Credentials Provisioning and Management via EAP (EAP-CREDS)

draft-pala-eap-creds-01

Abstract

With the increase number of devices, protocols, and applications that rely on strong credentials (e.g., digital certificates, keys, or tokens) for network access, the need for a standardized credentials provisioning and management framework is paramount. The .1x architecture allows for entities (e.g., devices, applications, etc.) to authenticate to the network by providing a communication channel where different methods can be used to exchange different types of credentials. However, the need for managing these credentials (i.e., provisioning and renewal) is still a hard problem to solve.

This specifications define how to support the provisioning and management of authentication credentials that can be exploited in different environments (e.g., Wired, WiFi, cellular, etc.) to users and/or devices by using EAP together with standard provisioning protocols.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 3, 2019.
Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Requirements notation ........................................... 3
2. Introduction ....................................................... 3
3. Overview of existing solutions ................................. 4
4. Scope Statement ................................................ 4
5. EAP-CREDS Protocol ............................................... 5
   5.1. EAP-CREDS as tunneled mechanism only .................... 5
   5.2. Message Flow .............................................. 5
   5.3. Phase One: Initialization ................................. 6
   5.4. Phase Two: Provisioning ................................. 7
   5.5. Phase Three: Validation ................................. 8
6. EAP-CREDS Message Format .................................... 10
   6.1. Message Header ............................................ 10
   6.2. Message Payload .......................................... 11
      6.2.1. General TLV format ................................ 11
      6.2.2. EAP-CREDS defined TLVs .............................. 12
         6.2.2.1. The Protocol TLV ................................. 13
         6.2.2.2. The Profiles TLV ................................ 13
         6.2.2.3. The CredsInfo TLV ............................... 13
         6.2.2.4. The AuthToken TLV ............................... 14
         6.2.2.5. The Challenge TLV ............................... 14
         6.2.2.6. The ChallengeResponse TLV ...................... 14
         6.2.2.7. The Error TLV .................................. 14
7. EAP-CREDS Messages ........................................... 14
   7.1. The EAP-CREDS-Init Message .............................. 14
      7.1.1. Version TLV usage .................................. 14
      7.1.2. The ProvProto TLV usage ............................ 15
      7.1.3. The CredsInfo TLV usage ............................ 15
      7.1.4. The IdProvider TLV ................................. 15
   7.2. The EAP-CREDS-ProtoFlow Message ....................... 16
      7.2.1. The ProvProtoHeader TLV ............................ 16
      7.2.2. The ProvProtoData TLV ............................... 16
1. Requirements notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Introduction

Many environments are, today, moving towards requiring strong authentication when it comes to gain access to networks. The .1x architecture provides the network administrators with the possibility to check credentials presented by a device even before providing any IP service to it.

However, the provisioning and management of these credentials is a hard problem to solve and many vendors opt for long-lived credentials that can not be easily revoked, replaced, or simply renewed.

This specification addresses the problem of providing a simple-to-use and simple-to-deploy conduit for credentials management by extending the EAP protocol to support credentials provisioning and management functionality. In particular, the EAP-CREDS method defined here provides a generic framework that can carry the messages for provisioning different types of credentials. The method can be use as a stand-alone method or it can be used as an inner methods of EAP-TTLS, EAP-TEAP, or any other tunnelling method that can provide the required encryption and (at minimum) server-side authentication to
deliver server-side generated secrets (e.g., private keys, passwords, secret keys, etc.)

3. Overview of existing solutions

Currently there are many protocols that address credentials lifecycle management. In particular, when it comes to digital certificates, some of the most deployed management protocols are:

- Certificate Management Protocol (CMP) [RFC4210]
- Certificate Management over CMS (CMC) [RFC5272][RFC6402]
- Enrollment over Secure Transport (EST) [RFC7030]
- Automated Certificate Management Environment

However, none of these protocols provide native support for client that do not have IP connectivity yet (e.g., because they do not have network-access credentials, yet). The EAP-CREDs provides the possibility to use such protocols (i.e., message-based) by defining a series of messages that can be used to encapsulate the provisioning messages for the specified protocol. In particular, examples of the message flow for the major provisioning protocols are provided in Annex Appendix A.

In addition to these protocols, EAP-CREDs also defines a series of simple messages that provide a generic enrollment protocol that allows not only certificates but also other types of credentials (e.g., username/password pairs or symmetric secrets) to be delivered to the client as part of the provisioning and/or renewal process. The set of messages that make up the generic provisioning protocol is referred to as the CREDS protocol (not to be confused with the EAP-CREDs).

4. Scope Statement

This document focuses on the definition of the EAP-CREDs method to convey credentials provisioning and managing messages between the client and the AAA server. Moreover, the document defines how to encode messages for the main IETF provisioning protocols.

This document, however, does not provide specifications for how and where the credentials are generated. In particular, the credentials could be generated directly within the AAA server or at a different location (i.e., the Certificate Service Provider or CSP) site. Different authentication mechanisms (e.g., TLS, etc.) can be used to secure the communication between the server’s endpoint and the CSP.
5. EAP-CREDS Protocol

This section outlines the operation of the protocol and message flows. The format of the CREDS messages is given in Section 6.

5.1. EAP-CREDS as tunneled mechanism only

EAP-CREDS requires that an outer mechanism is in place between the Peer and the Server in order to provide authentication and confidentiality of the messages exchanged via EAP-CREDS. This choice was taken to simplify the message flow and abstract EAP-CREDS from the secure-channel establishment mechanism.

In other words, the method assumes that an appropriate protection outer layer has been established to prevent the possibility to leak information or to allow man-in-the-middle attacks.

5.2. Message Flow

EAP-CREDS message flow is logically subdivided into three different phases:

**Phase One (Required). CREDS Initialization.** During this phase the Peer and the Server exchange the information needed to select the appropriate credentials management protocol. In particular, the Server sends its initial message of type ('Init') with the details about what protocols are supported, and additional information such as the version of EAP-CREDS and the supported profiles identifiers.

**Phase Two (Required). Provisioning Protocol Flow.** In this phase, the Peer and the Server exchange the provisioning protocol’s messages encapsulated in a EAP-CREDS message of type ProtoFlow. The messages contain only two TLVs: the first one (optional) carries information that might be normally conveyed via the transport protocol (e.g., HTTP headers), while the second one (required) carries the provisioning protocol’s messages.

**Phase Three (Optional). Credentials Validation.** This optional phase can be initiated by the server and it is used to validate that the Peer has properly installed the credentials and can use them to authenticate itself. Depending on the credentials’ type, the messages can carry a challenge/nonce, the value of the secret/token, or other information. The format of the credentials is supposed to be known by the provider and the device.
5.3. Phase One: Initialization

The following figure provides the message flow for Phase One:

```
+------------+                                  +--------------+
|  EAP Peer  |                                  |  EAP Server  |
+------------+                                  +--------------+
|                                                  |
|<----------- Outer Tunnel Established ----------->|  Phase One  |
|                                                  |  Begins     |
|                                                  |             |
|<---------[1] EAP-Request/EAP-CREDS ---------------|             |
|             (Type=Init,Ver,Supported Protocols,   |
|                   Providers)                      |             |
|                                                  |             |
|<---------[2] EAP-Response/EAP-CREDS ---------------|             |
|             (Type=Init,Ver,Proto,Provider)       |
|                                                  |             |
|<---------[3] EAP-Request/EAP-CREDS ---------------|             |
|             (Type=Init, << Empty >>)             |
|                                                  |             |
|                                                  |  Phase One  |
|                                                  |  Ends       |
```

During the CREDS initialization, after the establishment of the outer mechanism (e.g., EAP-TLS, EAP-TEAP, EAP-TTLS, etc.), the initial the server sends the EAP-Request/EAP-CREDS(Type=Init) message with the ('Versions') TLV to indicate the supported versions of EAP-CREDS (i.e. a list of all supported version of EAP-CREDS), the ('Supported-Protocol') TLV to indicate the list of supported provisioning protocols, followed by the ('Providers') TLVs that allow the client to pick the credentials' vendor.

The peer selects the version, the protocol, and optionally the provider, and sends back the response in a EAP-Response/EAP-CREDS(Type=Init) message to the server which includes the ('Versions'), ('Protocols'), and optionally the ('Providers') TLVs with only one value each. If multiple values are detected in the message from the Peer, the message shall be discarded and the appropriate error message shall be issued by the Server.

The server checks that the requested protocol and parameters are supported and, if not (or if the server detects an error), it sends an error message to the peer and notify the outer (tunneling) layer.
In case the server can serve the request from the client, it sends an empty EAP-Request/EAP-CREDS(Type=Init) message to indicate that the server is ok with the selected parameters and is waiting on the Peer to start Phase Two of the protocol.

5.4. Phase Two: Provisioning

The following figure provides the message flow for Phase 2:

```
+------------+                                  +--------------+
|  EAP Peer  |                                  |  EAP Server  |
+------------+                                  +--------------+

| EAP-Response/EAP-CREDS -------------> |  Phase Two | Begins |
| (Type=ProtoFlow,ProtoData)          |            |       |

| EAP-Request/EAP-CREDS ------------> |            |       |
| (Type=ProtoFlow,ProtoData)         |            |       |

The server starts phase two of the EAP-CREDS protocol by sending an EAP-Request/EAP-CREDS(Type=ProtoFlow) message with the selected protocol’s details to the Peer. This indicates that the Server is ready to initiate the provisioning protocol.

Specifically, the Server MUST include the ‘Ver’ and ‘Proto’ TLVs to indicate the EAP-CREDS version agreed upon by the parties and the specific provisioning protocol to use. The ‘provider’ TLV is optional and MUST be included if a selection was made by the Peer. The ‘provider’ TLV MAY be included in the message even if the Peer did not make a selection.
After that, the Peer sends its first message to the Server by sending the EAP-Response/EAP-CREDS(Type=ProtoFlow) message. This message contains the selected provisioning protocol’s message data and some extra fields (e.g., transport-protocol headers).

The Server replies to the Peer’s message with EAP-Request/EAP-CREDS(Type=ProtoFlow) messages until the provisioning protocol reaches an end (the client will send a ‘ProtoFlow’ message with an empty body) or an error condition (the party that detected the error condition will use a ‘ProtoFlow’ message with an ‘Error’ TLV to convey the issue and terminate the protocol).

The communication between the client and the server continues until the selected protocol and action is correctly performed or a failure is detected and reported.

5.5. Phase Three: Validation

The following figure provides the message flow for Phase 3:
Phase three is optional and it is used by the server to request the client to validate (proof) that the new credentials have been installed correctly before issuing the final Success message.

In order to do that, the server and the client exchange two or three EAP-Request/EAP-CREDS(Type=Validate) and EAP-Response/EAP-CREDS(Type=Validate) messages where the correctness of the exchanged credentials is verified by the server and (optionally) by the client.

In particular, when the client receives the first Validate message from the server, it calculates the response to the challenge (based on the type of credentials and the protocol itself - if supported, this would be a ‘test’ authentication) and sends the response back to the server.
Optionally, the client can add the ExtraChallenge TLV that carries
the extra challenge information that was used to calculate the
ChallengeResponse TLV (if any). This field can be used to prevent
the use of the Validate phase as an encryption or validation oracle.

Optionally, the client can add the Challent TLV to the response.
When the server receives the EAP-Response/EAP-CREDS(Type=Validate)
with the Challenge TLV in it, it MUST calculate the response to the
challenge and send back the response to the client in an EAP-Request/
EAP-CREDS(Type=Validate) with the ChallengeResponse TLV and,
optionally, the ExtraChallenge TLV.

In case of issues with the validation of the newly deployed
credentials, both the client and the server should consider those
credentials invalid (or unusable) and should issue the required
failure message(s).

6. EAP-CREDS Message Format

The EAP-CREDS defines the following message types:

EAP-CREDS/Init
EAP-CREDS/ProtoFlow
EAP-CREDS/Validate

Each of these message types have the basic structure as identified in
Section 6.1 and in Section 6.2 and contain zero, one, or more TLVs.
The allowed TLVs for the different types of messages are described in
Section 7. The internal structure of the different types of TLVs is
described in Section 6.2.1.

6.1. Message Header

The EAP-CREDS messages consist of the standard EAP header (see
Section 4 of [RFC3748]), followed by the version of the EAP-CREDS (4
bits) and a field (4 bits) reserved for future use. The header has
the following structure:
Where the Code, Identifier, Length, and Type fields are all part of the EAP header as defined in [RFC3748].

The Type field in the EAP header is <TBD> for EAP-CREDS.

The Ver (Version) bitfield (4 bits) identifies the version of the EAP-CREDS protocol used for the exchange. This document defines the value to use for this field as ‘0x1’ (0001).

The Flags bitfield (4 bits) is currently not used and it is reserved for future use. The value of the bitfield shall be ignored when the Version (‘Ver’) field is set to ‘0x1’ (0001).

6.2. Message Payload

The Data part of the message is organized as zero, one, or more TLV objects whose structure is defined in this section. In particular, the general structure of a TLV is described in Section 6.2.1, while the specific structures for the supported TLVs is provided in Section 6.2.2.

6.2.1. General TLV format

Each TLV object has the same basic structure that is defined as follows:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| TLV Type | Length |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Value... |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where:

TLV-Type
This field is used to indicate the type of data that the TLV carries. The different types of TLVs are described in Section 6.2.2.

Length

This field carries the size of the value of the TLV. In particular, the overall size of a TLV (i.e., the header plus the value) can be calculated by adding the size of the header (6 octets) to the value of the Length field (i.e., the size of the TLV’s value).

TLV Type

This field carries the type of TLV which determines its internal structure. The supported values for this fields are provided in the following table:

<table>
<thead>
<tr>
<th>Message Type</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Error TLV</td>
</tr>
<tr>
<td>1</td>
<td>EAP-CREDS Version TLV</td>
</tr>
<tr>
<td>2</td>
<td>Provisioning Protocol TLV</td>
</tr>
<tr>
<td>3</td>
<td>Provisioning Provider TLV</td>
</tr>
<tr>
<td>4</td>
<td>Provisioning Protocol Header TLV</td>
</tr>
<tr>
<td>5</td>
<td>Provisioning Protocol Data TLV</td>
</tr>
<tr>
<td>6</td>
<td>Credentials Information TLV</td>
</tr>
<tr>
<td>7</td>
<td>Authorization Token TLV</td>
</tr>
<tr>
<td>8</td>
<td>Registration Token TLV</td>
</tr>
<tr>
<td>9</td>
<td>Challenge TLV</td>
</tr>
<tr>
<td>10</td>
<td>Challenge Response TLV</td>
</tr>
</tbody>
</table>

Table 1: EAP-CREDS Supported TLVs Types

6.2.2. EAP-CREDS defined TLVs

EAP-CREDS messages’s payload comprises zero, one, or more TLVs that are encoded in a single EAP-CREDS message. The values for the TLV Type that are supported by this specifications are listed in Table 1.

This section describes the structure of the different supported TLVs and their usage in the different messages.
6.2.2.1. The Protocol TLV

TLV Type

<TBD> - Provisioning Protocol TLV

Length

4 octets

Protocol ID

The Protocol ID field is 2 octets. The allowed values for this field are maintained in a subdirectory by IANA. The initial values for the provisioning protocol identifiers can be found in Section 13.1.

Protocol Version

The Protocol Version field is 2 octet (uint16). When no version is specified for a protocol (i.e., either it does not support multiple versions or it does not matter), the value should be set to ‘0x0’.

6.2.2.2. The Profiles TLV

6.2.2.3. The CredInfo TLV

Type (Int; 1Byte) | HashAlgoId (Int; 1 byte) | Installed On (GMT; 15 bytes) | ExpiresOn (GMT; 15 bytes) | Salt/Nonce (RND; 32 bytes) | id_len (Int; 2 bytes) | id_value | cred_hash_len (Int; 2 bytes) | Cred_Hash (Binary; 20-64 bytes)
6.2.4. The AuthToken TLV

6.2.5. The Challenge TLV

6.2.6. The ChallengeResponse TLV

6.2.7. The Error TLV

7. EAP-CREDS Messages

This section describes each message and what TLVs are allowed or required. EAP-CREDS defines the following values for the Message Type (Type):

+------------+---------------------+--------------------------------+
|  Message   | Name                | Description                    |
|    Type    |                     |                                |
|------------+---------------------+--------------------------------+
|     0      | EAP-CREDS-Init      | Initialization Phase           |
|     1      | EAP-CREDS-ProtoFlow | Carries Provisioning Protocol  |
|            |                     | Messages                       |
|     2      | EAP-CREDS-Validate  | Validates newly installed      |
|            |                     | credentials                    |
|------------+---------------------+--------------------------------+

Table 2: EAP-CREDS Message Types

7.1. The EAP-CREDS-Init Message

The EAP-CREDS-Init message type is used in Phase One only of EAP-CREDS. The message flow is depicted in Section 5.3. This message supports the following TLVs: Version, ProvProto, CredsInfo, and IdProvider.

In this section we specify how these TLVs are used in the phase-one messages.

7.1.1. Version TLV usage

The Version TLV is used to convey the version of EAP-CREDS to be used for the current session.

The Server uses one or more Version TLVs in the EAP-Request/EAP-CREDS(Type=Init) message to provide the Peer with the list of EAP-CREDS versions supported. At minimum, the Server must include one Version TLV with the value of ’0x1’. If the Server detects multiple occurrences of this TLV in the reply from the Peer, an error shall be issued and the EAP-CREDS should abort.
The Peer, on the other hand, MUST include only one Version TLV in the EAP-Response/EAP-CREDS(Type=Init) message to indicate the version of EAP-CREDS that the client wants to use for the session. The Peer MUST pick from the list provided in the message from the server and, if no common version is supported by the client, then the client shall send an error message (i.e., an EAP-Response/EAP-CREDS(Type=Init) with at least one EAP-CREDS-Err TLV).

7.1.2. The ProvProto TLV usage

The ProvProto TLV is used to indicate which provisioning protocols are supported by the peer and the server.

The Server uses one or more ProvProto TLVs in the EAP-Request/EAP-CREDS(Type=Init) message to provide the Peer with the list of supported provisioning protocol. At minimum, the Server must include one ProvProto TLV with the Simple Provisioning Protocol value (SSP MUST be supported by EAP-CREDS servers). If the Server detects multiple occurrences of this TLV in the reply from the Peer, an error shall be issued and the EAP-CREDS should abort.

The Peer, on the other hand, MUST include only one ProvProto TLV in the EAP-Response/EAP-CREDS(Type=Init) message to indicate the protocol it intends to use in phase two. The Peer MUST pick from the list provided in the message from the server and, if no common protocol is supported by the client, then the client shall send an error message (i.e., an EAP-Response/EAP-CREDS(Type=Init) with at least one EAP-CREDS-Err TLV).

7.1.3. The CredsInfo TLV usage

The CredsInfo TLV is used by the peer to convey information to the EAP-CREDS server about the installed (if any) credentials for the network being accessed.

The Server does not use this TLV in its EAP-CREDS-Init messages.

The Peer, on the other hand, MUST include one CredsInfo TLV for each installed credentials (that the network is authorized to manage) in the EAP-Response/EAP-CREDS(Type=Init) message. If no credentials are available, yet, then the client SHALL NOT include this TLV in its EAP-CREDS-Init message.

7.1.4. The IdProvider TLV

The IdProvider TLV is used to convey the list of supported providers that can be used for managing credentials (e.g., a list of identity providers).
The Server uses one or more IdProvider TLVs in the EAP-Request/EAP-CREDS(Type=Init) message to provide the Peer with the list of supported credentials providers. The server can omit the set of TLVs in case a single provider is supported (or if the selection of the provider is done based on different factors – e.g., the authenticated credentials via the tunneling mechanism).

The Peer, on the other hand, MAY include only one IdProvider TLV in the EAP-Response/EAP-CREDS(Type=Init) message to indicate which provider it wants the Server to use. The Peer MUST pick from the list provided in the message from the server. If the client does not support any of the providers listed in the Server’s message or if no selection is provided and the Peer requires a specific provider, then an EAP-Response/EAP-CREDS(Type=Init) with an EAP-CREDS-Err TLV MUST be sent to the server as a response.

7.2. The EAP-CREDS-ProtoFlow Message

The EAP-CREDS-ProtoFlow message type is used in Phase Two only of EAP-CREDS. The message flow is depicted in Section 5.4. This message type supports the following TLVs: ProvProtoHeaders and ProvProtoData.

In this section we specify how these TLVs are used in the phase-one messages.

7.2.1. The ProvProtoHeader TLV

The ProvProtoHeader TLV is used to convey one or more headers (or extra information) that add relevant information for the provisioning protocol.

Both on the Peer and on the Server, this TLV should be used to, for example, provide HTTP headers that might be required for some provisioning protocols that do not properly abstract from the transport layer (e.g., ACME and EST are examples of this type of protocols).

7.2.2. The ProvProtoData TLV

The ProvProtoData TLV is used to transport the body of the messages for the provisioning protocol across the Peer and the Server.

Both on the Peer and on the Server, this TLV is used to carry the provisioning protocol’s messages. In particular, the protocols’ messages are encapsulated in this TLV without further re-encoding of the message. Only one instance of this TLV is allowed in EAP-CREDS-ProtoFlow messages.
7.3. The ('Validate') Message

8. Error Handling in EAP-CREDS

This section provides a description of the error handling by using the CREDS-Error-TLV in a CREDS message.

9. Fragmentation

Although EAP does not directly support handling of fragmented packets, EAP-CREDS requires that its messages are encapsulated via an outer method that MUST provide fragmentation support.

Because of the outer method requirements in particular, removing any support for fragmented messages in EAP-CREDS removes the duplication of packets (e.g., Acknowledgment Packets) sent across the Peer and the Server, thus resulting in a smaller number of exchanged messages.

10. Using EAP-CREDS with EAP-TEAP
11. Using EAP-CREDS with EAP-TTLS
12. EAP-CREDS and Simple Provisioning Protocol (SPP)

EAP-CREDS supports a Simple Provisioning Protocol (SPP) which comprises a series of messages that enables the management not only of certificates, but also of other types of credentials like username/password pairs, asymmetric keys, and symmetric keys.

EAP-CREDS/SSP defines the following flow of messages for requesting the provisioning of credentials.

13. IANA Considerations

This document uses a new EAP type, EAP-CREDS, whose value (TBD) MUST be allocated by IANA from the EAP TYPEs subregistry of the RADIUS registry. This section provides guidance to the Internet Assigned Numbers Authority (IANA) regarding registration of values related to the EAP-CREDS protocol, in accordance with [RFC8126].
The EAP Method Type number for EAP-CREDS needs to be assigned.

This document also requires IANA to create new registries as defined in the following subsections.

### 13.1. Provisioning Protocols

<table>
<thead>
<tr>
<th>Message Type</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Unspecified</td>
</tr>
<tr>
<td>1</td>
<td>Simple Provisioning Protocol (SPP)</td>
</tr>
<tr>
<td>2</td>
<td>Basic Certificate Management Protocol (CMP-S)</td>
</tr>
<tr>
<td>3</td>
<td>Full Certificate Management Protocol (CMP-F)</td>
</tr>
<tr>
<td>4</td>
<td>Enrollment over Secure Transport (EST)</td>
</tr>
<tr>
<td>5</td>
<td>Certificate Management over CMS (CMC)</td>
</tr>
<tr>
<td>6</td>
<td>Automatic Certificate Management Environment (ACME)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>49141 ... 65534</td>
<td>Vendor Specific</td>
</tr>
</tbody>
</table>

Table 3: EAP-CREDS Inner Protocol Identifiers

Assignment of new values for new cryptosuites MUST be done through IANA with "Specification Required" and "IESG Approval" as defined in [RFC8126].

### 13.2. Token Types

<table>
<thead>
<tr>
<th>Token Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Unspecified</td>
</tr>
<tr>
<td>1</td>
<td>JWT</td>
</tr>
<tr>
<td>2</td>
<td>Kerberos</td>
</tr>
<tr>
<td>3</td>
<td>OAuth</td>
</tr>
<tr>
<td>200...254</td>
<td>Vendor Specific</td>
</tr>
</tbody>
</table>

Table 4: Token Types

Assignment of new values for new Message Types MUST be done through IANA with "Expert Review" as defined in [RFC8126].
14. Security Considerations

Several security considerations need to be explicitly considered for the system administrators and application developers to understand the weaknesses of the overall architecture.

As part of the TLS negotiation, the server presents a certificate to the peer. The peer SHOULD verify the validity of the EAP server certificate and SHOULD also examine the EAP server name presented in the certificate in order to determine whether the EAP server can be trusted. When performing server certificate validation, implementations MUST provide support for the rules in [RFC5280] for validating certificates against a known trust anchor.

15. Acknowledgments

The authors would like to thank everybody who provided insightful comments and helped in the definition of the deployment considerations.

16. Normative References


Appendix A. EAP-CREDS Example Message Flow for Provisioning Standards

A.1. EAP-CREDS and CMP

Describe how to use EAP-CREDS with CMP.

A.2. EAP-CREDS and EST

Describe how to use EAP-CREDS with EST.

A.3. EAP-CREDS and CMS

Describe how to use EAP-CREDS with CMS.

A.4. EAP-CREDS and ACME

Describe how to use EAP-CREDS with ACME.

Author’s Address

Massimiliano Pala
CableLabs
858 Coal Creek Cir
Louisville, CO 80027
US

Email: m.pala@openca.org
URI: http://www.linkedin.com/in/mpala