size (in bytes) of the password.
     - The *CredIDValue* field must be set to the username value.
   - The ('Provisioning-Params') TLV where:

    - The *Min Length* field must be set to the minimum acceptable size (in bytes) for passwords.
    - The *Max Length* field must be set to the maximum acceptable size (in bytes) for passwords.
    - The *Algorithm* field must be set to (0).
    - The *Flags* bit field must have the following configuration:
        - Bit 0 – Set,
        - Bit 1 ... 7 – Not Set.
    - The *ObjectIdentifier* field must be omitted (i.e., the TLV ends with the flags bitfield).
  - The ('Provisioning-Data') TLV must carry one (and only one) ('Credentials-Data') TLV where:
    - The *CredType* field must be set to (3) for username and password.
    - The *Format* field must be set to (0) for binary format.
    - The *Encoding* field must be set to: three (3) if the value of the credential sent to the UE is Base64 encoded, six (6) if the value of the credential sent to the UE is ASCII encoded, or seven (7) if the value of the credential sent to the UE is UTF-8 encoded. In case no specific encoding is to be used when parsing the credential, the value of zero (0) must be used.
    - The *Value* field must be set to the value of the credentials encoded according to the format specified in the Encoding field value.

  Optionally, the AAA server may include the following TLV:

  - The ('Profile') TLV with the value set to the profile identifier (if any) associated with username and password type of credentials with specific parameters (e.g., the size of the password). This TLV can be used to allow one to specify, for example, different configurations (or profiles) for a type of credential (e.g., "Cellular Phone Password Profile" or "IoT Password Profile").

2. ***The UE sends the EAP-Response/EAP-CREDS(Type=Provisioning) packet.*** The UE must include the following TLVs:
  - The ('Protocol') TLV with the value set to the SPP protocol (0x1).
  - The ('Provisioning-Data') TLV with:
    - One ('Credentials-Info') TLV where the values must identify the provisioned username and password, if and only if the credentials pushed by the AAA server to the UE are supported by the UE, or
    - One ('Error') TLV with the appropriate failure code if the UE detects any error during the installation of the credentials – no ('Credentials-Info') TLV must be present in this case.

If no error is reported by the UE, the provisioning of the new username and password credential is completed successfully. The server can now start the Provisioning phase again for a different set of credentials (if any exist and need to be managed), move to the Validation phase, or successfully terminate the EAP-CREDS session by issuing a *Success* packet.

7.4.2 Provisioning Co-Generated secrets

When username-and-password based credentials are generated by the server, the complete message flow for EAP-CREDS/SPP is depicted in Figure 7-3.

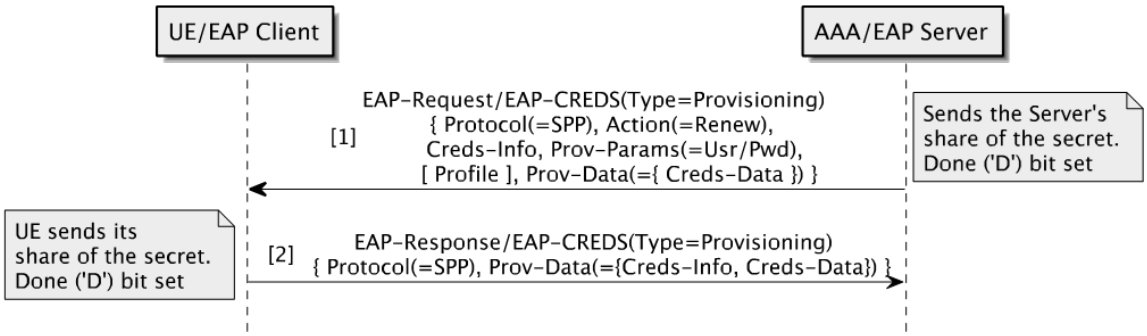

Figure 7-3: Co-generated Username and Password provisioning message flow

The call flow for Username and Password provisioning with co-generation is as follows:

1. ***The AAA Server sends the EAP-Request/EAP-CREDS(Type=Provisioning) packet.*** The AAA server must include the same TLVs as in the first message as specified in Section 7.4.1 with the following changes:
   - In the ('Provisioning-Params') TLV where:
     - The *Flags* bit field must have the following configuration:
       - Bit 0 – Set,
       - Bit 1 – Set,
       - Bit 2 ... 7 – Not Set.
2. ***The UE sends the EAP-Response/EAP-CREDS(Type=Provisioning) packet.*** The UE must include the same TLVs as in the second message as specified in Section 7.4.1 with the following changes:
   - Add a ('Credentials-Data') TLV encoded in the ('Provisioning-Data') TLV with the following setting:
     - The *CredType* field must be set to (3) for username and password.
     - The *Format* field must be set to (0) for binary format.
     - The *Encoding* field must be set to: three (3) if the value of the credential is Base64 encoded, six (6) if the value of the credential is ASCII encoded, or seven (7) if the value of the credential is UTF-8 encoded. In case no specific encoding is to be used when parsing the credential, the value of zero (0) must be used.
     - The *Value* field must be set to the UE-generated part of the credentials encoded according to the format specified in the Encoding field value.

The final password to be used for future authentication is generated by concatenating the two shares of the password: first the one from the Server, then the one from the UE.

If no error is reported by the UE or the Server, the provisioning of the new username and password credential is completed successfully. The server can now start the Provisioning phase again for a different set of credentials (if any exist and need to be managed), move to the Validation phase, or successfully terminate the EAP-CREDS session by issuing a *Success* packet.

7.4.3 Registering Client-Side Generated secrets

When username-and-password based credentials are generated by the peer, the complete message flow for EAP-CREDS/SPP is depicted in Figure 7-4.

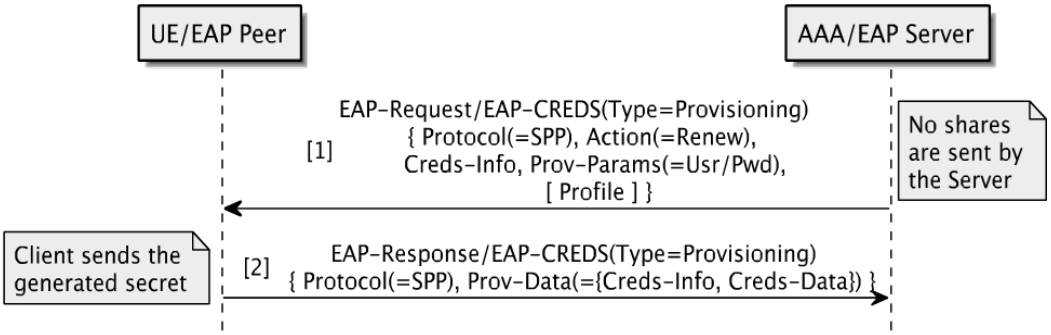

Figure 7-4: Client-Side Username and Password provisioning message flow

The call flow for Username and Password provisioning with client-side generation is as follows:

1. ***The AAA Server sends the EAP-Request/EAP-CREDS(Type=Provisioning) packet.*** The AAA server must include the same TLVs as in the first message as specified in Section 7.4.1 with the following changes:
   - In the ('Provisioning-Params') TLV where:
     - The *Flags* bit field must have the following configuration:
       - Bit 0 – Not Set,
       - Bit 1 – Set,
       - Bit 2 ... 7 – Not Set.
   - No ('Provisioning-Data') TLV must be present in the message as there is no part of the secret that is generated on the Server.
2. ***The UE sends the EAP-Response/EAP-CREDS(Type=Provisioning) packet.*** The UE must include the same TLVs as in the second message as specified in Section 7.4.1 with the following changes:
   - The ('Credentials-Data') TLV encoded in the ('Provisioning-Data') TLV with the following setting:
     - The *CredType* field must be set to (3) for username and password.
     - The *Format* field must be set to (0) for binary encoding.
     - The *Encoding* field must be set to: three (3) if the value of the credential is Base64 encoded, six (6) if the value of the credential is ASCII encoded, or seven (7) if the value of the credential is UTF-8 encoded. In case no specific encoding is to be used when parsing the credential, the value of zero (0) must be used.
     - The *Value* field must be set to the UE-generated password encoded according to the format specified in the Encoding field value.

In this case, the password provided by the UE is the one used for the authentication. It is advisable to perform a quality check of the password to ensure it conforms to the risk profile selected for the access network.

If no error is reported by the UE or the Server, the provisioning of the new username and password credential is completed successfully. The server can now start the Provisioning phase again for a different set of credentials (if any exist and need to be managed), move to the Validation phase, or successfully terminate the EAP-CREDS session by issuing a *Success* packet.

7.4.4 Security Considerations

One important set of security considerations is related to the symmetric secrets' generation. In particular, when Client-Side or Server-Side generation happens, one party is in sole control over the quality of the secret that is being generated – before accepting/storing the new secret, the receiving party should check it for randomness and reject secrets that do not align with the risk profile for the specific access network.

Because of these considerations, the safest secret generation is the co-generation mechanism followed by the server-side one (generally more secure than client-generated ones).

7.5 Certificate Based Extended Credentials Management

One of the most difficult tasks when it comes to certificate management is renewal. CBRS-A allows for certificate-based credentials to be managed via EAP-CREDS. This section describes how to use EAP-CREDS and the SPP protocol to provision such credentials.

7.5.1 Provisioning Server-Side Generated Certificates

When certificate-based credentials are generated by the server, the complete message flow for EAP-CREDS/SPP is depicted in Figure 7-5.

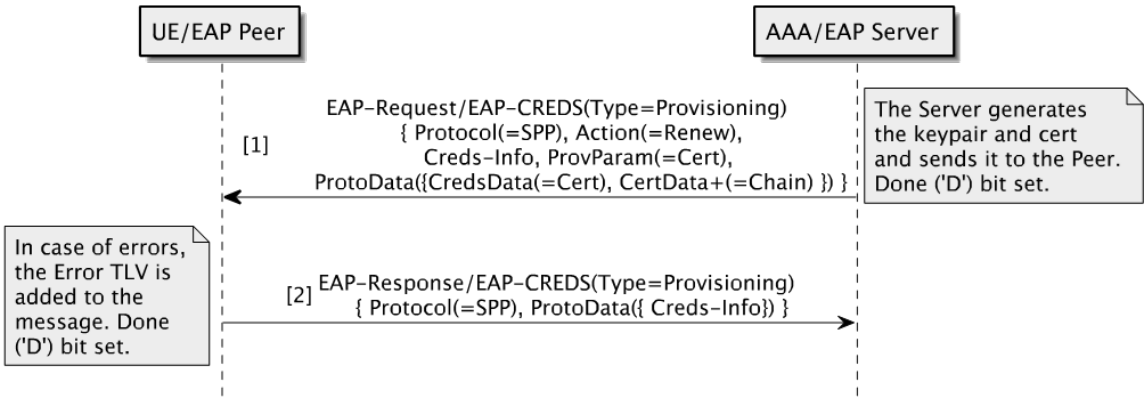

Figure 7-5: Server-Side generated Certificate provisioning message flow

The call flow for certificate provisioning with server-side generation is as follows:

1. ***The AAA Server sends the EAP-Request/EAP-CREDS(Type=Provisioning) packet.*** The AAA server must include the following TLVs:
    - The ('Protocol') TLV with the value set to the SPP protocol (0x1).
    - The ('Action') TLV with the value set to Registration (0x0) or Renewal (0x1).
    - The ('Credentials-Info') TLV where the fields must identify the specific credential to be Renewed. In particular, for certificate based credentials, the following values should be used:
        - The *CredType* field must carry a value of (0) to indicate that use of X.509 certificates.
        - The *ProtoID* field must carry a value of (1) to indicate that these credentials have been retrieved via SPP.
        - The *IssuedOn* field must carry the issuance (or creation) timestamp of the credential.

    - The *ExpiresOn* field must carry the expiration date in GMT format for the credentials in a null-terminated string. If no expiration is set for the credential, the value of field shall contain only the null-termination octet (0x0).
    - The *CredentialsLength* field must be set to the size (in bytes) of the password.
    - The *CredIDValue* field must be set to a uniquely identifying data (i.e., the SHA-256 of the DER representation of the certificate or a more human-friendly null-terminated string).
  - The ('Provisioning-Params') TLV where:
    - The *Min Length* field must be set to (0) and ignored on the client.
    - The *Max Length* field must be set to (0) and ignored by the client.
    - The *Algorithm* field must be set to (1) if RSA keys are used or to (2) if ECDSA keys are used.
    - The *Flags* bit field must have the following configuration:
      - Bit 0 – Set,
      - Bit 1 ... 7 – Not Set.
    - The *ObjectIdentifier* field must be omitted (i.e., the TLV ends with the flags bitfield).
  - The ('Provisioning-Data') TLV must carry one (and only one) ('Credentials-Data') TLV where:
    - The *CredType* field must be set to (0) to indicate an X.509 certificate and key.
    - The *Format* field must be set to (3) to indicate the use of a PKCS#12 data structure (DER encoded) to deliver the credentials. In particular, the PKCS#12 must contain the following bags:
      - The Private Key bag,
      - Intermediate CAs bags (one for each intermediate CA),
      - The Root CA bag.
    - The *Encoding* field must be set to (1) for DER encoding.
    - The *Value* field must be set to the value of the PKCS#12 (DER encoded) that contains the new set of credentials.

  Optionally, the AAA server may include the following TLV:

  - The ('Profile') TLV with the value set to the profile identifier (if any) associated with the current type of credentials.

2. ***The UE sends the EAP-Response/EAP-CREDS(Type=Provisioning) packet.*** The UE must include the following TLVs:
   - The ('Protocol') TLV with the value set to the SPP protocol (0x1)
   - The ('Provisioning-Data') TLV with:
     - One ('Credentials-Info') TLV where the values must identify the provisioned X.509 credentials, if and only if the credentials pushed by the AAA server to the UE are supported by the UE (and were correctly installed), or
     - One ('Error') TLV with the appropriate failure code if the UE detects any error during the installation of the credentials – no ('Credentials-Info') TLV must be present in this case.

If no error is reported by the UE, the provisioning of the new certificate is completed successfully. The server can now start the Provisioning phase again for a different set of credentials (if any exist and need to be managed), move to the Validation phase, or successfully terminate the EAP-CREDS session by issuing a *Success* packet.

7.5.2 Provisioning Co-Generated Certificates

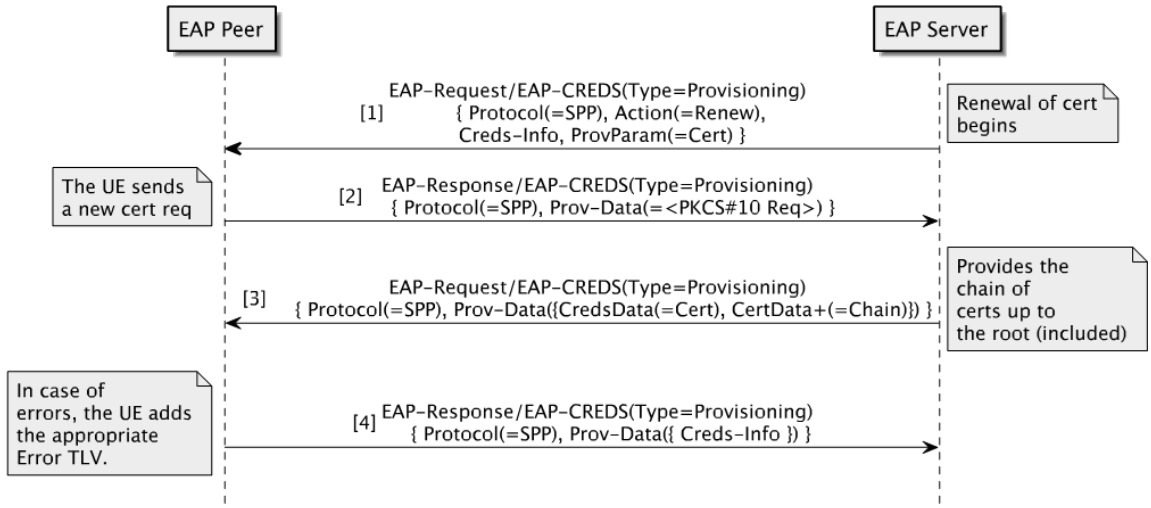

Figure 7-6: Co-generated Certificate provisioning message flow

The call flow for certificate provisioning with server and client side co-generation is as follows:

1. ***The AAA Server sends the EAP-Request/EAP-CREDS(Type=Provisioning) packet.*** The AAA server must include the following TLVs:
   - The ('Protocol') TLV with the value set to the SPP protocol (0x1).
   - The ('Action') TLV with the value set to Registration (0x0) or Renewal (0x1).
   - The ('Credentials-Info') TLV where the fields must identify the specific credential to be Registered or Renewed. In particular, for certificate based credentials, the following values should be used:
     - The *CredType* field must carry a value of (0) to indicate that use of X.509 certificates.
     - The *ProtoID* field must carry a value of (1) to indicate that these credentials have been retrieved via SPP.
     - The *IssuedOn* field must not be set (i.e., set to zero (0x0)).
     - The *ExpiresOn* field must carry the expiration date in GMT format for the credentials in a null-terminated string. If no expiration is set for the credential, the value of field shall contain only the null-termination octet (0x0).
     - The *CredentialsLength* field must be set to zero (0x0).
     - The *CredIDValue* field must be present in case of renewal of credentials while it should be omitted in case of registration of new credentials.
   - The ('Provisioning-Params') TLV where:

- The *Min Length* field must be set to the minimum value for the key size (in bits) that is supported by the network (i.e., minimum level of security) – if applicable – or zero (0) to indicate no constraints. For example, for RSA keys this field can be set to 2048 while for ECDSA keys this field shall be set to zero (0) since the objectIdentifier (the curve) determines the size of the key.
- The *Max Length* field must be set to the maximum value for the key size (in bits) that is supported by the network (i.e., maximum key size) – if applicable – or zero (0) to indicate no constraints. For example, for RSA keys this field can be set to 4096 while for ECDSA keys this field shall be set to zero (0) since the objectIdentifier (the curve) determines the size of the key.
- The *Algorithm* field must be set to (1) if RSA keys are used or to (2) if ECDSA keys are used.
- The *Flags* bit field must have the following configuration:
  - Bit 0 – Set (1),
  - Bit 1 – Set (1),
  - Bit 2 ... 7 – Not Set (0).
- The *ObjectIdentifier* field must be omitted for RSA keys (i.e., the TLV ends with the flags bitfield). For ECDSA keys, the ObjectIdentifier field must carry the DER representation of the field (or curve) to be used for key generation.

- (Optional) The ('Profile') TLV may be added by the AAA server with the value set to the profile identifier (if any) associated with the current type of credentials. This value can be used in subsequent sessions to request credentials with the same profile.

2. ***The UE sends the EAP-Response/EAP-CREDS(Type=Provisioning) packet.*** The UE must include the following TLVs:
   - The ('Protocol') TLV with the value set to the SPP protocol (0x1).
   - The ('Provisioning-Data') TLV with:
     - One ('Certificate-Request') TLV where:
       - The *Encoding* field must be set to (1) for DER encoding.
       - The *Format* field must be set to (2) for PKCS#10 format.
       - The *Value* field must be set to the DER encoding of the PKCS#10 request.
     - One ('Error') TLV with the appropriate failure code if the UE detects any error during the installation of the credentials – no ('Credentials-Info') TLV must be present in this case.
3. ***The AAA Server sends the EAP-Request/EAP-CREDS(Type=Provisioning) packet.*** The AAA server must include the following TLVs:
   - The ('Protocol') TLV with the value set to the SPP protocol (0x1).
   - The ('Provisioning-Data') TLV with:
     - One ('Credentials-Data') TLV where:
       - The *CredType* field must be set to (0) for X.509 certificate type.
       - The *Format* field must be set to (0) for binary format.
       - The *Encoding* field must be set to (1) for DER encoding.

        - The *Value* field must be set to the value of the DER representation of the issued certificate.
    - One ('Certificate-Data') TLV for each certificate in the chain up to the root (included). In particular, for each ('Certificate-Data') TLV:
        - The *Flags* field must be set according to the type of certificate. In particular, for a trusted root CA the value (0x07) shall be used, while for an intermediate CA certificates, the value shall be set to (0x02).
        - The *Encoding* field shall be set to (0x01) for DER encoding.
        - The *Value* field shall be set to the DER representation of the certificate.
    - One ('Error') TLV with the appropriate failure code if the Server detects any error during the issuing of the credentials – no ('Credentials-Info') TLV must be present in this case.

4. ***The UE sends the EAP-Response/EAP-CREDS(Type=Provisioning) packet.*** The UE must include the following TLVs:
    - The ('Protocol') TLV with the value set to the SPP protocol (0x1).
    - The ('Provisioning-Data') TLV with:
        - One ('Credentials-Info') TLV where the values must identify the provisioned X.509 credentials, if and only if the credentials pushed by the AAA server to the UE are supported by the UE (and were correctly installed), or
        - One ('Error') TLV with the appropriate failure code if the UE detects any error during the installation of the credentials – no ('Credentials-Info') TLV must be present in this case.

If no error is reported by the Server, the provisioning of the new certificate is completed successfully. The server can now start the Provisioning phase again for a different set of credentials (if any exist and need to be managed), move to the Validation phase, or successfully terminate the EAP-CREDS session by issuing a *Success* packet.

7.5.3 Registering Self-Signed (UE-generated) or Device Certificates

When devices have already been provisioned with credentials that can be used directly on the network to authenticate the device, CBRS-A allows to directly register those credentials in the system. In particular, when a certificate is used for this purpose (either self-signed or properly chained), the UE can share the certificate with the server for registration.

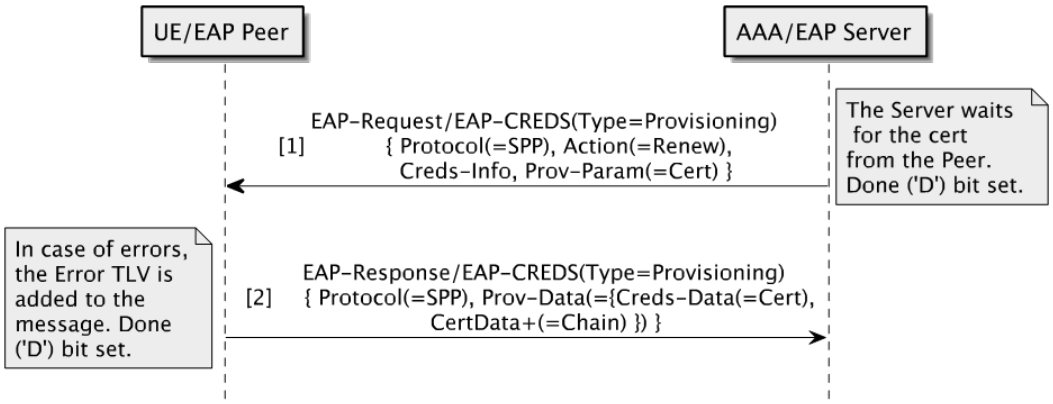

Figure 7-7: Client-Side Certificate provisioning (or Registration) message flow

The call flow for certificate registration (i.e., client-side or 3rd party generation) is as follows:

1. ***The AAA Server sends the EAP-Request/EAP-CREDS(Type=Provisioning) packet.*** The AAA server must include the following TLVs:
   - The ('Protocol') TLV with the value set to the SPP protocol (0x1).
   - The ('Action') TLV with the value set to Registration (0x0) or Renewal (0x1).
   - The ('Credentials-Info') TLV where the fields must identify the specific credential to be Registered or Renewed. In particular, for certificate based credentials, the following values should be used:
     - The *CredType* field must carry a value of (0) to indicate that use of X.509 certificates.
     - The *ProtoID* field must carry a value of (1) to indicate that these credentials have been retrieved via SPP.
     - The *IssuedOn* field must not be set (i.e., 0x0).
     - The *ExpiresOn* field must not be set (i.e., 0x0).
     - The *CredentialsLength* field must be set to zero (0x0).
     - The *CredIDValue* field must be present in case of renewal of credentials and set to the credentials' identifier. The field should be omitted in case of registration of new credentials .
   - The ('Provisioning-Params') TLV where:
     - The *Min Length* field must be set to the minimum value for the key size (in bits) that is supported by the network (i.e., minimum level of security) – if applicable – or zero (0) to indicate no constraints. For example, for RSA keys this field can be set to 2048 while for ECDSA keys this field shall be set to zero (0) since the objectIdentifier (the curve) determines the size of the key.
     - The *Max Length* field must be set to the maximum value for the key size (in bits) that is supported by the network (i.e., maximum key size) – if applicable – or zero (0) to indicate no constraints. For example, for RSA keys this field can be set to 4096 while for ECDSA keys this field shall be set to zero (0) since the objectIdentifier (the curve) determines the size of the key.
     - The *Algorithm* field must be set to (1) if RSA keys are used or to (2) if ECDSA keys are used.
     - The *Flags* bit field must have the following configuration:
       - Bit 0 – Not Set (0),

        - Bit 1 – Set (1),
        - Bit 2 ... 7 – Not Set (0).
    - The *ObjectIdentifier* field must be omitted for RSA keys (i.e., the TLV ends with the flags bitfield). For ECDSA keys, the ObjectIdentifier field must carry the DER representation of the field (or curve) to be used for key generation.
  - No ('Provisioning-Data') TLV shall be present in the packet.

2. ***The UE sends the EAP-Response/EAP-CREDS(Type=Provisioning) packet.*** The UE must include the following TLVs:
   - The ('Protocol') TLV with the value set to the SPP protocol (0x1)
   - The ('Provisioning-Data') TLV with:
     - One ('Credentials-Data') TLV with the following values:
       - The *CredType* field must be set to (0x00) to indicate an X.509 certificate.
       - The *Format* field must be set to (0x00) to indicate raw format (i.e., the certificate itself).
       - The *Encoding* field must be set to (1) for DER encoding.
       - The *Value* field must be set to the value of the certificate that is registered as a credential to access the network.
     - (Optional) One ('Certificate-Data') TLV for each certificate in the chain up to the root (included). In particular, for each ('Certificate-Data') TLV:
       - The *Flags* field must be set according to the type of certificate. In particular, for a trusted root CA the value (0x07) shall be used, while for an intermediate CA certificate, the value shall be set to (0x02).
       - The *Encoding* field shall be set to (0x01) for DER encoding.
       - The *Value* field shall be set to the DER representation of the certificate.

In case of errors, instead of the ('Credentials-Data') TLV, one ('Error') TLV shall be used instead with the appropriate failure code – i.e., no ('Credentials-Info') TLV must be present in this case.

If no error is reported by the Server, the registration and/or renewal of the new (or existing) certificate is completed successfully. The server can now start the Provisioning phase again for a different set of credentials (if any exist and need to be managed), move to the Validation phase, or successfully terminate the EAP-CREDS session by issuing a *Success* packet.

7.5.4 Security Considerations

UEs should generate their own keypair (i.e., by using the co-generation of certificate based credentials) unless there are technical reasons to do otherwise. In particular, when a UE does not have a good random numbers generator on board, the server-side generation of the keys might provide better security as the generated keys will not be guessable based on possible weaknesses in the UE.

8 Support for Bootstrapping Credentials

When the UE does not have any valid credentials for the Network that it is authenticating to, it cannot provide any ('Credentials-Info') TLV in its 'Init' message. This indicates to the Server that new credentials MUST be registered before the UE is allowed on the network.

8.1 Authorizing UE for Credentials' Provisioning

EAP-CREDS provides a simple mechanism for the UE to leverage an out-of-band Token, Passphrase, or One Time Token (OTT) that may already be available on the UE (e.g., a device certificate or a 'spendable' credentials token like a kerberos ticket or a crypto-currency transaction) and that can be verified by the Server.

In particular, when the UE wants to register new credentials (and the Server requires the use of additional authorization data) it may need to provide (a) a Token, (b) a challenge value, and (c) a response to the challenge value.

Note: The method for retrieving this out-of-band information (e.g., an operator's registration portal) is out of scope for this document.

To deliver the registration token, the UE must encode the required token in a ('Token-Data') TLV, the challenge value in a ('Challenge-Data') TLV, and, finally, the response to the challenge in the ('Challenge-Response') TLV.

Although the use of ('Challenge-Data') and ('Challenge-Response') TLVs is optional, it is suggested that if a token is used for bootstrapping the trust, it should provide a way to verify a secret that might be associated with it (i.e., the use of bearer tokens or API keys is discouraged because the disclosure of such tokens enables any device to register – a sound re-use mechanism shall be in place to avoid stealing and/or sharing of these secrets across devices).

8.2 Token-based Trust Bootstrapping

CBRS-A allows for devices to use token-based credentials to register to an access network before the device is actually allowed to join in. In particular, when the UE does not have valid credentials to access the network it must include, in its Initialization message, the ('Token-Data') and the ('Challenge-Response') TLVs. The ('Token-Data') TLV's fields (i.e., Token Type, Encoding, and Value) shall be set according to the token type.

In particular, if no ('Challenge-Data') was sent by the Server in its Initialization message, or if the Token does not have an associated challenge-response mechanism (i.e., there is no secret associated with the token), then the UE shall compute the ('Challenge-Response') TLV's contents by calculating the hash (SHA-256) over the concatenation of the ('Challenge-Data') TLV (if present) and the ('Token-Data') TLV. The value for the ('Challenge-Data') TLV's Type must be (0x02) for EAP-CREDS-SYMMETRIC.

The Server, when registering new devices, shall use a validation mechanism that allows a UE to provide a response to a challenge sent by the server. In case this mechanism is not present, the Server shall verify, at minimum, that the message from the UE contains the ('Challenge-Response') TLV and that its value is properly calculated as described in the previous paragraph. The use of the ('Challenge-Data') TLV can help prevent replay attacks.

8.3 Device Certificate based Trust Bootstrapping

Devices can use certificate-based credentials to register to a CBRS-A network. In particular, when the UE is not yet registered to access the network, it must include, in its Initialization message, the ('Token-Data') and the ('Challenge-Response') TLVs.

In this case, in order to register, the UE shall provide its device certificate in the ('Token-Data') where the Token Type shall be set to (0x04) to indicate an X.509 certificate. The Encoding shall be set to (0x01) to indicate DER is to be used. The Value field shall contain the DER-encoded representation of the certificate used for registration. The ('Challenge-Response') TLV shall contain the signature in a detached CMS [43] calculated over the concatenation of the ('Token-Data') TLV and the ('Challenge-Data') TLV. The CMS data shall be encoded in DER. The TLV and the value for the ('Challenge-Data') TLV's Type must be (0x01) for EAP-CREDS-ASYMMETRIC.

Also in this case, the use of the ('Challenge-Data') TLV can help prevent replay attacks.

8.4 Security Considerations

It is very important that the authorization token is disclosed only to authorized servers - the UE shall verify the identity of the server it is talking to before using any authorization tokens. In particular, the disclosure of authorization tokens that are not meant for the network that is being accessed might leak important information (e.g., the authorization token itself). This can be accomplished, usually, by verifying the identity of the Server first (in the outer mechanism) and then verify that the target of the Token is the Server the UE is actually talking to.

9 EAP-TTLS with MS-CHAP-v2 for 5GC

9.1 General

3GPP TS 33.501 [45] defines four authentication methods including 5G-AKA, EAP-AKA', EAP-TLS, and EAP-TTLS. In addition, according to Annex I of 3GPP TS 33.501 [45], other EAP methods are allowed for Non-Public Network (NPN) as long as the EAP method supports key generation to allow the subsequent key derivation in 5G after the EAP authentication.

Based on Annex U of 3GPP TS 33.501 [45], EAP-TTLS executes between the UE and the AUSF via the SEAF, with the UE, the SEAF and the AUSF taking the roles of EAP peer, EAP pass-through authenticator and the EAP server respectively. After the establishment of TLS tunnel between the UE and the AUSF, MS-CHAP-v2 is executed between the UE and the backend AAA server supporting MS-CHAP-v2 via the NSSAAF. This clause specifies two additional deployment options of EAP-TTLS [8] with MS-CHAP-v2 [10] for 5GC. When user credentials are stored in the UDM, MS-CHAP-v2 is executed between the UE and the UDM. When user credentials are stored in the AAA server, MS-CHAP-v2 is executed between the UE and the AAA server without the NSSAAF.

9.2 EAP-TTLS with MS-CHAP-v2

EAP-TTLS with MS-CHAP-v2 consists of three phases:

1. authentication method selection;
2. TLS tunnel establishment with EAP-TTLS; and
3. execution of MS-CHAP-v2.

9.2.1 Authentication Method Selection

In phase 1 (Figure 9-1), an authentication method is selected by the UDM based on the UE SUPI [44], which is of the type of IMSI or NAI in the form of username@realm. The "username" shall be either "anonymous" or omitted if the subscriber identifier privacy is required by the home network and the public key of the home network is not provisioned in the UE.

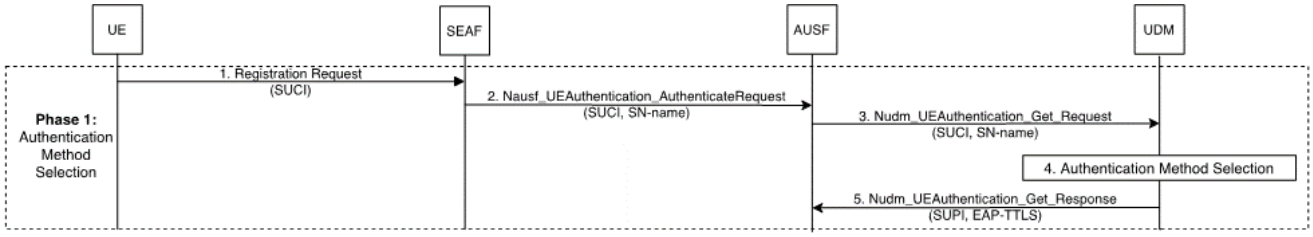

Figure 9-1: Authentication Method Selection (Phase 1)

1. The UE sends to the SEAF a Registration Request message, including the SUCI which is constructed from the UE SUPI.
2. The SEAF sends to the AUSF Nausf_UEAuthentication_Authenticate Request message, including the SUCI and the SN-name (the serving network name).

3. The AUSF sends to the UDM the Nudm_UEAuthentication_Get Request, including the SUCI and the SN-name.
4. The UDM de-conceals the SUCI to obtain the SUPI. If the SUCI is not constructed using the null-scheme, the UDM invokes the SIDF located within the UDM to de-conceal the SUCI. The UDM then selects the primary authentication method of EAP-TTLS, based on the SUPI. If the SUPI is of the form of NAI and the "username" portion of the NAI is omitted or "anonymous", the UDM may select the primary authentication method based on the "realm" portion of the NAI.
5. The UDM sends to the AUSF an indicator of the chosen EAP-TTLS in the Nudm_UEAuthentication_Get Response, which also includes the SUPI.

9.2.2 EAP-TTLS

The second stage of authentication is to establish a TLS tunnel between the UE and the AUSF by executing EAP-TTLS (Figure 9-2).

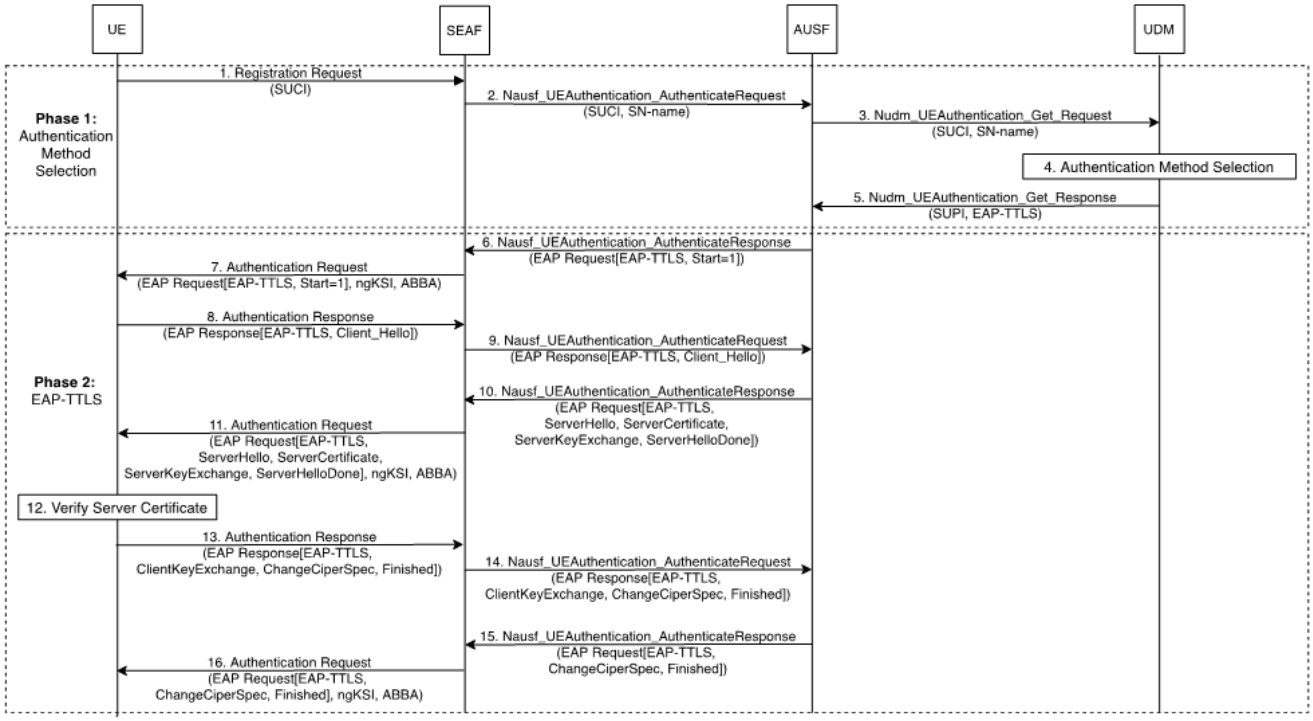

Figure 9-2: EAP-TTLS (Phase 2)

6. The AUSF starts EAP-TTLS by sending to the SEAF a Nausf_UEAuthentication_Authenticate Response message containing an EAP-Request message of EAP-type=EAP-TTLS with the Start (S) bit set, denoted as EAP-Request [EAP-TTLS, Start=1].
7. The SEAF forwards to the UE the EAP-Request [EAP-TTLS, Start=1] in the Authentication Request message, including the ngKSI and the ABBA parameters.
8. The UE replies to the SEAF an Authentication Response message containing an EAP-Response [EAP-TTLS] message whose data field encapsulates a TLS ClientHello message, denoted as EAP-Response [EAP-TTLS, ClientHello].

9. The SEAF forwards to the AUSF the EAP-Response [EAP-TTLS, ClientHello] message in a Nausf_UEAuthentication_Authenticate Request message.
10. The AUSF replies to the SEAF with EAP-Request [EAP-TTLS] message whose data field encapsulates a TLS ServerHello message, a TLS ServerCertificate message, a TLS ServerKeyExchange message, and a TLS ServerHelloDone message. Such EAP-Request message, denoted as EAP-Request [EAP-TTLS, ServerHello, ServerCertificate, ServerKeyExchange, ServerHelloDone], is encapsulated in a Nausf_UEAuthentication_Authenticate Response message.
11. The SEAF forwards to the UE the EAP-Request [EAP-TTLS, ServerHello, ServerCertificate, ServerKeyExchange, ServerHelloDone] message in an Authentication Request message, including the ngKSI and the ABBA parameters.
12. The UE authenticates the AUSF by validating the server certificate included in the EAP-Request message received in step 11. The UE needs to be provisioned with certificates of a trust anchor to validate the AUSF server certificate.
13. If the TLS server authentication is successful, then the UE replies to the SEAF with EAP-Response [EAP-TTLS] in an Authentication Response message. The data field of the EAP-Response [EAP-TTLS] message contains a TLS ClientKeyExchange message, a TLS ChangeCipherSpec message, and a TLS Finished message. This EAP-Response message is denoted as EAP-Response [EAP-TTLS, ClientKeyExchange, ChangeCipherSpec, Finished].
14. The SEAF forwards to the AUSF the EAP-Response [EAP-TTLS, ClientKeyExchange, ChangeCipherSpec, Finished] message in a Nausf_UEAuthentication_Authenticate Request message.
15. The AUSF sends to the SEAF an EAP-Request [EAP-TTLS] message with its data field encapsulating a TLS ChangeCipherSpec message and a TLS Finished message. This EAP-Request message, denoted as EAP-Request [EAP-TLS, ChangeCipherSpec Finished], is encapsulated in a Nausf_UEAuthentication_Authenticate Response message.
16. The SEAF forwards to the UE EAP-Request [EAP-TLS, ChangeCipherSpec Finished] message in an Authentication Request message, including the ngKSI and the ABBA parameters.

By this point, the UE and the AUSF have successfully established a TLS tunnel to execute the MS-CHAP-v2.

9.2.3 MS-CHAP-v2

The phase 3 of the authentication executes MS-CHAP-v2 between the UE and the UDM or the AAA server within the TLS tunnel between the UE and the AUSF established in the phase 2.

When the MS-CHAP-v2 is executed between the UE and the UDM server (Figure 9-3), the interface between the AUSF and the UDM is based on HTTP/2 and security of the interface is based on TLS.

When the MS-CHAP-v2 is executed between the UE and the AAA server (Figure 9-4), the interface between the AUSF and the AAA server shall support RADIUS or Diameter. The security of the interface is based on the security of RADIUS or Diameter.

The message flows of the two use cases differ in only steps 19, 20, and 21.

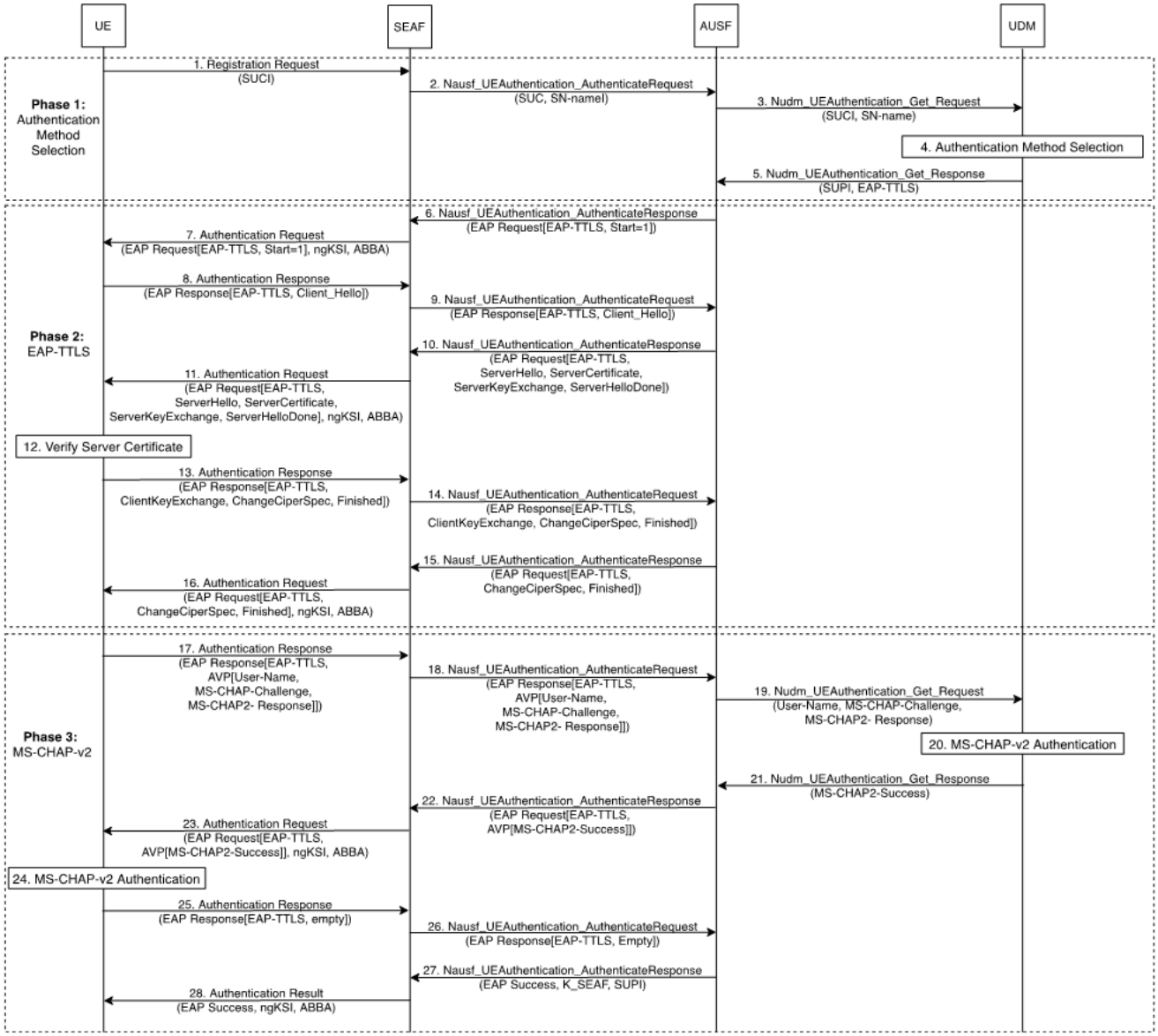

Figure 9-3: MS-CHAP-v2 between the UE and the UDM (Phase 3)

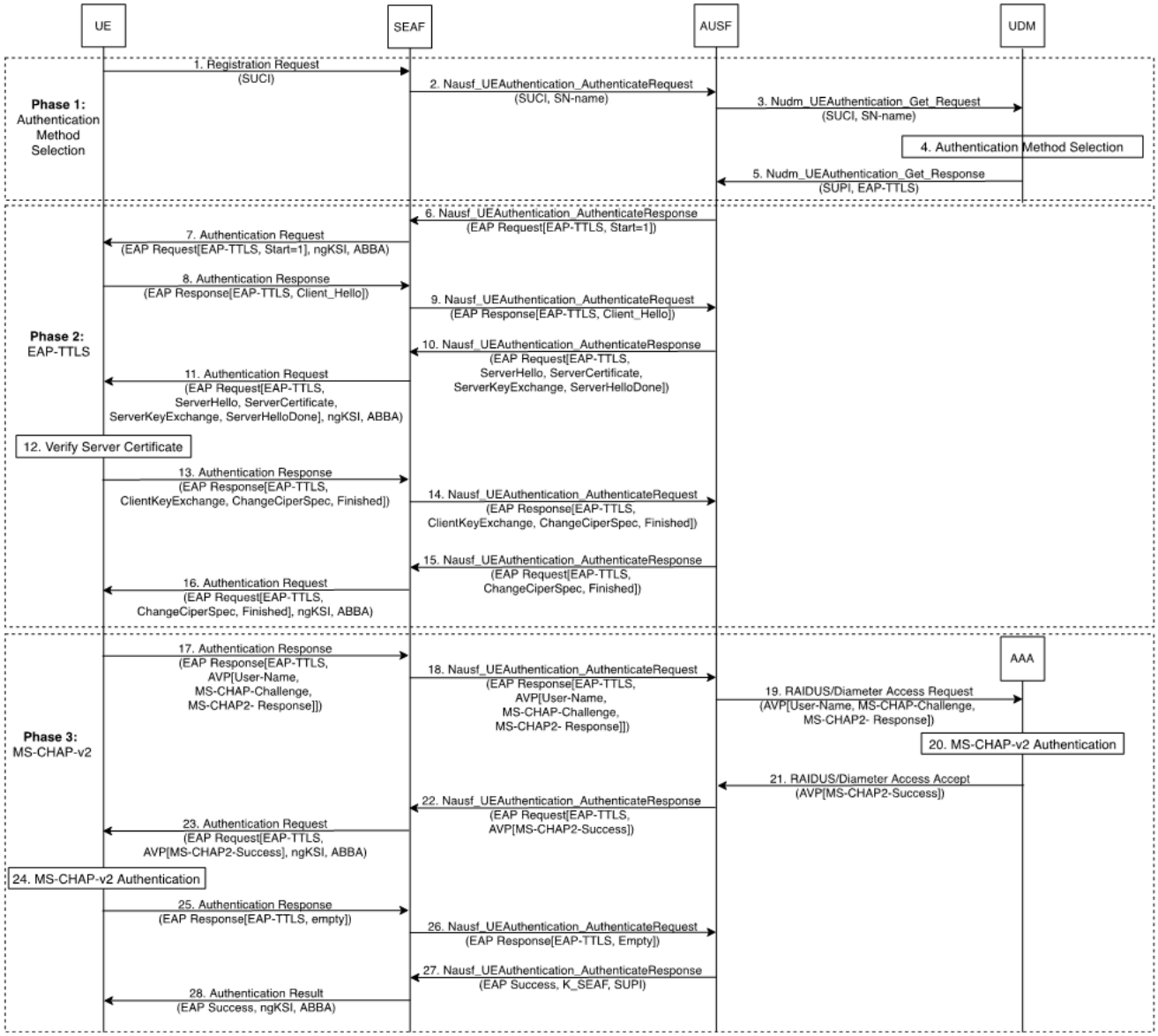

Figure 9-4: MS-CHAP-v2 between the UE and the AAA (Phase 3)

17. The UE starts MS-CHAP-v2 by forwarding to the SEAF an EAP-Request [EAP-TTLS] message in an Authentication Response message. As described in Section 11.2.4 of RFC 5281 [8], the data field of the EAP-Request [EAP-TTLS] encapsulates the AVPs of User-Name, the MS-CHAP-Challenge, and the MS-CHAP2-Response, denoted as EAP-Request [EAP-TTLS, AVP[User-Name, MS-CHAP-Challenge, MS-CHAP2-Response]].

Note: The data field AVP[User-Name, MS-CHAP-Challenge, MS-CHAP2-Response] is protected within the TLS tunnel.

18. The SEAF forwards to the AUSF the EAP-Request [EAP-TTLS, AVP[User-Name, MS-CHAP-Challenge, MS-CHAP2-Response]] in Nausf_UEAuthentication_Authenticate Request message.

19. The AUSF receives the EAP-Request [EAP-TTLS, AVP[User-Name, MS-CHAP-Challenge, MS-CHAP2-Response]] message and decrypts the data field AVP[User-Name, MS-CHAP-Challenge, MS-CHAP2-Response]. The AUSF verifies that the MS-CHAP2-Challenge AVP and the MS-CHAP2-Response AVP according to Section 11.2.4 of RFC 5281 [8].

    If the verification is successful and when the UE shall be authenticated by the UDM (Figure 9-3), the AUSF sends to the UDM the User-Name, MS-CHAP-Challenge, and MS-CHAP2-Response in the Nudm_UEAuthentication_Get Request message.

Note: How to encode the User-Name, MS-CHAP-Challenge, and MS-CHAP2-Response in the Nudm_UEAuthentication_Get Request message is implementation specific.

    If the verification is successful and when the UE shall be authenticated by the AAA (Figure 9-4), the AUSF sends to the AAA the AVP[User-Name, MS-CHAP-Challenge, and MS-CHAP2-Response] in an RADIUS/Diameter Access Request message.

20. The UDM (Figure 9-3) or the AAA server (Figure 9-4) authenticates the UE by verifying the MS-CHAP2-Response value against a computed response based on the received MS-CHAP-Challenge value and the credential associated with User-Name according to RFC 2759 [10].

21. If the UE is authenticated by the UDM and the authentication is successful, the UDM sends to the AUSF an MS-CHAP2-Success in Nudm_UEAuthentication_Get Response message (Figure 9-3).

    If the UE is authenticated by the AAA server and the authentication is successful, the AAA server sends to the AUSF the AVP[MS-CHAP2-Success] in a RADIUS/Diameter Access Accept message (Figure 9-4).

Note: if the AUSF has the access to the user credentials, e.g., provided by UDM in step 5 or locally stored, step 20 can be performed by AUSF and steps 19 and 21 can be eliminated.

22. The AUSF sends to the SEAF an Nausf_UEAuthentication_Authenticate Response message including EAP-Request [EAP-TTLS, AVP[MS-CHAP2-Success]].

23. The SEAF forwards to the UE the EAP-Request [EAP-TTLS, AVP[MS-CHAP2-Success]] in an Authentication request message, including the ngKSI and the ABBA parameters.

24. The UE authenticates the UDM by verifying the MS-CHAP2-Success according to RFC 2759 [10].

25. If the authentication is successful, the UE sends to the SEAF an EAP-Response [EAP-TTLS] message with its data field set to empty, denoted as EAP-Response [EAP-TTLS, empty], in an Authentication response message.

26. The SEAF forwards the EAP-Response [EAP-TLS, empty] message to the AUSF in a Nausf_UEAuthentication_Authenticate Request message.

27. The AUSF derives KAUSF and KSEAF according to 3GPP TS 33.501 [45]. The AUSF sends to the SEAF an EAP-Success message along with the SUPI and the KSEAF in a Nausf_UEAuthentication_Authenticate Response message.

28. The SEAF forwards to the UE the EAP-Success message in an Authentication Result message or a Security Mode Command message.

    Upon receiving the EAP-Success message, the UE derives the KAUSF and the KSEAF in the same way as the AUSF according to 3GPP TS 33.501 [45].

By this point, the EAP-TTLS and MS-CHAP-v2 have been successfully completed.

Appendices (Informative)

Appendix A Trust Management

This Annex provide guidelines for the deployment of trust infrastructures that can be used for both UEs and the authentication infrastructure (i.e., AAA and OSU servers). Moreover, this Annex also consider the impact of deploying Online Sign Up (OSU) services to provide certificate-based credentials to UEs.

Appendix A.1 Centralized vs. Distributed PKIs

There are two main PKI models that operators can adopt. The first one is a centralized model where certificates are issued within a common PKI. This model allows the operator to have full control over the structure of the PKI and the certificate profiles.

The second model is a distributed model where participating operators use a common PKI (i.e., a common Root CA) where each operator will be able to obtain and operate its own Intermediate CA. This model allows the possibility to use a single Trust Anchor in UEs to validate the authentication infrastructure's credentials by using a single Trust Anchor.

Note: In this section, the term "CBRS Infrastructure Authentication CA" is use in a generic sense and refers to a generic (non CBRS operated) Certification Authority that issues certificates to be used in the CBRS context.

The following Figure A-1 provides a minimum viable PKI for providing certificates for the authentication infrastructure (i.e., server-side authentication):

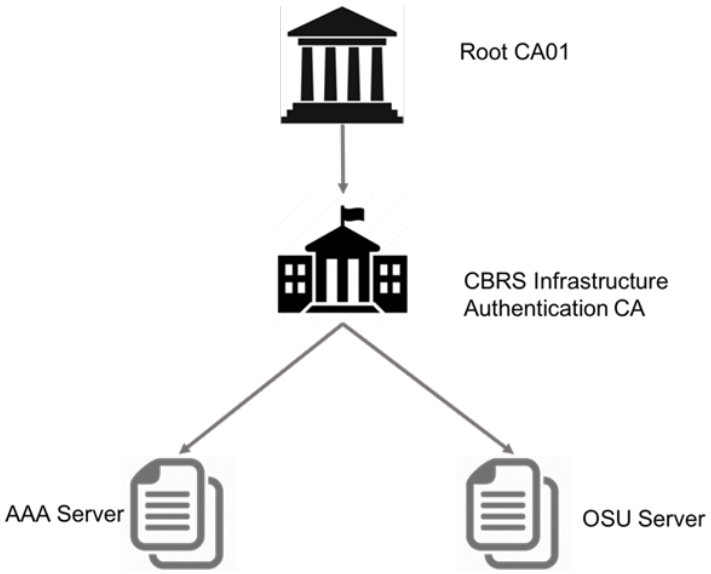

Figure A-1: Minimum viable PKI for providing certificates for the authentication infrastructure

In this model, the AAA Server Certificate is used to authenticate the AAA server to the UE during the extended authentication process (i.e., EAP-TTLS and EAP-TLS methods) while the OSU Server Certificate is used to authenticate the OSU server during the UE registration process (i.e., during the TLS session establishment when the UE connects to the OSU server).

Once established, this PKI can also be used to provision subscribers' certificate to UEs via the OSU server (if supported) or via other out-of-band mechanisms.

Appendix A.2 UE Subscriber Certificate Provisioning for EAP-TLS

The provision of UE certificates can happen through different processes and protocols: out-of-band or online (in-band) mechanisms. The out-of-band provisioning case (e.g., via a web portal or via pre-installed credentials in USIM), is not covered in this document.

For the second case, i.e. online provisioning, the operator may decide to support certificate provisioning and installation procedures as described in Section 5.14 of MFA [6] by deploying support for an Online Sign Up (OSU) server or other mechanisms (e.g., via the EAP protocol itself).

Appendix A.2.1 Network Impact and Security Considerations for OSU deployment

In case the operator decides to deploy support for OSU, the operator shall follow the procedures for on-boarding new clients as described in Section 5.14 of [6]. In this case, Section 5.5.2.1 of TS 1002 [3] is modified to include support for Online Sign-up (OSU). In particular, the deployment of the OSU server adds several requirements from a network perspective and introduces important security requirements that the SP must take into serious consideration.

In particular, the deployment of the OSU component requires the following changes in the SP's CBRS network:

- Support for the OSU must be broadcasted by the network. In particular, the OSU support information is delivered to the UE by using SDP Query for OSU information as described in Section 5.10.7 of MFA [6].
- The UE shall request a PDN connection for a default Access Point Name (APN) after indicating its intention to engage with the OSU process by using an OSU-specific Attach Type.
- The network operator shall support the UE to enter a sub-state of the EMM-REGISTERED that provides a PDN connection restricted to provisioning a specific (set of) OSU server(s) and does not grant access to normal service.
- Depending on the type of credentials that were provisioned during the onboarding process, the OSU AAA server must be able to update the SP's AAA server with the new credentials

Appendix A.3 Considerations about Manufacturer (or Device) Certificates

When using extended authentication or when the UE attaches in Credentials Provisioning mode, the use of a device certificate is suggested to provide client-side authentication both in the EAP-TTLS case and the EAP-TLS one (in credentials Provisioning Attach mode only). The manufacturer certificate is used to convey some important parameters that may be also included in the Subscriber's certificate profile in case the operator wants to tie the subscription to a specific device.

Note: It is important to understand that the manufacturer (or device) certificate is not related to the user subscription, but it is a device-specific identity only.

The provisioning mechanism of this type of certificate is not specified in this document. In particular, it is assumed that the device is pre-installed with the manufacturer (or device) certificate (and the corresponding private key) by the manufacturer in a secure fashion. In the manufacturer certificate, the following extensions shall be present:

- IMEI (OID 1.3.6.1.4.1.40808.1.1.3)
- MEID (OID 1.3.6.1.4.1.40808.1.1.4)
- DEVID (OID 1.3.6.1.4.1.40808.1.1.5)

Optionally, the following field might be present:

- MACADDRESS (OID 1.3.6.1.1.1.1.22)

In order to foster interoperability across manufacturers, operators might decide to provide the possibility to issue a manufacturer's CA certificate for approved devices under a common PKI or under a specific operator PKI. Additionally, operators might decide to include the manufacturer's Root CA into their authentication servers (e.g., OSU and AAA) to be able to correctly validate the device certificate when used.

It is suggested that manufacturers who would like to obtain a certificate from the common or operator-specific PKIs (i.e., for issuing certificates for its own devices) shall pass an audit against the Certificate Policy that applies to the specific PKI. The following Figure A-2 depicts an extended model for the PKI which includes the Device Manufacturers' Intermediate Certification Authorities:

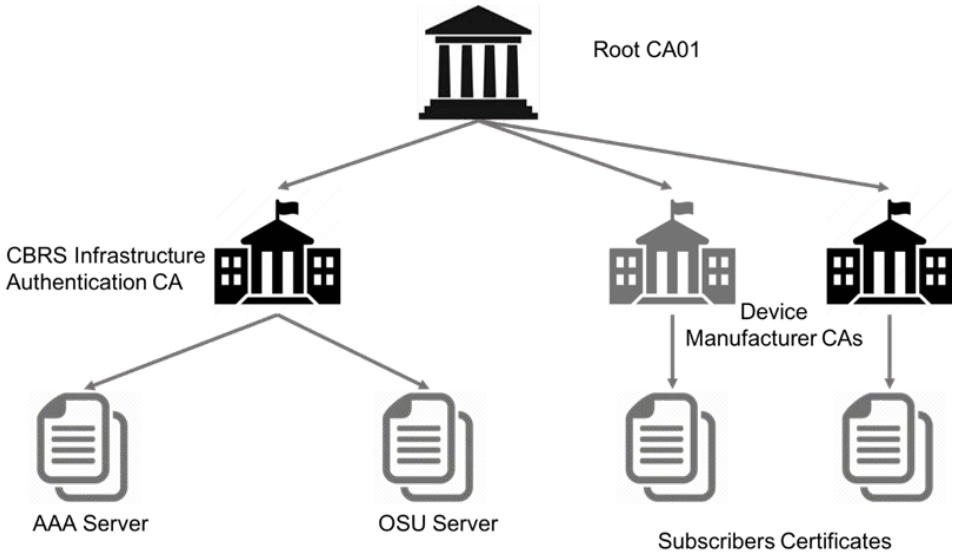

Figure A-2: Extended model for the PKI

Appendix A.3.1 User Equipment Trust Anchors Installation

In order for the UE to be able to verify the authentication infrastructure, UEs must be installed with the Root CA certificate (e.g., Root CA01). This allows the UE to verify that the chain of certificates presented during authentication by both the AAA server and the OSU server anchors to a trusted entity.

Appendix A.3.2 Authentication Infrastructure Trust Anchors Installation

In order for AAA and OSU servers to be able to verify the identity of devices (if manufacturer's certificates are installed on the UEs) or of subscribers (if EAP-TLS is used as the authentication mechanism), the Root CA certificates must be installed. If a common PKI is used for both subscribers and manufacturers, then only a single root is required to be installed on the server. In case different PKIs are used instead, all the Root CAs from different manufacturers are required on all servers.

Appendix A.4 Certificates Profiles

This section provides examples for the definition of the profile for the different types of certificates for the authentication infrastructure (i.e., AAA and OSU servers). The profile for subscriber certificates is left unspecified as this might varies greatly among providers.

Table A-1: CBRS Authentication Infrastructure – Root CA Certificate Profile

Attribute Name		**Settings**		
Version		v3		
Serial number		Unique Positive Integer assigned by the CA		
Issuer DN		o=<organization name> ou=Root CA01 cn=CBRS Root Certification Authority		
Subject DN		o=<organization name> ou=Root CA01 cn=CBRS Root Certification Authority		
Validity Period		48 yrs		
Signature Algorithm		Sha256WithRSAEncryption (1 2 840 113549 1 1 11)		
Public Key		RSA 4096 bits		
Parameters		NULL		
Standard Extensions	**OID**	**Include**	**Criticality**	**Value**
keyUsage	{id-ce 15}	X	TRUE	
keyCertSign				Set
cRLSign				Set
basicConstraints	{id-ce 19}	X	TRUE	
cA				Set
subjectKeyIdentifier	{id-ce 14}	X	FALSE	
keyIdentifier				Calculated per Method 1
subjectAltName	{id-ce 17}	O	FALSE	
directoryName				Set by the issuing CA

Table A-2: CBRS Authentication Infrastructure – Intermediate CA Certificate Profile

Attribute Name		**Settings**		
Version		v3		
Serial number		Unique Positive Integer assigned by the CA		
Issuer DN		o=<organization name> ou=Root CA01 cn=CBRS Root Certification Authority		
Subject DN		o=<organization name> ou=Infrastructure Authentication cn=Certification Authority 01		
Validity Period		Up to 16 yrs		
Signature Algorithm		Sha256WithRSAEncryption (1 2 840 113549 1 1 11)		
Public Key		RSA 4096 bits		
Parameters		NULL		
Standard Extensions	**OID**	**Include**	**Criticality**	**Value**
keyUsage	{id-ce 15}	X	TRUE	
keyCertSign				Set
cRLSign				Set
basicConstraints	{id-ce 19}	X	TRUE	
cA				Set
pathLenConstraint				0
subjectKeyIdentifier	{id-ce 14}	X	FALSE	
keyIdentifier				Calculated per Method 1
authorityKeyIdentifier	{id-ce 35}	X	FALSE	
keyIdentifier				Calculated per Method 1

Table A-3: CBRS Authentication Infrastructure – AAA Server Certificate

Attribute Name	**Settings**
Version	v3
Serial number	Unique Positive Integer assigned by the CA

Attribute Name		Settings		
Issuer DN		o=<organization name> ou=CBRS Infrastructure Authentication cn= Certification Authority 01		
Subject DN		o=<organization name> ou=CBRS Infrastructure Authentication ou=AAA Services cn=<server FQDN>		
Validity Period		Up to 4 yrs		
Signature Algorithm		Sha256WithRSAEncryption (1 2 840 113549 1 1 11)		
Public Key		RSA 2048 bits		
Parameters		NULL		
Standard Extensions	**OID**	**Include**	**Criticality**	**Value**
keyUsage	{id-ce 15}	X	TRUE	
digitalSignature				Set
keyEncipherment				Set
authorityKeyIdentifier	{id-ce 35}	X	FALSE	
keyIdentifier				Calculated per Method 1
subjectAltName	{id-ce 17}	X	FALSE	
dNSName				<server FQDN>

Table A-4: CBRS Authentication Infrastructure – OSU Server Certificate

Attribute Name	Settings
Version	v3
Serial number	Unique Positive Integer assigned by the CA
Issuer DN	o=<organization name> ou=CBRS Infrastructure Authentication cn= Certification Authority 01
Subject DN	o=<organization name> ou=CBRS Infrastructure Authentication ou=OSU Services cn=<server FQDN>
Validity Period	Up to 4 yrs
Signature Algorithm	Sha256WithRSAEncryption (1 2 840 113549 1 1 11)

Attribute Name		**Settings**		
Public Key		RSA 2048 bits		
Parameters		NULL		
Standard Extensions	**OID**	**Include**	**Criticality**	**Value**
keyUsage	{id-ce 15}	X	TRUE	
digitalSignature				Set
keyEncipherment				Set
authorityKeyIdentifier	{id-ce 35}	X	FALSE	
keyIdentifier				Calculated per Method 1
subjectAltName	{id-ce 17}	X	FALSE	
dNSName				<server FQDN >

All certificates issued under this PKI shall follow the procedure described in a Certificate Policy that governs the practices followed by the selected Certificate Service Provider.

Appendix B Using Vendor-Specific provisioning in EAP-CREDS

In order to use EAP-CREDS together with your another provisioning protocol, the messages from the provisioning protocol need to be sent to the other party. In EAP-CREDS, this is done by encoding the provisioning protocol messages inside the ('Provisioning-Data') TLV. In case the provisioning protocol uses additional data for its operations (e.g., uses HTTP Headers), this data can be encoded in a separate ('Provisioning-Headers') TLV.

Since the implementation of the provisioning endpoint could happen in a (logically or physically) different component, a method is needed to identify when a provisioning protocol has actually ended. In EAP-CREDS, the 'D' bit in the message headers is used for this purpose.

In the first message of Phase Two, the Server provides the client with all the selected parameters for one specific credential that needs attention (or for a new credential) to be managed by the network. In particular, the server provides, at minimum, the ('Protocol') TLV, the ('Action') TLV, and the ('Provisioning-Params') or the ('Credentials-Info') TLV.

After checking the parameters sent by the Server, if the Peer does not support any of the proposed ones, it MUST send a message with one single ('Error') TLV with the appropriate error code(s). The server, can then decide if to manage a different set of credentials (if more where reported by the Peer in its Phase One message) or if to terminate the EAP session with an error.

The Peer and the Server exchange Provisioning messages until an error is detected (and the appropriate error message is sent to the other party) or until Phase Two is successfully completed.

Appendix C Change History

Table C-1: Change History

Version	Date	Description
V0.1	YYYY-MM-DD	Initial Draft
V1.0 (r0.1.1)	2018-01-17	First Published Version of the Document.
V1.0 (r0.1.2)	2018-02-05	Added certificate Management and Validation section.
V1.0 (r0.1.4)	2018-03-06	Added references for certificates management and installation. Added requirements for supporting OSU server. Initial Trust Infrastructure discussion.
V1.0 (r0.2.0)	2018-04-14	Updated document scope, added security considerations, added credentials storage considerations, updated considerations for OSU deployment, general document restructuring.
V1.0 (r0.2.1)	2018-04-15	Added EAP tunneling details and requirements.
V1.0 (r0.2.2)	2018-04-16	Added missing reference to MSCHAPv2, Updated terminology for 3GPP and NHN access modes.
V1.0 (r0.3.0)	08-05-2018	Reorganized the document with Stage 2 and 3 sections.
V1.0 (r0.3.3)	2018-05-18	Added call flows for EAP-TLS and EAP-TTLS. Extended considerations and details about EAP-TTLS usage.
V1.0 (r0.3.4)	2018-05-21	Added call flows for EAP-TLS and EAP-TTLS. Extended considerations and details about EAP-TTLS usage.
V1.0 (r0.3.5)	2018-06-25	Added Stage 2 and 3 initial considerations for 3GPP-based (non-EPS-AKA) access mode.
V1.0 (r0.3.8)	2018-07-09	Included last changes for Stage 2 and 3 coming from the CL's Gap Analysis in 3GPP specs.
V1.0 (r0.3.9)	2018-07-13	Added considerations about Policy to address SPR and PCRF.
V2.0 (r4.0)	2018-10-27	Implemented comment resolutions from the balloting process.
V2.0 (r4.1)	2018-10-31	Removed “Contributors”, fixed typos, fixed inconsistent use of lower and upper cases, and added missing section number.
V2.0 (r4.2)	2018-12-14	Implemented comment resolutions from re-balloting process.
V2.0 (r4.3)	2019-01-11	Reformatting to be consistent with TS-1002.
V3.0 (r8)	2019-07-15	Added credentials management sections (7,8) and Appendix B
V3.0 (r9)	2019-07-24	Completed description for server-generated and co-generated username/password provisioning.
V3.0 (r10)	2019-08-12	Added security considerations for secrets generation.
V3.0 (r11)	2019-09-23	Integrated Changed to include 5G NR

Version	Date	Description
V3.0 (r12)	2019-10-14	Finalized Normative Section for Credentials Management
V3.0 (r13)	2019-10-14	Fixed username/password and certificate co-gen. Appendix B.
V3.0 (r14)	2019-01-12	Incorporated Feedback from ballot process for Release 3
V4.0 (r1)	2020-06-22	Introduced EAP-TTLS for 5GC.
V4.0 (r2)	2020-06-27	Changed version number from “Version 4.0.0 (rev 1)” to “version 3.2.0 (toward V4.0.0).
V4.0 (r3)	2020-07-18	Updated Section 1 and Section 9. Missing abbreviations added; updated “change history”.
V4.0 (r4)	2020-07-20	Updated the format of the version numbers from “version 3.4.0 (toward V4.0.0)” to “Version 3.0.0 (rev 4, moving toward V4.0.0).
V4.0 (r5)	2020-09-14	Implemented comments from the ballot process for Release 4.
V4.1 (r1)	2022-05-09	Implemented changes to align with 3GPP TS 33.501 (v17.5.0) on EAP-TTLS.
V4.1 (r2)	2022-12-30	Update to new template.