Network Working Group
Internet-Draft
Updates: RFC7170 (if approved)
Intended status: Standards Track
Expires: 26 February 2022

E. Lear
O. Friel
N. Cam-Winget
Cisco Systems
D. Harkins
HP Enterprise

25 August 2021

TEAP Update and Extensions for Bootstrapping draft-lear-eap-teap-brski-06

Abstract

In certain environments, in order for a device to establish any layer three communications, it is necessary for that device to be properly credentialed. This is a relatively easy problem to solve when a device is associated with a human being and has both input and display functions. It is less easy when the human, input, and display functions are not present. To address this case, this memo specifies extensions to the Tunnel Extensible Authentication Protocol (TEAP) method that leverages Bootstrapping Remote Secure Key Infrastructures (BRSKI) in order to provide a credential to a device at layer two. The basis of this work is that a manufacturer will introduce the device and the local deployment through cryptographic means. In this sense the same trust model as BRSKI is used.

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 26 February 2022.

Table of Contents

1. Introduction

[I-D.ietf-anima-bootstrapping-keyinfra] (BRSKI) specifies a means to provision credentials to be used as credentials to operationally access networks. It was designed as a standalone means where some limited access to an IP network is already available. This is not always the case. For example, IEEE 802.11 networks generally require authentication prior to any form of address assignment. While it is possible to assign an IP address to a device on some form of an open network, or to accept some sort of default credential to establish initial IP connectivity, the steps that would then follow might well require that the device is placed on a new network, requiring reseting all layer three parameters.

A more natural approach in such cases is to more tightly bind the provisioning of credentials with the authentication mechanism. One such way to do this is to make use of the Extensible Authentication Protocol (EAP) [RFC3748] and the Tunnel Extensible Authentication Protocol (TEAP) method [RFC7170]. Thus we define new TEAP Type-Length-Value (TLV) objects that can be used to transport the BRSKI protocol messages within the context of a TEAP TLS tunnel.

[RFC7170] discusses the notion of provisioning peers. Several different mechanisms are available. Section 3.8 of that document acknowledges the concept of not initially authenticating the outer TLS session so that provisioning may occur. In addition, exchange of multiple TLV messages between client and EAP server permits multiple provisioning steps.

1.1. Terminology

The reader is presumed to be familiar with EAP terminology as stated in [RFC3748]. In addition, the following terms are commonly used in this document.

* BRSKI: Bootstrapping Remote Secure Key Infrastructures, as defined in [I-D.ietf-anima-bootstrapping-keyinfra]. The term is also used to refer to the flow described in that document.

* EST: Enrollment over Secure Transport, as defined in [RFC7030].

* Voucher: a signed JSON object as defined in [RFC8366].

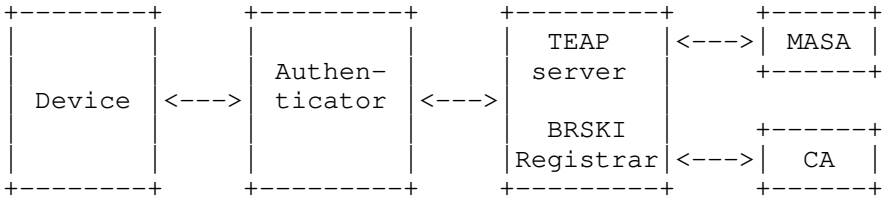
2. TEAP BRSKI Architecture

The TEAP BRSKI architecture is illustrated in Section 3. The device talks to the TEAP server via the Authenticator using any compliant transport such as [IEEE8021X]. The architecture illustrated shows an Authenticator distinct from the TEAP server. This is a deployment optimization and when so deployed the communication between Authenticator and TEAP server is a AAA protocol such as RADIUS or DIAMETER.

The architecture illustrated shows a co-located TEAP server and BRSKI registrar. Not only are these two functions co-located, they MUST be the same entity. This ensures that the entity identified in the device's voucher request (the TEAP server) is the same entity that signs the voucher request (the registrar).

The registrar communicates with the BRSKI MASA service for the purposes of getting signed vouchers.

The registrar also communicates with a Certificate Authority in order to issue LDevIDs. The architecture shows the registrar and CA as being two logically separate entities, however the CA may be integrated into the registrar. The device is not explicitly aware of whether the CA and registrar functions are integrated.



3. BRSKI Bootstrap and Enroll Operation

This section summarises the current BRSKI operation. The BRSKI flow assumes the device has an IDevID and has a manufacturer installed trust anchor that can be used to validate the BRSKI voucher. The BRSKI flow compromises serveral main steps from the perspective of the device:

* Step 1: Device discovers the registrar

* Step 2: Device establishes provisional TLS connection to registrar

* Step 3: Device sends voucher request message and receives signed voucher response

* Step 4: Device validates voucher and validates provisional TLS connection to registrar

* Step 5: Device downloads additional local domain CA information

* Step 6: Device downloads Certificate Signing Request (CSR) attributes

* Step 7: Device does a certificate enroll to obtain an LDevID

* Step 8: Device periodically reenrolls via EST to refresh its LDevID

Most of the operational steps require the device, and thus its internal state machine, to automatically complete the next step without being explicitly instructed to do so by the registrar. For example, the registrar does not explicitly tell the device to download additional local domain CA information, or to do an EST enroll to obtain an LDevID.

3.1. Discovery of Trusted MASA

BRSKI section 2.8 outlines how the Registrar discovers the correct MASA to connect with. BRSKI section 5.3 outlines how the Registrar can make policy decisions about which devices to trust.

Similar approaches are applicable for TEAP servers executing BRSKI. For example, the TEAP server may be configured with a list of trusted manufacturing CAs. During device bootstrap, only devices with an IDevID signed by a trusted manufacturing CA may be allowed to etablishes a TLS connection with the TEAP server, and the TEAP server could then extract the MASA URI from the device's IDevID.

3.2. Executing BRSKI in a TEAP Tunnel

This section outlines how the main BRSKI steps outlined above map to TEAP, and how BRSKI and enrollment can be accomplished inside a TEAP TLS tunnel. The following new TEAP TLVs are introduced:

* BRSKI-VoucherRequest

* BRSKI-Voucher

* CSR-Attributes

The following steps outline how the above BRSKI flow maps to TEAP.

* Step 1: Device discovers the registrar

When BRSKI is executed in a TEAP tunnel, the device exchanges BRSKI TLVs with the TEAP server. The discovery process for devices is therefore the standard wired or wireless LAN EAP server discovery process. The discovery processes outlined in section 4 of [I-D.ietf-anima-bootstrapping-keyinfra] are not required for initial discovery of the registrar.

* Step 2: Device establishes provisional TLS connection to registrar

The device establishes an outer TEAP tunnel with the TEAP server and does not validate the server certificate. The device presents its LDevID as its identity certificate if it has a valid LDevID, otherwise it presents its IDevID. The TEAP server validates the device's certificate using its implicit or explicit trust anchor database. If the device presents an IDevID it is verified against a database of trusted manufacturer certificates. Server policy may also be used to control which certificate the device is allowed present, as described in section {pki-certificate-authority-considerations}.

If the presented credential is sufficient to grant access, the TEAP server can return a TEAP Result TLV indicating success immediately. The device may still send a Request-Action TLV including a BRSKI-VoucherRequest TLV in response to the TEAP Result TLV if it does not have, but requires, provisioning of trust anchors for validating the TEAP server certificate. Note that no inner EAP method is required for this, only an exchange of TEAP TLVs.

[todo] Question: as the device wants the server to reply with a BRSKI-Voucher TLV, does it really send a Request-Action TLV containing a BRSKI-VoucherRequest TLV, or does it send a Request-Action TLV containing a BRSKI-Voucher TLV?? The TEAP draft is a bit ambiguous here. Normally, if one end sends a Request-Action including XXX-TLV, it means it wants the far end ot send an XXX-TLV...

[todo] Question: general TEAP protocol question: does the device have to send a Request-Action w/BRSKI-VoucherRequest or can it send a BRSKI-VoucherRequest on its own? I'm not clear on this.

If the TEAP server requires that the device execute a BRSKI flow, the server sends a Request-Action TLV that includes a BRSKI-VoucherRequest TLV. For example, if the device presented its IDevID but the TEAP server requires an LDevID.

[todo] Question: to nit pick, the server should send a Request-Action TLV including a PKCS#10 TLV to tell the client to enroll. How does the server really know that the client has the correct trust

established (as previously received by a BRSKI-Voucher)? If the client sends an IDevID, does server always send a Request-Action including both BRSKI-VoucherRequest and PKCS#10 TLVs? Whats the client behaviour? I assume client can spontaneously send BRSKI-VoucherRequest and/or PCSK#10 without being explicitly instructed to. Just need to get the language correct here.

The TEAP server may also require the device to reenroll, for example, if the device presented a valid LDevID that is very closed to expiration. The server may instruct a device to reenroll by sending a Request-Action TLV that includes a zero byte length PKCS#10 TLV.

* Step 3: Device sends voucher request message and receives signed voucher response

The device sends a BRSKI-RequestVoucher TLV to the TEAP server. The TEAP server forwards the RequestVoucher message to the MASA server, and the MASA server replies with a signed voucher. The TEAP server sends a BRSKI-Voucher TLV to the device.

If the MASA server does not issue a signed voucher, the TEAP server sends an EAP-Error TLV with a suitable error code to the device.

For wireless devices in particular, it is important that the MASA server only return a voucher for devices known to be associated with a particular registrar. In this sense, success indicates that the device is on the correct network, while failure indicates the device should try to provision itself within wireless networks (e.g, go to the next SSID).

* Step 4: Device validates voucher and validates provisional TLS connection to registrar

The device validates the signed voucher using its manufacturer installed trust anchor, and uses the CA information in the voucher to validate the TLS connection to the TEAP server.

If the device fails to validate the voucher, then it sends a TEAP-Error TLV indicating failiure to the TEAP server.

Similarly, if the device validates the voucher, but fails to validate the provisional TLS connection, then it sends a TEAP-Error TLV indicating failure to the TEAP server. Note that the outer TLS tunnel has already been established, thus allowing the client to send a TEAP-Error TLV to the server inside that tunnel to indicate that it failed to verify the provisionally accepted outer TLS tunnel server identity.

* Step 5: Device downloads additional local domain CA information

On completion of the BRSKI flow, the device SHOULD send a Trusted-Server-Root TLV to the TEAP server in order to discover additional local domain CAs. This is equivalent to section [todo] from [I-D.ietf-anima-bootstrapping-keyinfra].

* Step 6: Device downloads CSR attributes

No later than the completion of step 5, server MUST send a CSR-Attributes TLV to peer server in order to discover the correct fields to include when it enrolls to get an LDevID.

* Step 7: Device does a certificate enroll to obtain an LDevID

When executing the BRSKI flow inside a TEAP tunnel, the device does not directly leverage EST when doing its initial enroll. Instead, the device uses the existing TEAP PKCS#10 and PCKS#7 TEAP mechanisms.

Once the BRSKI flow is complete, the device can now send a PKCS#10 TLV to enroll and request an LDevID. If the TEAP server instructed the device to start the BRSKI flow via a Request-Action TLV that includes a BRSKI-RequestVoucher TLV, then the device MUST send a PKCS#10 in order to start the enroll process. The TEAP server will handle the PKCS#10 and ultimately return a PKCS#7 including an LDevID to the device.

If the TEAP server granted the device access on completion of the outer TEAP TLS tunnel in step 2 without sending a Request-Action TLV, the device does not have to send a PKCS#10 to enroll.

At this point, the device is said to be provisioned for local network access, and may authenticate in the future via 802.1X with its newly acquired credentials.

* Step 8: Device periodically reenrolls to refresh its LDevID

When a device's LDevID is close to expiration, there are two options for re-enrollment in order to obtain a fresh LDevID. As outlined in Step 2 above, the TEAP server may instruct the device to reenroll by sending a Request-Action TLV including a PKCS#10 TLV. If the TEAP server explicilty instructs the device to reenroll via these TLV exchange, then the device MUST send a PKCS#10 to reenroll and request a fresh LDevID.

However, the device SHOULD reenroll if it determines that its LDevID is close to expiration without waiting for explicit instruction from the TEAP server. There are two options to do this.

Option 1: The device reenrolls for a new LDevID directly with the EST CA outside the context of the 802.1X TEAP flow. The device uses the registrar discovery mechanisms oulined in [I-D.ietf-anima-bootstrapping-keyinfra] to discover the registrar and the device sends the EST reenroll messages to the discovered registrar endpoint. No new TEAP TLVs are defined to facilitate discover of the registrar or EST endpoints inside the context of the TEAP tunnel.

Option 2: When the device is performing a periodic 802.1X authentication using its current LDevID, it reenrolls for a new LDevID by sending a PKCS#10 TLV inside the TEAP TLS tunnel.

4. PKI Certificate Considerations

There are multiple noteworthy PKI certificate handling considerations. These include:

* PKI CA handling when establishing the TEAP tunnel
* PKI CA handling establishing trust using BRSKI
* IDevID and LDevID expiration times
* Specifying LDevID Subject and Subject Alternative Names
* PKCS#10 retry handling

These are described in more detail here.

4.1. TEAP Tunnel Establishment

Because this method establishes a client identity, if the peer has not been previously bootstrapped, or otherwise cannot successfully authenticate, it will use a generic identity string of teap-bootstrap@TBD1 as its network access identifier (NAI).

BRSKI section 5.3 outlines the policy decisions a Registrar may make when deciding whether to accept connections from clients. Similarly, the TEAP server operator may configure a set of trusted CAs for validating incoming TLS connections from clients. The operator may want to 'allow any device from a specific vendor', or from a set of vendors, to access the network. Network operators may do this by restricting network access to clients that have a certificate signed by one of a small set of trusted manufacturer/supplier CAs.

When the client sends its ClientHello to initiate TLS tunnel establishment, it is possible for the TEAP server to restrict the certificates that the client can use for tunnel establishment by including a list of CA distinguished names in the certificate_authorities field in the CertificateRequest message. The client should only continue with the handshake if it has a certificate signed by one of the indicated CAs.

In practice, network operators will likely want to onboard devices from a large number of device manufacturers, with each manufacturer using a different root CA when issuing IDevIDs. If the number of different manufacturer root CAs is large, this could result in very large TLS handshake messages. Therefore, the TEAP server may send a CertificateRequest message and not specify any certificate_authorities, thus allowing the client present a certificate signed by any authority in its Certificate message.

If the client has both an IDevID and an LDevID, the client should present the LDevID in preference to its IDevID, if allowed by server policy.

Once the client has sent its TLS Finished message, the TEAP server can make a policy decision, based on the CA used to sign the client's certificate, on whether to establish the outer TLS tunnel or not.

The TEAP server may delegate policy decisions to the MASA or CA function. For example, the TEAP server may declare EAP success and grant network access if the client presents a valid LDevID signed by a trusted domain CA. However, if the client presents an IDevID signed by a trusted manufacturer CA, the TEAP server may establish the TLS tunnel but not declare EAP success and grant network access until the client successfully completes a BRSKI Voucher exchange and PKCS#10/PKCS#7 exchange inside that tunnel.

It is recommended that the client validate the certificate presented by the server in the server's Certificate message, but this may not be possible for clients that have not yet provisioned appropriate trust anchors. If the client is in the provisioning phase and has not yet completed a BRSKI flow, it will not have trust anchors installed yet, and thus will not be able to validate the server's certificate. The client must however note the certificate presented by the server for (i) inclusion in the BRSKI-RequestVoucher TLV and for (ii) validation once the client has discovered the local domain trust anchors.

If the client does not present a suitable certificate to the server, the server MUST terminate the connection and fail the EAP request. If the TEAP server is unable to validate the client's certificate using its implicit or explicit trust anchor database it MUST fail the EAP request.

On establishment of the outer TLS tunnel, the TEAP server will make a policy decision on next steps. Possible policy decisions include:

* Option 1: Server grants client full network access and returns EAP-Success. This will typically happen when the client presents a valid LDevID. Network policy may grant client network access based on IDevID without requiring the device to enroll to obtain an LDevID.

* Option 2: Server requires that client perform a full BRSKI flow, and then enroll to get an LDevID. This will typically happen when the client presents a valid IDevID and network policy requires all clients to have LDevIDs. The server sends a Request-Action TLV that includes a BRSKI-RequestVoucher TLV to the client to instruct it to start the BRSKI flow.

* Option 3: Server requires that the client reenroll to obtain a new LDevID. This could happen when the client presents a valid LDevID that is very close to expiration time, or the server's policy requires an LDevID update. The server sends an Action-Request TLV including a PKCS#10 TLV to the client to instruct it to reenroll.

4.2. BRSKI Trust Establishment

If the server requires that client perform a full BRSKI flow, it sends a Request-Action TLV that includes a zero byte length BRSKI-RequestVoucher TLV to the client. The client sends a new BRSKI-RequestVoucher TLV to the server, which contains all data specified in [I-D.ietf-anima-bootstrapping-keyinfra] section 5.2. The client includes the server certificate it received in the server's Certificate message during outer TLS tunnel establishment in the proximity-registrar-cert field. The client signs the request using its IDevID.

The server includes all additional information as required by [I-D.ietf-anima-bootstrapping-keyinfra] section 5.4 and signs the request prior to forwarding to the MASA.

The MASA responds as per [I-D.ietf-anima-bootstrapping-keyinfra] section 5.5. The response may indicate failure and the server should react accordingly to failures by sending a failure response to the client, and failing the TEAP method.

If the MASA replies with a signed voucher and a successful result, the server then forwards this response to the client in a BRSKI-Voucher TLV.

When the client receives the signed voucher, it validates the signature using its built in trust anchor list, and extracts the pinned-domain-cert field. The client must use the CA included in the pinned-domain-cert to validate the certificate that was presented by the server when establishing the outer TLS tunnel. If this certificate validation fails, the client must fail the TEAP request and not connect to the network.

[TBD- based on client responses, the registrar sends a status update to the MASA]

4.3. Certificate Expiration Times

[IEEE8021AR] section 7.2.7.2 states:

> notAfter: The latest time a DevID is expected to be used. Devices possessing an IDevID are expected to operate indefinitely into the future and should use the value 99991231235959Z. Solutions verifying an IDevID are expected to accept this value indefinitely.

TEAP servers SHOULD follow the 802.1AR standard when validating IDevIDs.

TEAP servers SHOULD reject LDevIDs with expired certificates and SHOULD NOT allow clients to connect with recently expired LDevIDs. If a client presents a recently expired LDevID it SHOULD be forced to authenticate using its IDevID and then reenroll to obtain a valid LDevID.

4.4. LDevID Subject and Subject Alternative Names

BRSKI section 5.9.2 specifies that the pledge MUST send a CSR Attributes request to the registrar. The registrar MAY use this mechanism to instruct the pledge about the identities it should include in the CSR request it sends as part of enrollment. The registrar may use this mechanism to tell the pledge what Subject or Subject Alternative Name identity information to include in its CSR request. This can be useful if the Subject must have a specific value in order to complete enrollment with the CA.

4.5. PKCS#10 Retry Handling

They will be scenarios where the TEAP server is willing to handle a PKCS#10 request from a client and issue a certificate via a PKCS#7 response, however, the TEAP server is unable to immediately completely the request and needs to instruct the client to retry later after a specified time interval.

A new Retry-After TLV is defined that the TEAP server uses to specify a retry interval in seconds. New error codes are defined to handle these two alternate retry scenarios.

* The TEAP tunnel remains up: The client is instructed to resend the PKCS#10 request after a retry interval but inside the same TEAP tunnel. The TEAP server returns a Retry-After TLV to the client, and returns an Error TLV with a new code in the 1000-1999 range.

* The TEAP tunnel is torn down: The client is instructed to establish a new TEAP connection and TEAP tunnel after a retry interval, and resend the PKCS#10 request indside the new tunnel. The TEAP server returns a Retry-After TLV to the client, and returns an Error TLV with a new code in the 2000-2999 range.

5. Peer Identity

EAP [RFC3748] recommends that "the Identity Response be used primarily for routing purposes and selecting which EAP method to use". NAI [RFC7542] recommends ommitting the username part of an NAI in order to support username privacy, where appropriate.

A device that has not been bootstrapped at all SHOULD send an identity of teap-bootstrap@TBD1. Otherwise, a device SHOULD send its configured NAI.

The TEAP server may specify an NAI that it wishes the device to use. For example, the server may want a bootstrapped device to use an NAI of "abc123@example.com", or simply an NAI of "@example.com". This could be desirable in order to facilitate roaming scenarios. The server can do this by sending the device an NAI TLV inside the TEAP tunnel.

If the server specifies an NAI TLV, and the device handles the TLV, the device MAY use the specified NAI in all subsequent EAP authentication flows. If the device is not willing to handle the NAI TLV, it MUST reply with an Error TLV.

Authentication servers implementing this specification MAY reply with an Error TLV to any unrecognized NAI, or MAY attempt to bootstrap the device, regardless of the NAI. A device receving an Error from the server MAY attempt a new session without the NAI in order to bootstrap.

6. Channel and Crypto Binding

As the TEAP BRSKI flow does not define or require an inner EAP method, there is no explicit need for exchange of Channel-Binding TLVs between the device and the TEAP server.

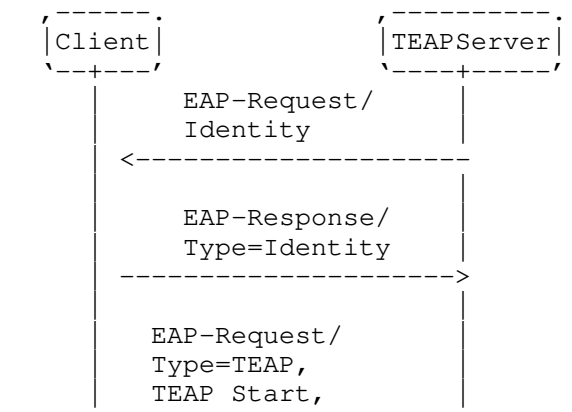
The TEAP BRSKI TLVs are expected to occur at the beginning of the TEAP Phase 2 and MUST occur before the final Crypto-Binding TLV. This draft does not exclude the possibility of having other EAP methods occur following the TEAP BRSKI TLVs and as such, the Crypto-Binding TLV process rules as defined in [RFC7170] apply.

7. Protocol Flows

This section outlines protocol flows that map to the three server policy options described in section Section 4.1. The protocol flows illustrate a TLS1.2 exchange. Pertinent notes are outlined in the protocol flows.

7.1. TEAP Server Grants Access

In this flow, the server grants access as server policy allows the client to access the network based on the identity certificate that the client presented. This means that either (i) the client has previously completed BRSKI and has presented a valid LDevID or (ii) the client presents an IDevID and network policy allows access based purely on IDevID.



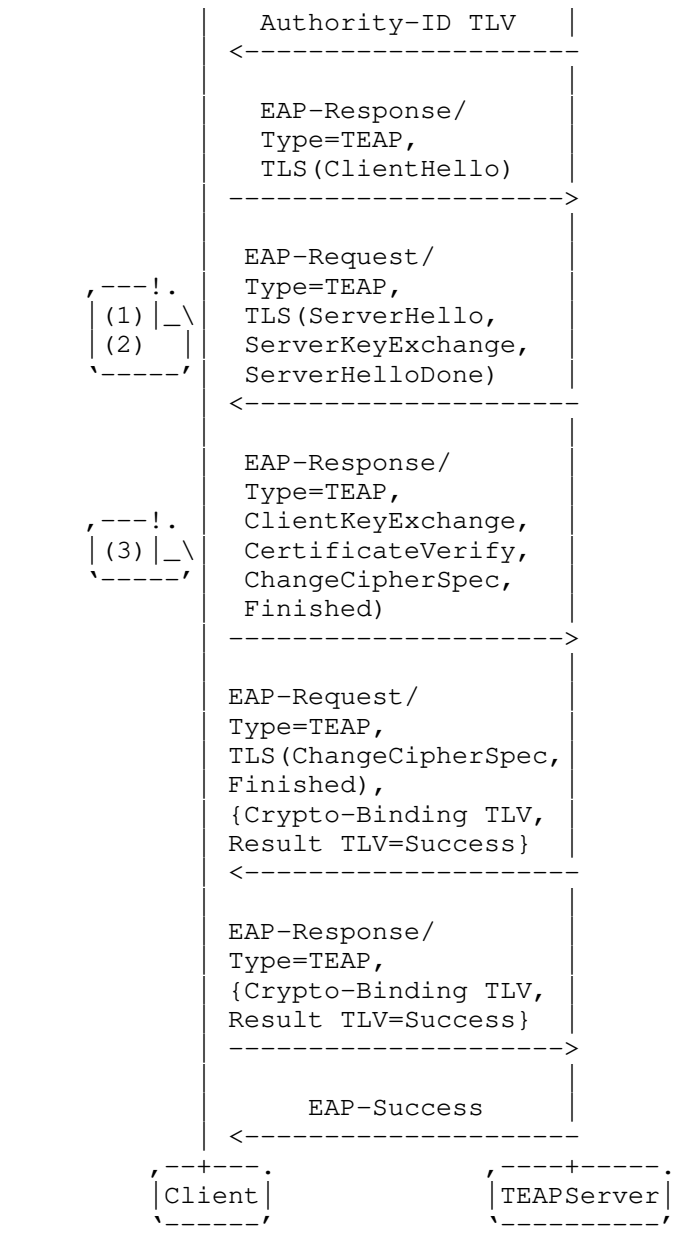


Figure 1: TEAP Server Grants Access

Notes:

(1) If the client has completed the BRSKI flow and has locally significant trust anchors, it must validate the Certificate received from the server. If the client has not yet completed the BRSKI flow, then it provisionally accepts the server Certificate and must validate it later once BRSKI is complete.

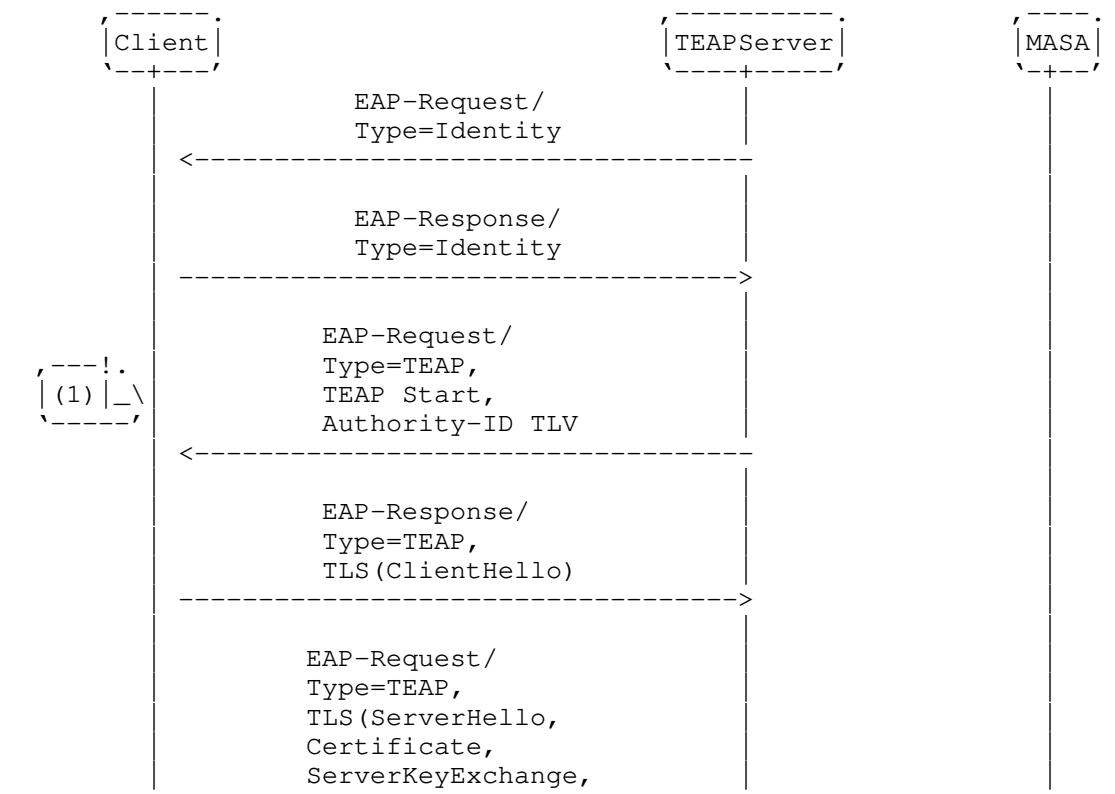
(2) The server may include certificate_authorities field in the CertificateRequest message in order to restrict the identity certificates that the device is allowed present.

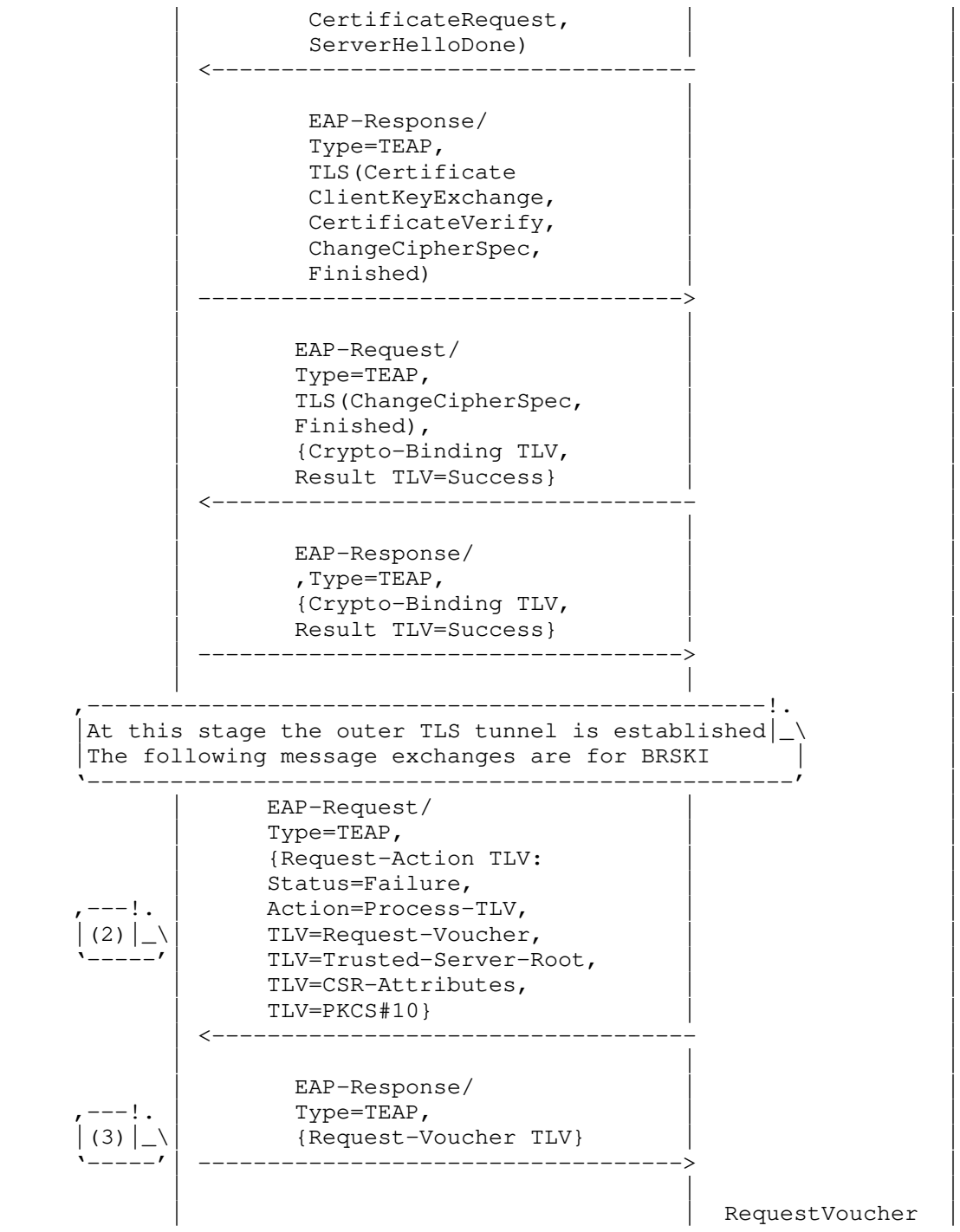
(3) The device will present its LDevID, if it has one, in preference to its IDevID, if allowed by server policy.

7.2. TEAP Server Instructs Client to Perform BRSKI Flow

In this two part flow, the server instructs the client to perform a BRSKI flow by exchanging TLVs once the outer TLS tunnel is established. After that, enrollment takes place.

In the first part of the flow, the MASA is depicted on the right.




CertificateRequest,
ServerHelloDone)
EAP-Response/
Type=TEAP,
TLS(Certificate
ClientKeyExchange,
CertificateVerify,
ChangeCipherSpec,
Finished)
EAP-Request/
Type=TEAP,
TLS(ChangeCipherSpec,
Finished),
{Crypto-Binding TLV,
Result TLV=Success}
EAP-Response/
,Type=TEAP,
{Crypto-Binding TLV,
Result TLV=Success}
At this stage the outer TLS tunnel is established
The following message exchanges are for BRSKI
EAP-Request/
Type=TEAP,
{Request-Action TLV:
Status=Failure,
Action=Process-TLV,
(2)
TLV=Request-Voucher,
TLV=Trusted-Server-Root,
TLV=CSR-Attributes,
TLV=PKCS#10}
EAP-Response/
Type=TEAP,
(3)
{Request-Voucher TLV}
RequestVoucher

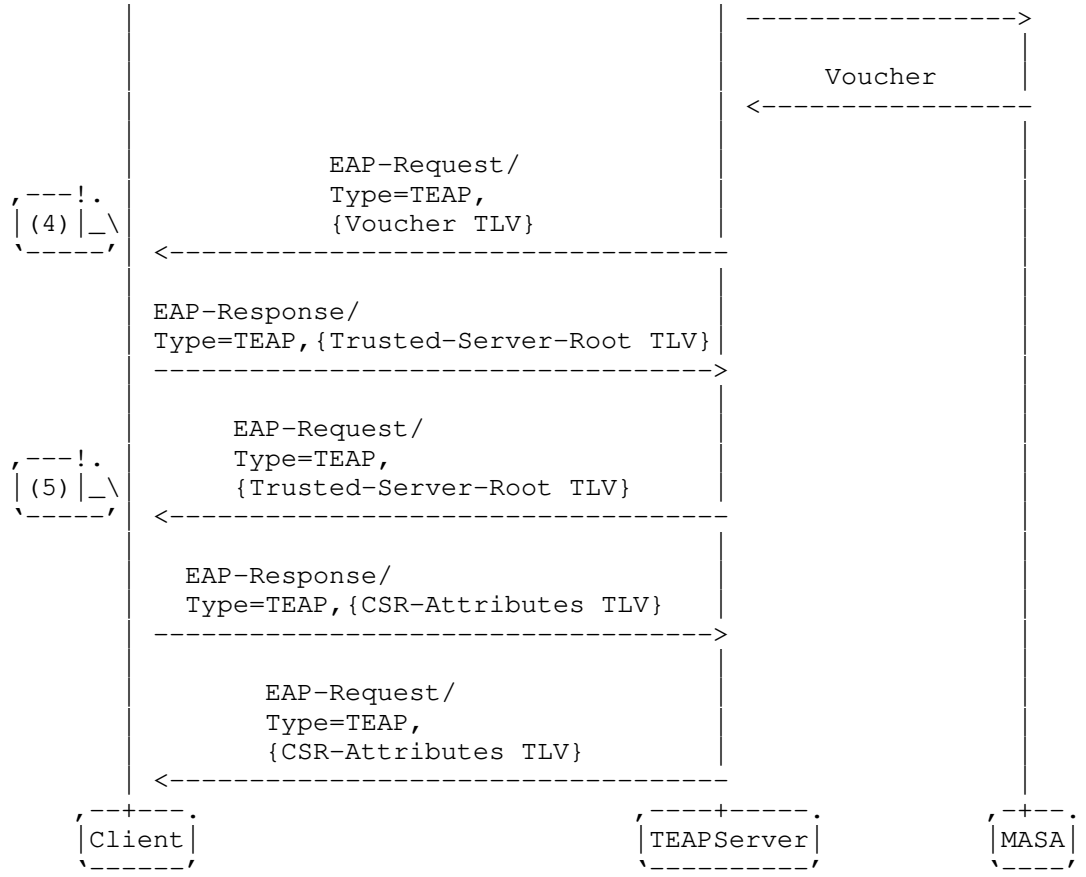


Figure 2: TEAP Server Instructs Client to Perform BRSKI Flow

The second part of the flow depicts the CA on the right.

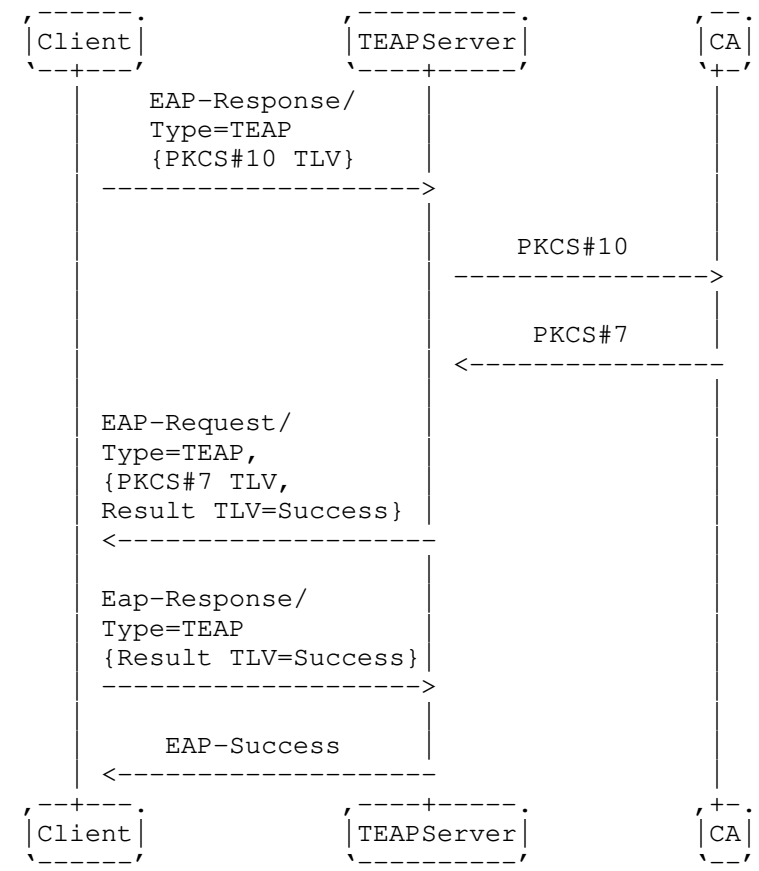


Figure 3: Enrollment after BRSKI Flow

Notes:

(1) If the client has not yet completed the BRSKI flow, then it provisionally accepts the server certificate and must validate it later once BRSKI is complete. The server validates the client certificate using its trust anchor database.

(2) The server instructs the client to start the BRSKI flow by sending a Request-Action TLV that includes a BRSKI-RequestVoucher TLV. The server also instructs the client to request trust anchors, to request CSR Attrites, and to initiate a PKCS certificate enrolment. As outlined in [RFC7170], the Request-Action TLV is sent after the Crypto-Binding TLV and Result TLV exchange.

(3) The client includes the certificate it received from the server in the RequestVoucher message.

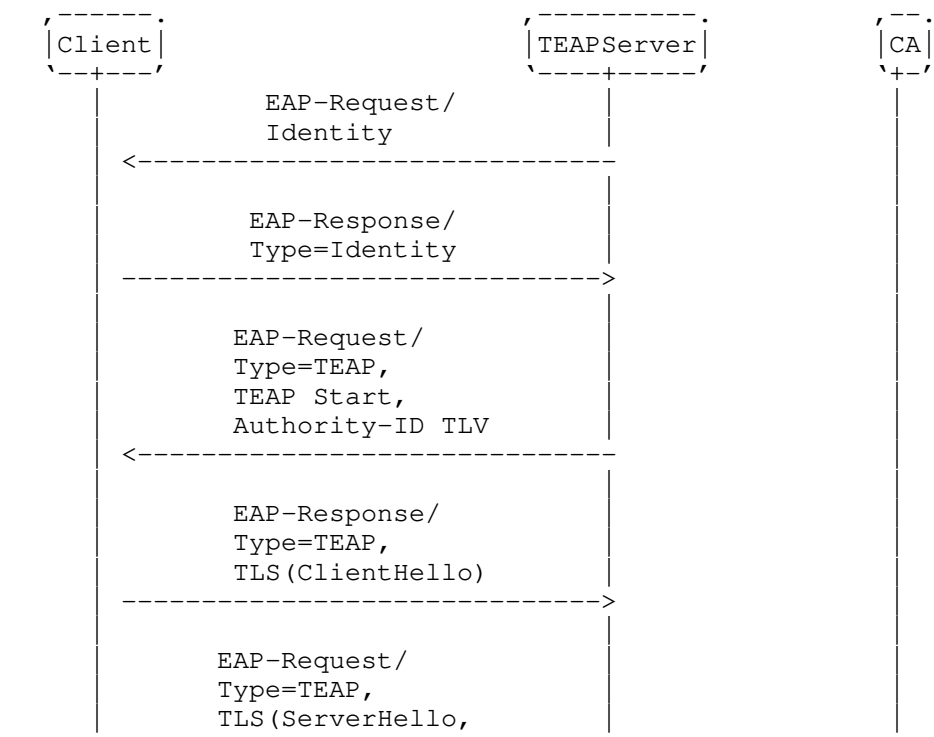
(4) Once the client receives and validates the voucher signed by the MASA, it must verify the certificate it previously received from the server.

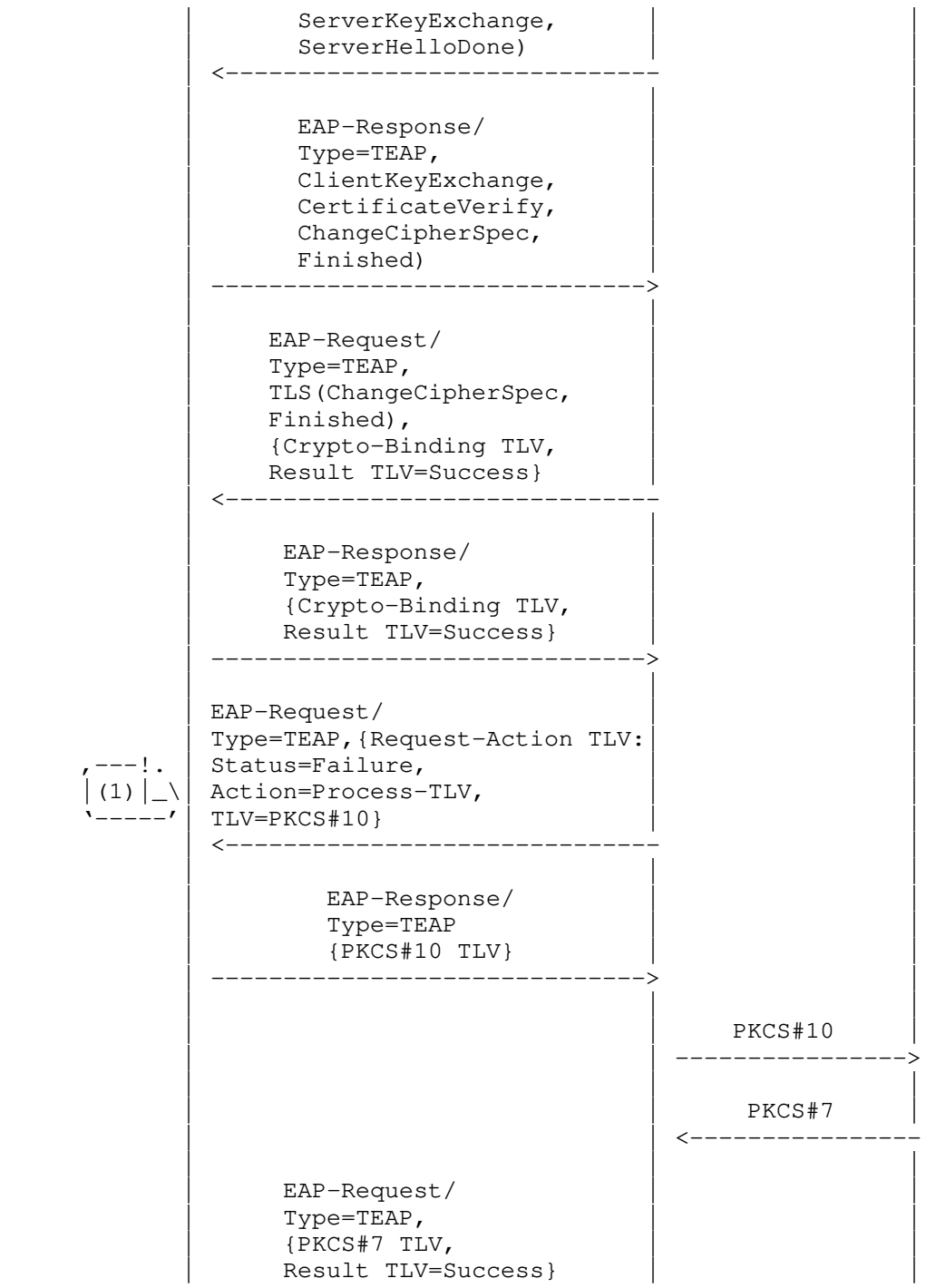
(5) As outlined in [RFC7170], the Trusted-Server-Root TLV is exchanged after the Crypto-Binding TLV exchange, and after the client has used the Voucher to authenticate the TEAP server identity. This is equivalent to section [todo] from [I-D.ietf-anima-bootstrapping-keyinfra].

(6) There is no need for an additional Crypto-Binding TLV exchange as there is no inner EAP method. All BRSKI exchanges are simply TLVs exchanged inside the outer TLS tunnel.

7.3. TEAP Server Instructs Client to Reenroll

In this flow, the server instructs the client to reenroll and get a new LDevID by exchanging TLVs once the outer TLS tunnel is established.




ServerKeyExchange,
ServerHelloDone)
EAP-Response/
Type=TEAP,
ClientKeyExchange,
CertificateVerify,
ChangeCipherSpec,
Finished)
EAP-Request/
Type=TEAP,
TLS(ChangeCipherSpec,
Finished),
{Crypto-Binding TLV,
Result TLV=Success}
EAP-Response/
Type=TEAP,
{Crypto-Binding TLV,
Result TLV=Success}
EAP-Request/
Type=TEAP,{Request-Action TLV:
Status=Failure,
Action=Process-TLV,
TLV=PKCS#10}
(1)
EAP-Response/
Type=TEAP
{PKCS#10 TLV}
PKCS#10
PKCS#7
EAP-Request/
Type=TEAP,
{PKCS#7 TLV,
Result TLV=Success}

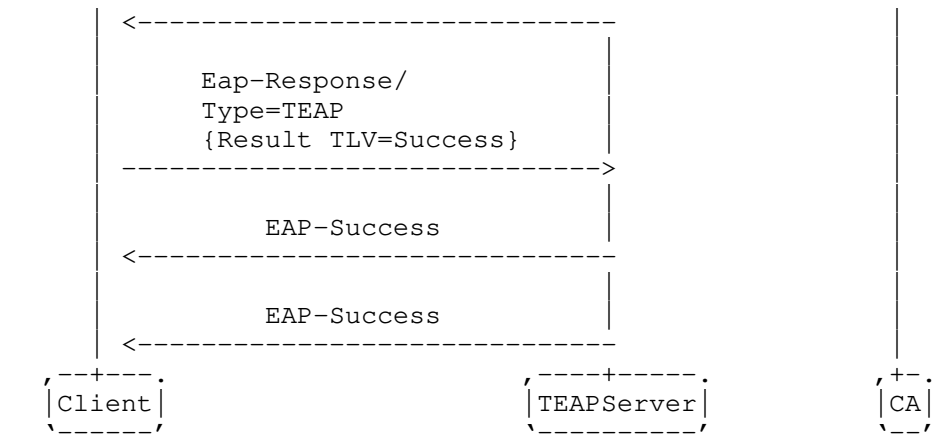


Figure 4: TEAP Server Instructs Client to Reenroll

(1) The server instructs the client to reenroll by sending a Request-Action TLV that includes a PKCS#10 TLV.

7.4. Out of Band Reenroll

This section shows how the device does a reenroll to refresh its LDEvID directly against the registrar outside the context of the TEAP tunnel.

8. TEAP TLV Formats

8.1. New TLVs

This document defines 5 new TEAP TLVs. The following table indicates whether the TLVs can be included in Request messages from TEAP server to device, or Response messages from device to TEAP server.

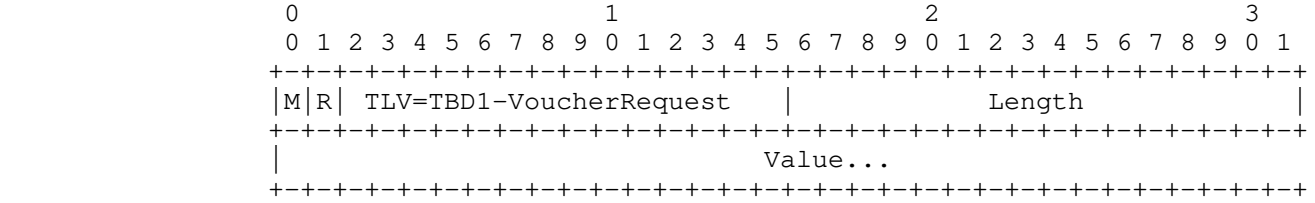
TLV	Message
BRSKI-VoucherRequest	Response
BRSKI-Voucher	Request
CSR-Attributes	Response
Retry-After	Response
NAI-Identity	Request

These new TLVs are detailed in this section.

8.1.1. BRSKI-RequestVoucher TLV

This TLV is used by the server as part of a Request-Action TLV to request from the peer that it initiate a voucher request. When used in this fashion, the length of this TLV will be set to zero. The Status field of the Request-Action TLV MUST be set to Failure.

It is also used by the peer to initiate the voucher request. When used in this fashion, the length of the TLV will be set to that of the voucher request, as encoded and described in Section 3.3 in [I-D.ietf-anima-bootstrapping-keyinfra].



The M and R bits are always expected to be set to 0.

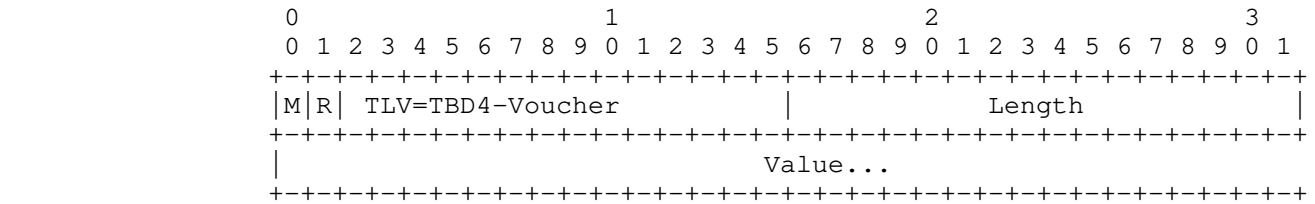
The server is expected to forward the voucher request to the MASA, and then return a voucher in a BRSKI-Voucher TLV as described below. If it is unable to do so, it returns an TEAP Error TLV with one of the defined errors or the following:

TBD2-MASA-Notavailable MASA unavailable
TBD3-MASA-Refused MASA refuses to sign the voucher

The peer terminates the TEAP connection, but may retry at some later point. The backoff mechanism for such retries should be appropriate for the device. Retries MUST occur no more frequently than once every two (TBD) minutes.

8.1.2. BRSKI-Voucher TLV

This TLV is transmitted from the server to the peer. It contains a signed voucher, as describe in [RFC8366].



Upon receiving this TLV the peer will validate the signature of the voucher, using its pre-installed manufacturer trust anchor (LDevID). It MUST also validate the certificate used by the server to establish the TLS connection.

If successful, it installs the new trust anchor contained in the voucher.

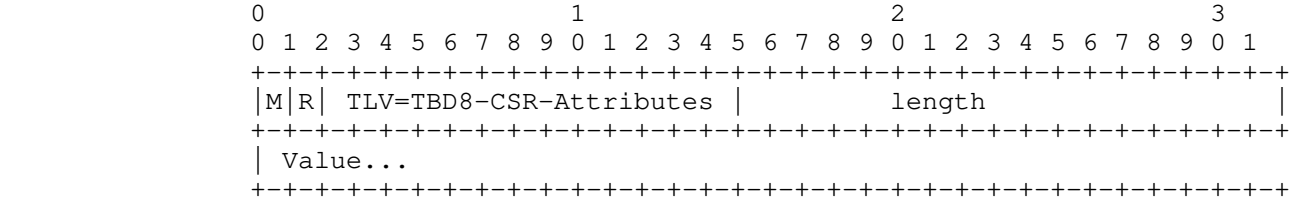
Otherwise, the peer transmits an TEAP error TLV with one of the following error messages:

TBD5-Invalid-Signature	The signature on the voucher is invalid
TBD6-Invalid-Voucher	The form or content of the voucher is invalid
TBD7-Invalid-TLS-Signer	The certificate used for the TLS connection could not be validated.

8.1.3. CSR-Attributes TLV

The server SHALL transmit this TLV to the peer, either along with the BRSKI-Voucher TLV or at any time earlier in a communication. The peer shall include attributes required by the server in any following CSR. The value of this TLV is the base64 encoding described in Section 4.5.2 of [RFC7030].

The TEAP server MAY use this TLV to specify the subject identity information to include in Subject or Subjecet Alternate Name fields in any following CSR.

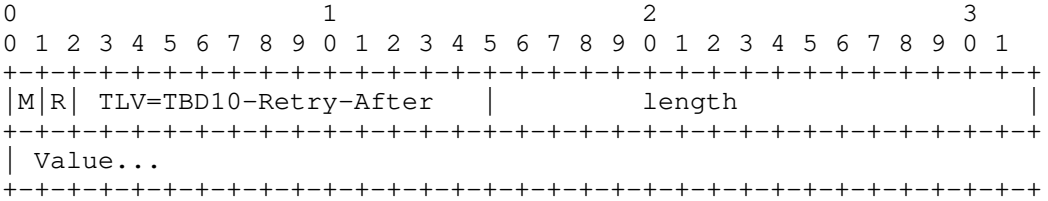


Again, the M and R values are set to 0. In the case where the client is unable to provide the requested attributes, an TEAP-Error is returned as follows:

TBD9-CSR-Attribute-Fail Unable to supply the requested attributes.

8.1.4. Retry-After TLV

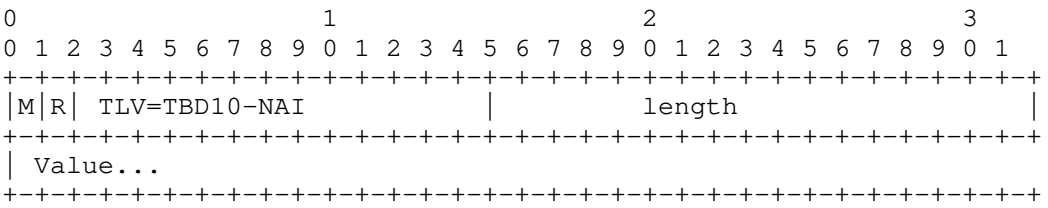
The server MUST transmit this TLV to the peer when repling to a PKCS#10 TLV request from the peer where the server is willing to fulfill the request and issue a certificate via a PKCS#7 response, but is unable to fulfill the request immediately. This TLV is used to tell the peer the mimimum lenght of time it MUST wait before resending the PKCS#10 request. The value of this TLV is the time in seconds that the peer MUST wait before resending the PKCS#10 request. The peer MUST resend the exact same PKCS#10 request.



Again, the M and R values are set to 0.

8.1.5. NAI TLV

The server may use this TLV to provision a realm-specific NAI on the device.



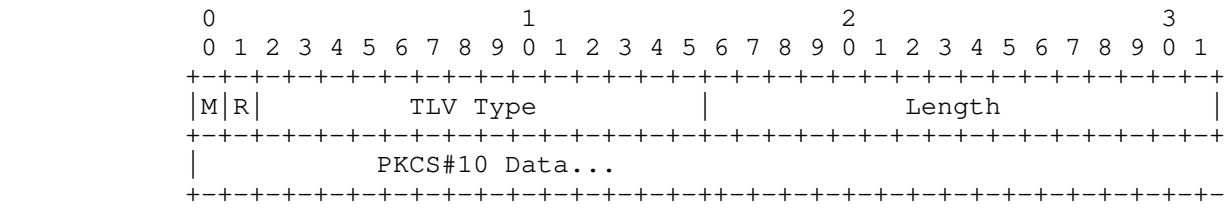
Again, the M and R values are set to 0.

8.2. Existing TEAP TLV Specifications

This section documents allowed usage of existing TEAP TLVs. The definition of the TLV is not changed, however clarifications on allowed values for the TLV fields is documented.

8.2.1. PKCS#10 TLV

[RFC7170] defines the PKCS#10 TLV as follows:



[RFC7170] does not explicitly allow a Length value of zero.

A Length value of zero is allowed for this TLV when the TEAP server sends a Request-Action TLV with a child PKCS#10 TLV to the client. In this scenario, there is no PKCS#10 Data included in the TLV. Clients MUST NOT send a zero length PKCS#10 TLV to the server.

8.3. TLV Rules

BRSKI TLVs can only be transported inside the TLS tunnel. The following table provides a guide to which TLVs may be encapsulated in which kind of packets, and in what quantity. The messages are as follows: Request is a TEAP Request, Response is a TEAP Response, Success is a message containing a successful Result TLV, and Failure is a message containing a failed Result TLV.

The following define the meaning of the table entries in the sections below:

0 This TLV MUST NOT be present in the message.

0+ Zero or more instances of this TLV MAY be present in the message.

0-1 Zero or one instance of this TLV MAY be present in the message.

1 Exactly one instance of this TLV MUST be present in the message.

Request Response Success Failure TLVs 0 0-1 0 0 BRSKI-VoucherRequest 0-1 0 0 0 BRSKI-Voucher 0 0-1 0 0 CSR-Attributes

9. Fragmentation

TEAP is expected to provide fragmentation support. Thus EAP-TEAP-BRSKI does not specifically provide any, as it is only expected to be used as an inner method to TEAP.

10. IANA Considerations

The IANA is requested to add entries into the following tables:

The following new TEAP TLVs are defined:

TBD1-VoucherRequest	Described in this document.
TBD4-Voucher	Described in this document.
TBD8-CSR-Attributes	Described in this document.
TBD10-Retry-After	Described in this document.

The following TEAP Error Codes are defined, with their meanings listed here and in previous sections:

TBD2-MASA-Notavailable	MASA unavailable
TBD3-MASA-Refused	MASA refuses to sign the voucher
TBD5-Invalid-Signature	The signature on the voucher is invalid
TBD6-Invalid-Voucher	The form or content of the voucher is invalid
TBD7-Invalid-TLS-Signer	The certificate used for the TLS connection could not be validated.
TBD9-CSR-Attribute-Fail	Unable to supply the requested attributes.
TBD11-Retry-PKCS#10	Retry PKCS#10 Request (1000 range code)
TBD12-Retry-PKCS#10	Retry PKCS#10 Request (2000 range code)
TBD13-NAI-Rejected	The device will not use the indicated NAI (1000 range code)

[[TODO: is there a registry of NAIs that map to TEAP methods? e.g. @eap-teap.net is reserved to indicate the peer wants to use TEAP method]]

11. Security Considerations

BRSKI [I-D.ietf-anima-bootstrapping-keyinfra] provides a zero touch way for devices to enroll in a certification authority (CA). It assumes the device has IP connectivity. For networks that will not grant IP connectivitiy before authenticating (with a local credential) this poses a Catch-22- can't get on the network without a credential and can't get a credential without getting on the network.

This protocol provides a way for BRSKI to be in an EAP method which allows the BRSKI conversation to happen as part of EAP authentication and prior to obtaining IP connectivity.

The security considerations of [I-D.ietf-anima-bootstrapping-keyinfra] apply to this protocol. Running BRSKI through EAP introduces some additional areas of concern though.

11.1. Issues with Provisionally Authenticated TEAP

This protocol establishes an unauthenticated TLS connection and passes data through it. Provided that the only messages passed in this state are self-protected BRSKI messages this does not present a problem. Passing any other messages or TLVs prior to authentication of the provisional TLS connection could potentially introduce security issues.

While the TLS connection is unauthenticated, it must still be validated to the fullest extent possible. It is critical that the device and the TEAP server perform all steps in TLS- checking the validity of the presented certificate, validating the signature using the public key of the certificate, etc- except ensuring the trust of the presented certificate.

11.2. Attack Against Discovery

The device discovery technique specified in this protocol is the standard EAP server discovery process. Since it is trivial to set up an 802.11 wireless access point and advertise any network, an attacker can impersonate a legitimate wireless network and attract unprovisioned pledges. Given that an unprovisioned device will not know the legitimate network to connect to, it will probably attempt the first network it finds, making the attack that much easier. This allows for a "rogue registrar" to provision and take control of the device.

If the MASA verifies ownership prior to issuance of a voucher, this attack can be thwarted. But if the MASA is in reduced security mode and does not verify ownership this attack cannot be prevented. Registrars SHOULD use the audit log of a MASA when deploying newly purchased equipment in order to mitigate this attack.

Another way to mitigate this attack is through normal "rogue AP" detection and prevention.

11.3. TEAP Server as Registration Authority

If the TEAP server is logically separate from the Certification Authority (CA) (see Section 2) it will be acting as a Registration Authority (RA) when it obtains the PKCS#10 TLV and replies with a PKCS#7 TLV (see [RFC7170], Sections 4.2.16 and 4.2.17, respectively). The assurance a RA makes to a CA is that the public key in the presented CSR is bound to an authenticated identity in way that will assure non-repudiation.

To make such an assurance, the TEAP server MUST authenticate the provisional TLS connection with the device by validating the voucher response received from the MASA. In addition, it is RECOMMENDED that the TEAP server indicate that proof-of-possession (see [RFC7170], Section 3.8.2) is required by including the challengePassword OID in the CSR Attributes TLV.

11.4. Trust of Registrar

The device accepts a trusted server (CA) certificate and installs it in its trust anchor database during step 5 (see Section 3.2). This can happen only after the provisional TLS connection has been authenticated using the voucher and the Crypto-Binding TLV has been validated.

12. Acknowledgments

The authors would like to thank Brian Weis for his assistance, and Alan Dakok for improving language consistency. In addition, with ruthlessly "borrowed" the concept around NAI handling from Tuomas Aura and Mohit Sethi.

13. References

13.1. Normative References

[I-D.ietf-anima-bootstrapping-keyinfra]
Pritikin, M., Richardson, M. C., Eckert, T., Behringer, M. H., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructure (BRSKI)", Work in Progress, Internet-Draft, draft-ietf-anima-bootstrapping-keyinfra-45, 11 November 2020, <https://www.ietf.org/archive/id/draft-ietf-anima-bootstrapping-keyinfra-45.txt>.

[IEEE8021AR]
Institute for Electrical and Electronics Engineers, "Secure Device Identity", 1998.

[IEEE8021X]
Institute for Electrical and Electronics Engineers, "IEEE Standard for Local and metropolitan area networks--Port-Based Network Access Control", 2010.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, Ed., "Extensible Authentication Protocol (EAP)", RFC 3748, DOI 10.17487/RFC3748, June 2004, <https://www.rfc-editor.org/info/rfc3748>.

[RFC7030] Pritikin, M., Ed., Yee, P., Ed., and D. Harkins, Ed., "Enrollment over Secure Transport", RFC 7030, DOI 10.17487/RFC7030, October 2013, <https://www.rfc-editor.org/info/rfc7030>.

[RFC7170] Zhou, H., Cam-Winget, N., Salowey, J., and S. Hanna, "Tunnel Extensible Authentication Protocol (TEAP) Version 1", RFC 7170, DOI 10.17487/RFC7170, May 2014, <https://www.rfc-editor.org/info/rfc7170>.

[RFC8366] Watsen, K., Richardson, M., Pritikin, M., and T. Eckert, "A Voucher Artifact for Bootstrapping Protocols", RFC 8366, DOI 10.17487/RFC8366, May 2018, <https://www.rfc-editor.org/info/rfc8366>.

13.2. Informative References

[RFC7542] DeKok, A., "The Network Access Identifier", RFC 7542, DOI 10.17487/RFC7542, May 2015, <https://www.rfc-editor.org/info/rfc7542>.

Appendix A. Changes from Earlier Versions

Draft -06: * nothing more than version bump

Draft -03: * Merge EAP server and Registrar * Security considerations * References improvements * Add Dan Harkins as co-author

Draft -02: * Flow corrections

Draft -01: * Add packet descriptions, IANA considerations, smooth out language.

Draft -00:

* Initial revision

Authors' Addresses

Eliot Lear
Cisco Systems
Richtistrasse 7
CH-8304 Wallisellen
Switzerland

Phone: +41 44 878 9200
Email: lear@cisco.com

Owen Friel
Cisco Systems
170 W. Tasman Dr.
San Jose, CA, 95134
United States

Email: ofriel@cisco.com

Nancy Cam-Winget
Cisco Systems
170 W. Tasman Dr.
San Jose, CA, 95134
United States

Email: ncamwing@cisco.com

Dan Harkins
HP Enterprise
3333 Scott Boulevard
Santa Clara, CA, 95054
United States

Email: dharkins@arubanetworks.com