Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 12, 2021

O. Friel
R. Barnes
Cisco
T. Hollebeek
DigiCert
M. Richardson
Sandelman Software Works

October 09, 2020

ACME for Subdomains
draft-friel-acme-subdomains-03

Abstract

This document outlines how ACME can be used by a client to obtain a certificate for a subdomain identifier from a certification authority. The client has fulfilled a challenge against a parent domain but does not need to fulfil a challenge against the explicit subdomain as certificate policy allows issuance of the subdomain certificate without explicit subdomain ownership proof.

Table of Contents

1. Introduction 2
2. Terminology 2
3. ACME Workflow and Identifier Requirements 3
4. Open Items 4
5. ACME Issuance of Subdomain Certificates 5
5.1. Pre-Authorization 5
5.2. Illustrative Call Flow 6
5.3. newOrder and newAuthz Handling 7
5.4. Examples 8
6. Resource Enhancements 9
6.1. Authorization Object 9
6.2. Directory Object Metadata 9
7. IANA Considerations 9
7.1. Authorization Object Fields Registry 9
7.2. Directory Object Metadata Fields Registry 9
8. Security Considerations 10
8.1. ACME Server Policy Considerations 11
9. Informative References 11
Appendix A. CA Browser Forum Baseline Requirements Extracts . . 12
Authors' Addresses 12

1. Introduction

ACME [RFC8555] defines a protocol that a certification authority (CA) and an applicant can use to automate the process of domain name ownership validation and X.509v3 (PKIX) [RFC5280] certificate issuance. This document outlines how ACME can be used to issue subdomain certificates, without requiring the ACME client to explicitly fulfil an ownership challenge against the subdomain identifiers - the ACME client need only fulfil an ownership challenge against a parent domain identifier.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are defined in the CA/Browser Baseline Requirements [CAB] and are reproduced here:

- Base Domain Name: The portion of an applied-for FQDN that is the first domain name node left of a registry-controlled or public suffix plus the registry-controlled or public suffix (e.g. "example.co.uk" or "example.com"). For FQDNs where the right-most domain name node is a gTLD having ICANN Specification 13 in its registry agreement, the gTLD itself may be used as the Base Domain Name.

- Domain Name: The label assigned to a node in the Domain Name System

- Domain Namespace: The set of all possible Domain Names that are subordinate to a single node in the Domain Name System

The following terms are used in this document:

- CA: Certification Authority

- CSR: Certificate Signing Request

- FQDN: Fully Qualified Domain Name

- Parent Domain: a node in the Domain Name System that has a Domain Name

- Subdomain: a Domain Name that is in the Domain Namespace of a given Parent Domain

3. ACME Workflow and Identifier Requirements

A typical ACME workflow for issuance of certificates is as follows:

1. client POSTs a newOrder request that contains a set of "identifiers"

2. server replies with a set of "authorizations" and a "finalize" URI

3. client sends POST-as-GET requests to retrieve the "authorizations", with the downloaded "authorization" object(s) containing the "identifier" that the client must prove that they control

4. client proves control over the "identifier" in the "authorization" object by completing the specified challenge, for example, by publishing a DNS TXT record

5. client POSTs a CSR to the "finalize" API

6. server replies with an updated order object that includes a "certificate" URI

7. client sends POST-as-GET request to the "certificate" URI to download the certificate

ACME places the following restrictions on "identifiers":

- section 7.1.4: the only type of "identifier" defined by the ACME specification is a fully qualified domain name: "The only type of identifier defined by this specification is a fully qualified domain name (type: "dns"). The domain name MUST be encoded in the form in which it would appear in a certificate."

- Section 7.4: the "identifier" in the CSR request must match the "identifier" in the newOrder request: "The CSR MUST indicate the exact same set of requested identifiers as the initial newOrder request."

- Sections 8.3: the "identifier", or FQDN, in the "authorization" object must be used when fulfilling challenges via HTTP: "Construct a URL by populating the URL template ... where the domain field is set to the domain name being verified"

- Section 8.4: the "identifier", or FQDN, in the "authorization" object must be used when fulfilling challenges via DNS: "The client constructs the validation domain name by prepending the label "_acme-challenge" to the domain name being validated."

ACME does not mandate that the "identifier" in a newOrder request matches the "identifier" in "authorization" objects.

4. Open Items

1. Does the client need a mechanism to indicate that they want to authorize a parent domain and not the explicit subdomain identifier? Or a mechanism to indicate that they are happy to authorize against a choice of identifiers? E.g. for foo1.foo2.bar.example.com, should the client be able to specify anywhere from 1 to 4 identifiers they are willing to fulfil challenges for?

2. Does the server need a mechanism to provide a choice of identifiers to the client and let the client chose which challenge to fulfil? E.g. for foo1.foo2.bar.example.com, should the server be able to specify anywhere from 1 to 4 identifiers that the client can pick from to fulfil?

Both 1 and 2 would require changes to the JSON object definitions. For 1, each identifier in the newOrder or newAuthz requests would need a child array of alternative identifiers the client is willing to fulfil. For 2, the current order object contains a set of authorizations that must all be completed, the authorization object contains a single identifier that all challenges are against, so therefore its not possible for the server to give the client a choice of identifiers to pick from.

This document does not currently define how 1 or 2 could be accomplished. This document merely defines how a client can submit a newOrder / newAuthz for one identifier (e.g. foo1.foo2.bar.example.com), and the server to choose a parent identifier (e.g. example.com) that it requires challenge fulfilment on, and specify that identifier in the authorization object.

5. ACME Issuance of Subdomain Certificates

As noted in the previous section, ACME does not mandate that the "identifier" in a newOrder request matches the "identifier" in "authorization" objects. This means that the ACME specification does not preclude an ACME server processing newOrder requests and issuing certificates for a subdomain without requiring a challenge to be fulfilled against that explicit subdomain.

ACME server policy could allow issuance of certificates for a subdomain to a client where the client only has to fulfil an authorization challenge for a parent domain of that subdomain. This allows a flow where a client proves ownership of, for example, "example.org" and then successfully obtains a certificate for "sub.example.org".

ACME server policy is out of scope of this document, however some commentary is provided in Section 8.1.

5.1. Pre-Authorization

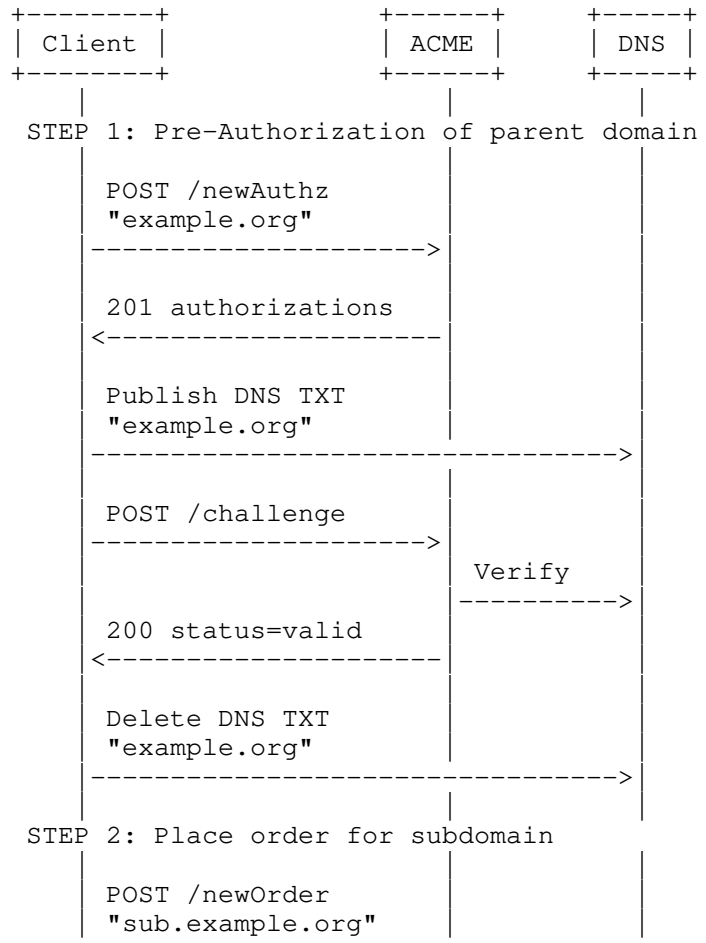
The standard ACME workflow has authorization objects created reactively in response to a certificate order. ACME also allows for pre-authorization, where clients obtain authorization for an identifier proactively, outside of the context of a specific issuance. This document allows for both workflows, and Section 5.3

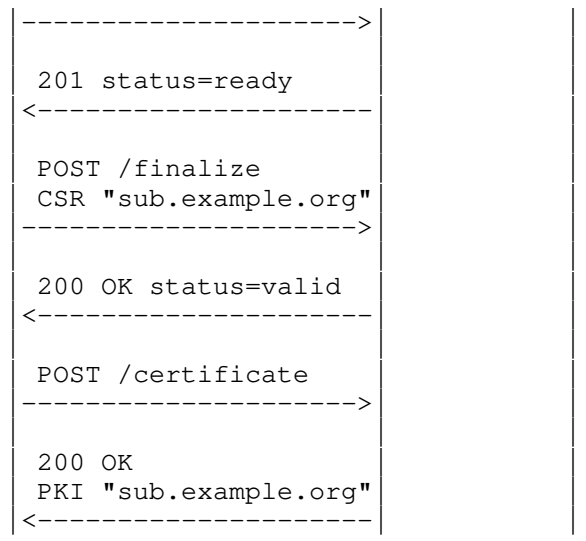
outlines how the ACME server handles newOrder and newAuthz requests for both workflows.

It may make sense to use the ACME pre-authorization flow for the subdomain use case, however, that is an operator implementation and deployment decision. With the ACME pre-authorization flow, the client could pre-authorize for the parent domain once, and then issue multiple newOrder requests for certificates for multiple subdomains.

5.2. Illustrative Call Flow

The call flow illustrated here uses the ACME pre-authorization flow. The call flow also illustrates the DNS-based proof of ownership mechanism, but the subdomain workflow is equally valid for HTTP based proof of ownership.





5.3. newOrder and newAuthz Handling

Servers may consider validation of a parent domain sufficient authorization for a subdomain. If a server has such a policy and a client is already authorized for the parent domain then:

- If the client submits a newAuthz request for a subdomain: The server MUST return status 200 (OK) response. The response body is the existing authorization object for the parent domain with status set to "valid".

- If the client submits a newOrder request for a subdomain: The server MUST return a 201 (Created) response. The response body is an order object with status set to "ready" and links to the unexpired authorizations against the parent domain.

If a server has such a policy and a client is not authorized for the parent domain then:

- If the client submits a newAuthz request for a subdomain: The server MUST return a status 201 (Created) response. The response body is a newly created authorization object for the parent domain with status set to "pending".

- If the client submits a newOrder request for a subdomain: The server MUST return a status 201 (Created) response. The response body is an order object with status set to "pending" and links to newly created authorizations objects against the parent domain.

5.4. Examples

In order to illustrate subdomain behaviour, let us assume that a client wishes to get certificates for subdomain identifiers "sub0.example.org", "sub1.example.org" and "sub2.example.org" under parent domain "example.org", and CA policy allows certificate issuance of these subdomain identifiers while only requiring the client to fulfil an ownership challenge for parent domain "example.org". Let us also assume that the client has not yet proven ownership of parent domain "example.org".

1. The client POSTs a newOrder request for identifier "sub0.example.org"

   The server creates an authorization object for identifier "example.org". The server replies with a 201 (Created) response. The response body is an order object with status set to "pending" and a link to newly created authorization object against the parent domain "example.org". Therefore, the server is instructing the client to fulfil a challenge against domain identifier "example.org" in order to obtain a certificate including identifier "sub0.example.org".

   The client completes the challenge for "example.org", POSTs a CSR to the order finalize URI, and downloads the certificate.

2. The client POSTs a newOrder request for identifier "sub1.example.org"

   The server replies with a 201 (Created) response. The response body is an order object with status set to "ready" and a link to the unexpired authorization against the parent domain "example.org".

   The client POSTs a CSR to the order finalize URI, and downloads the certificate.

3. The client POSTs a newAuthz request for identifier "sub2.example.org"

   The server replies with a 200 (OK) response. The response body is the previously created authorization object for "example.org" with status set to "valid".

6. Resource Enhancements

This document defines enhancements to the authorization and directory objects.

6.1. Authorization Object

If an ACME server allows issuance of certificates for subdomains of a parent domain, then the authorization object for the parent domain MUST include the optional "includeSubDomains" field, with a value of true.

The structure of an ACME authorization resource is enhanced to include the following optional field:

includeSubDomains (optional, boolean): This field MUST be present and true for authorizations where ACME server policy allows certificates to to be issued for subdomains of the identifier in the authorization object without explicit authorization of the subdomain

6.2. Directory Object Metadata

An ACME server can advertise support of issuance of subdomain certificates by including the boolean field "includeSubDomainsAuthorization" in its "ACME Directory Metadata Fields" registry. If not specified, then no default value is assumed. If an ACME server supports issuance of subdomain certificates, it can indicate this by including this field with a value of "true".

7. IANA Considerations

7.1. Authorization Object Fields Registry

The following field is added to the "ACME Authorization Object Fields" registry defined in ACME [RFC8555].

Field Name	Field Type	Configurable	Reference
includeSubDomains	boolean	false	RFC XXXX

7.2. Directory Object Metadata Fields Registry

The following field is added to the "ACME Directory Metadata Fields" registry defined in ACME [RFC8555].

Field Name	Field Type	Reference
includeSubDomainsAuthorization	boolean	RFC XXXX

8. Security Considerations

This document documents enhancements to ACME [RFC8555] that optimize the protocol flows for issuance of certificates for subdomains. The underlying goal of ACME for Subdomains remains the same as that of ACME: managing certificates that attest to identifier/key bindings for these subdomains. Thus, ACME for Subdomains has the same two security goals as ACME:

1. Only an entity that controls an identifier can get an authorization for that identifier

2. Once authorized, an account key's authorizations cannot be improperly used by another account

ACME for Subdomains makes no changes to:

- account or account key management
- ACME channel establishment, security mechanisms or threat model
- Validation channel establishment, security mechanisms or threat model

Therefore, all Security Considerations in ACME in the following areas are equally applicable to ACME for Subdomains:

- Threat Model
- Integrity of Authorizations
- Denial-of-Service Considerations
- Server-Side Request Forgery
- CA Policy Considerations

Some additional comments on ACME server opicy are given in the following section.

8.1. ACME Server Policy Considerations

The ACME for Subdomains and the ACME specifications do not mandate any specific ACME server or CA policies, or any specific use cases for issuance of certificates. For example, an ACME server could be used:

- to issue Web PKI certificates where the ACME server must comply with CA/Browser Forum [CAB] Baseline Requirements.
- as a Private CA for issuance of certificates within an organisation. The organisation could enforce whatever policies they desire on the ACME server.
- for issuance of IoT device certificates. There are currently no IoT device certificate policies that are generally enforced across the industry. Organsations issuing IoT device certificates can enforce whatever policies they desire on the ACME server.

ACME server policy could specify whether:

- issuance of subdomain certificates is allowed based on proof of ownership of a parent domain
- issuance of subdomain certificates is allowed, but only for a specific set of parent domains
- whether DNS based proof of ownership, or HTTP based proof of ownership, or both, are allowed

ACME server policy specification is exlpicitly out of scope of this document. For reference, extracts from CA/Browser Forum Baseline Requirements are given in the appendices.

9. Informative References

[CAB] CA/Browser Forum, "Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates", n.d., <https://cabforum.org/wp-content/uploads/CA-Browser-Forum-BR-1.7.1.pdf>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8555] Barnes, R., Hoffman-Andrews, J., McCarney, D., and J. Kasten, "Automatic Certificate Management Environment (ACME)", RFC 8555, DOI 10.17487/RFC8555, March 2019, <https://www.rfc-editor.org/info/rfc8555>.

Appendix A. CA Browser Forum Baseline Requirements Extracts

The CA/Browser Forum Baseline Requirements [CAB] allow issuance of subdomain certificates where authorization is only required for a parent domain. Baseline Requirements version 1.7.1 states:

- Section: "1.6.1 Definitions": Authorization Domain Name: The Domain Name used to obtain authorization for certificate issuance for a given FQDN. The CA may use the FQDN returned from a DNS CNAME lookup as the FQDN for the purposes of domain validation. If the FQDN contains a wildcard character, then the CA MUST remove all wildcard labels from the left most portion of requested FQDN. The CA may prune zero or more labels from left to right until encountering a Base Domain Name and may use any one of the intermediate values for the purpose of domain validation.

- Section: "3.2.2.4.6 Agreed-Upon Change to Website": Once the FQDN has been validated using this method, the CA MAY also issue Certificates for other FQDNs that end with all the labels of the validated FQDN. This method is suitable for validating Wildcard Domain Names.

- Section: "3.2.2.4.7 DNS Change": Once the FQDN has been validated using this method, the CA MAY also issue Certificates for other FQDNs that end with all the labels of the validated FQDN. This method is suitable for validating Wildcard Domain Names.

Authors' Addresses

Owen Friel
Cisco

Email: ofriel@cisco.com

Richard Barnes
Cisco

Email: rlb@ipv.sx

Tim Hollebeek
DigiCert

Email: tim.hollebeek@digicert.com

Michael Richardson
Sandelman Software Works

Email: mcr+ietf@sandelman.ca

Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 22, 2021

O. Friel
R. Barnes
Cisco
R. Shekh-Yusef
Auth0
M. Richardson
Sandelman Software Works

November 18, 2020

ACME Integrations
draft-ietf-acme-integrations-02

Abstract

This document outlines multiple advanced use cases and integrations that ACME facilitates without any modifications or enhancements required to the base ACME specification. The use cases include ACME integration with EST, BRSKI and TEAP.

Table of Contents

1. Introduction 2
2. Terminology 3
3. ACME Integration with EST 3
4. ACME Integration with BRSKI 6
5. ACME Integration with BRSKI Default Cloud Registrar 8
6. ACME Integration with TEAP 10
7. IANA Considerations 14
8. Security Considerations 14
8.1. Denial of Service against ACME infrastructure 15
9. Informative References 15
Authors' Addresses 17

1. Introduction

ACME [RFC8555] defines a protocol that a certificate authority (CA) and an applicant can use to automate the process of domain name ownership validation and X.509 (PKIX) certificate issuance. The protocol is rich and flexible and enables multiple use cases that are not immediately obvious from reading the specification. This document explicitly outlines multiple advanced ACME use cases including:

- ACME integration with EST [RFC7030]
- ACME integration with BRSKI [I-D.ietf-anima-bootstrapping-keyinfra]
- ACME integration with BRSKI Default Cloud Registrar [I-D.friel-anima-brski-cloud]
- ACME integration with TEAP [RFC7170] and TEAP Update and Extensions for Bootstrapping [I-D.lear-eap-teap-brski]

The integrations with EST, BRSKI (which is based upon EST), and TEAP enable automated certificate enrolment for devices.

ACME for subdomains [I-D.friel-acme-subdomains] outlines how ACME can be used by a client to obtain a certificate for a subdomain identifier from a certificate authority where the client has fulfilled a challenge against a parent domain, but does not need to fulfil a challenge against the explicit subdomain. This is a useful optimization when ACME is used to issue certificates for large

numbers of devices as it reduces the domain ownership proof traffic (DNS or HTTP) and ACME traffic overhead, but is not a necessary requirement.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are used in this document:

- BRSKI: Bootstrapping Remote Secure Key Infrastructures [I-D.ietf-anima-bootstrapping-keyinfra]
- CA: Certificate Authority
- CMC: Certificate Management over CMS
- CSR: Certificate Signing Request
- EST: Enrollment over Secure Transport [RFC7030]
- FQDN: Fully Qualified Domain Name
- RA: PKI Registration Authority
- TEAP: Tunneled Extensible Authentication Protocol [RFC7170]

3. ACME Integration with EST

EST [RFC7030] defines a mechanism for clients to enroll with a PKI Registration Authority by sending CMC messages over HTTP. EST section 1 states:

"Architecturally, the EST service is located between a Certification Authority (CA) and a client. It performs several functions traditionally allocated to the Registration Authority (RA) role in a PKI."

EST section 1.1 states that:

"For certificate issuing services, the EST CA is reached through the EST server; the CA could be logically "behind" the EST server or embedded within it."

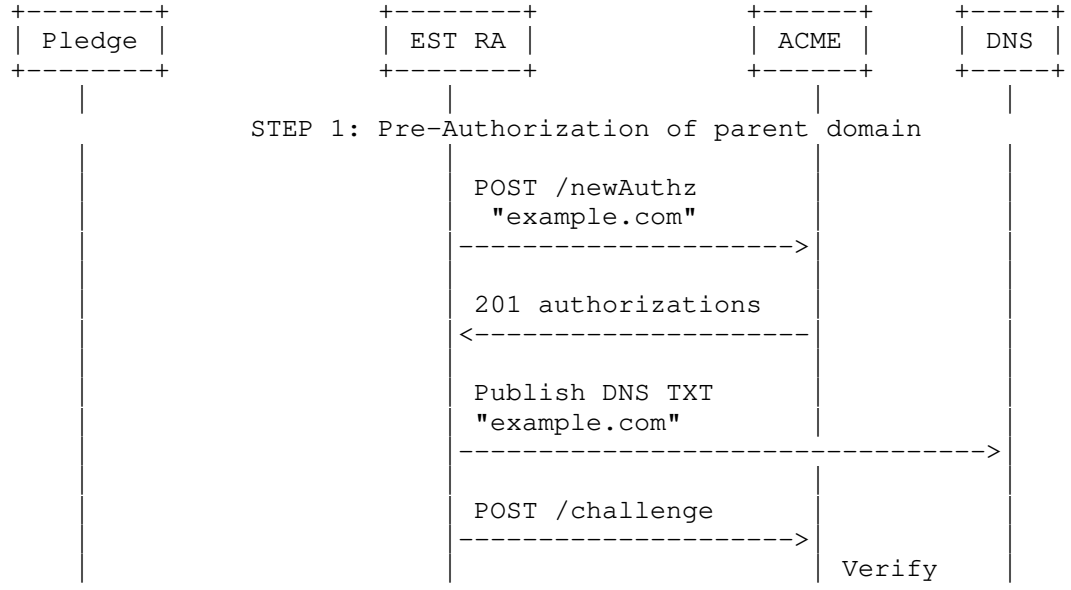
When the CA is logically "behind" the EST RA, EST does not specify how the RA communicates with the CA. EST section 1 states:

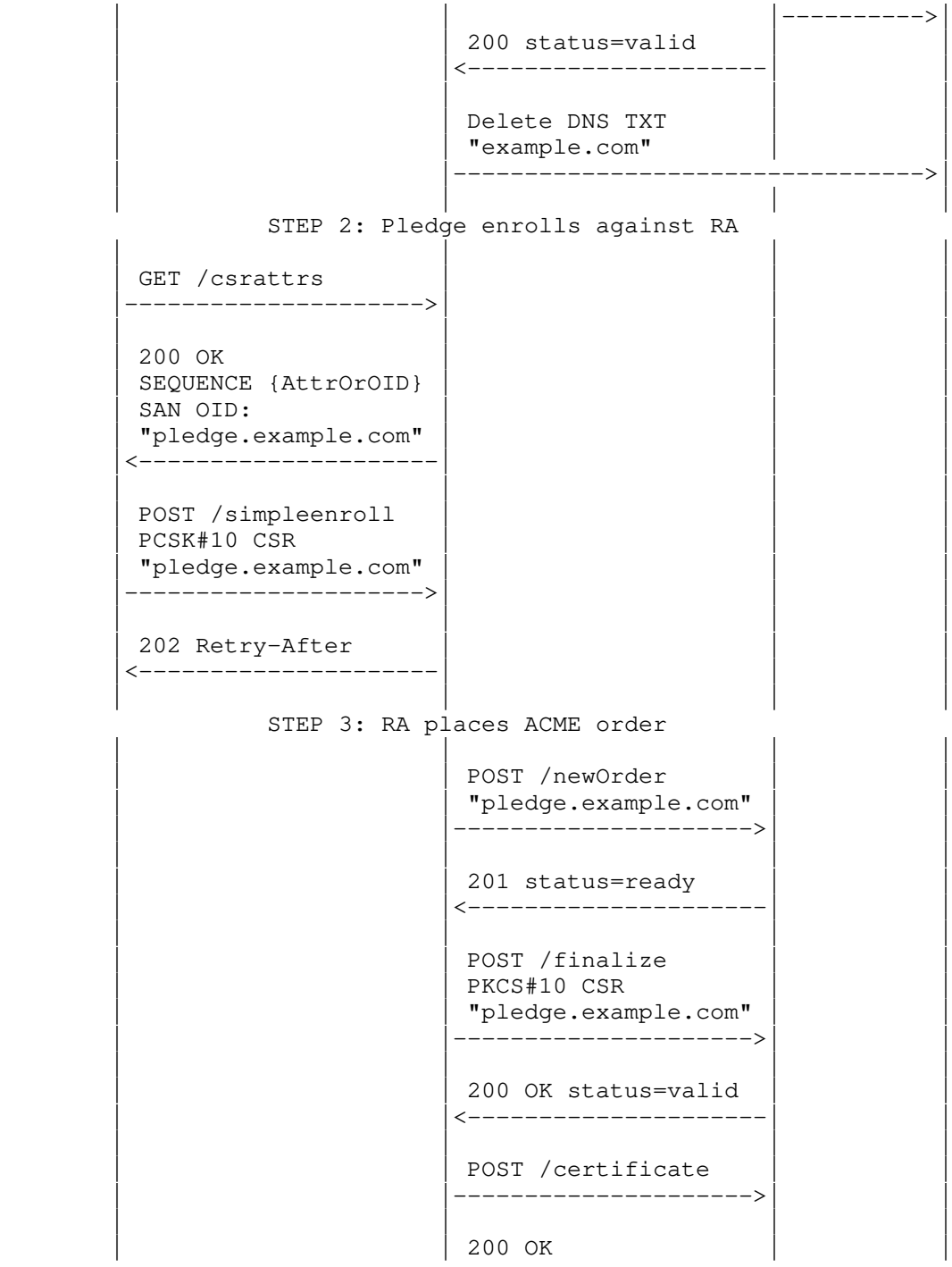
"The nature of communication between an EST server and a CA is not described in this document."

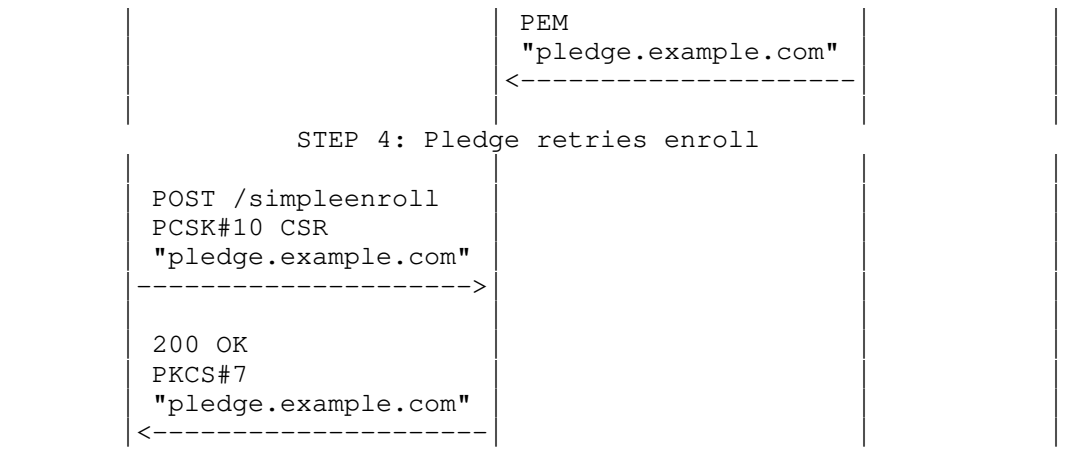
This section outlines how ACME could be used for communication between the EST RA and the CA. The example call flow leverages [I-D.friel-acme-subdomains] and shows the RA proving ownership of a parent domain, with individual client certificates being subdomains under that parent domain. This is an optimization that reduces DNS and ACME traffic overhead. The RA could of course prove ownership of every explicit client certificate identifier.

The call flow illustrates the client calling the EST /csrattrs API before calling the EST /simpleenroll API. This enables the EST server to indicate to the client what attributes it expects the client to include in the CSR request sent in the /simpleenroll API. For example, EST servers could use this mechanism to tell the client what fields to include in the CSR Subject and Subject Alternative Name fields.

The call flow illustrates the EST RA returning a 202 Retry-After response to the client's simpleenroll request. This is an optional step and may be necessary if the interactions between the RA and the ACME server take some time to complete. The exact details of when the RA returns a 202 Retry-After are implementation specific.




200 status=valid
Delete DNS TXT
"example.com"
STEP 2: Pledge enrolls against RA
GET /csrattrs
200 OK
SEQUENCE {AttrOrOID}
SAN OID:
"pledge.example.com"
POST /simpleenroll
PCSK#10 CSR
"pledge.example.com"
202 Retry-After
STEP 3: RA places ACME order
POST /newOrder
"pledge.example.com"
201 status=ready
POST /finalize
PKCS#10 CSR
"pledge.example.com"
200 OK status=valid
POST /certificate
200 OK



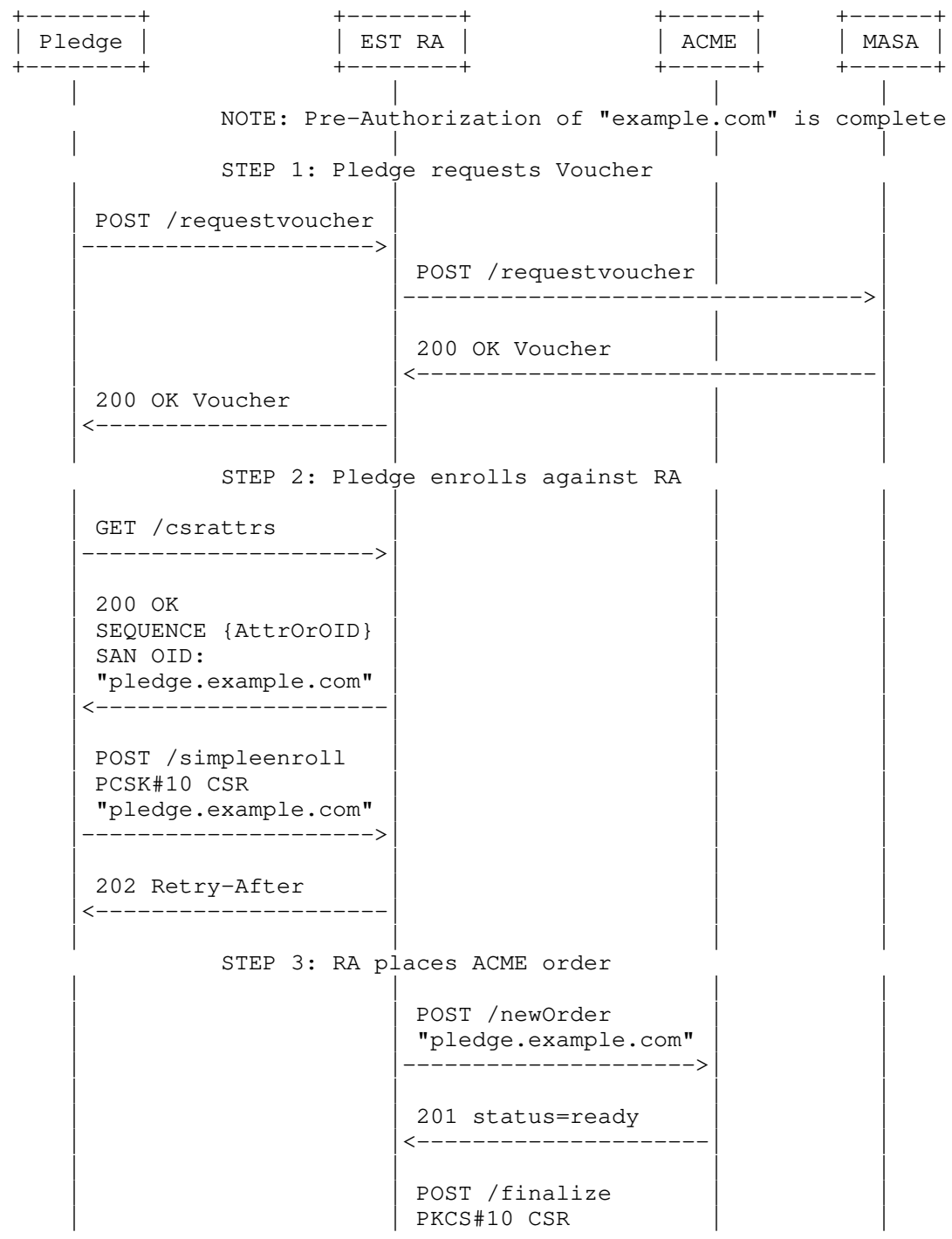
4. ACME Integration with BRSKI

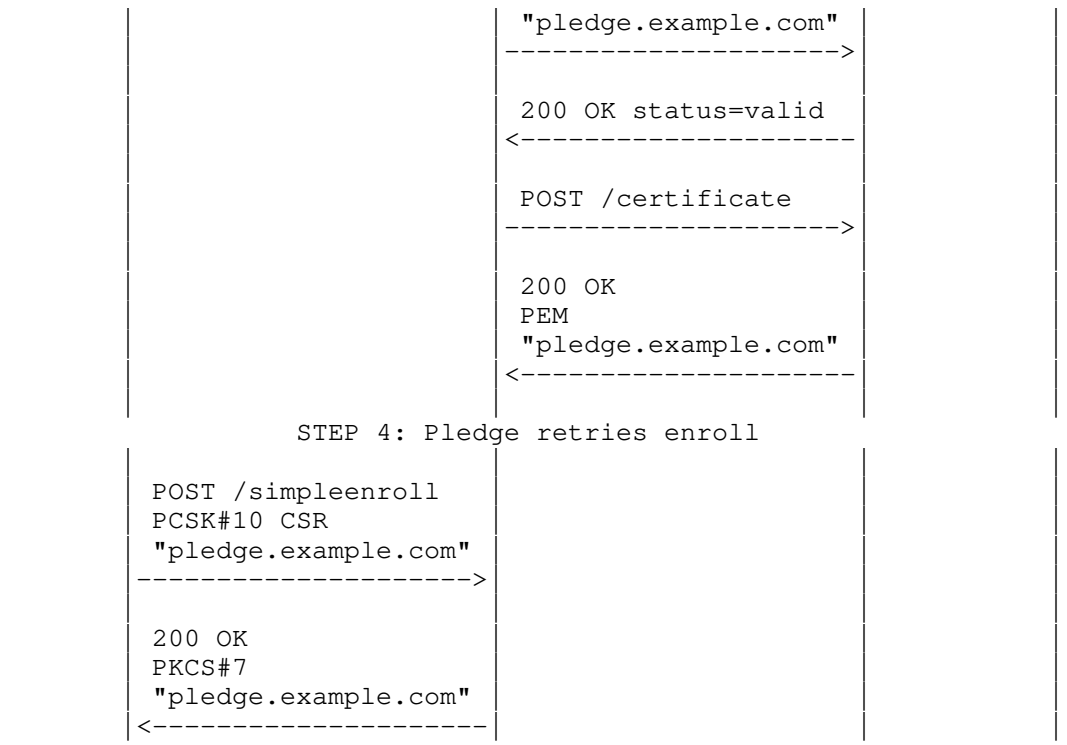
BRSKI [I-D.ietf-anima-bootstrapping-keyinfra] is based upon EST [RFC7030] and defines how to autonomically bootstrap PKI trust anchors into devices via means of signed vouchers. EST certificate enrollment may then optionally take place after trust has been established. BRKSI voucher exchange and trust establishment are based on EST extensions and the certificate enrollment part of BRSKI is fully based on EST. Similar to EST, BRSKI does not define how the EST RA communicates with the CA. Therefore, the mechanisms outlined in the previous section for using ACME as the communications protocol between the EST RA and the CA are equally applicable to BRSKI.

The following call flow shows how ACME may be integrated into a full BRSKI voucher plus EST enrollment workflow. For brevity, it assumes that the EST RA has previously proven ownership of a parent domain and that pledge certificate identifiers are a subdomain of that parent domain. The domain ownership exchanges between the RA, ACME and DNS are not shown. Similarly, not all BRSKI interactions are shown and only the key protocol flows involving voucher exchange and EST enrollment are shown.

Similar to the EST section above, the client calls EST /csrattrs API before calling the EST /simpleenroll API. This enables the server to indicate what fields the pledge should include in the CSR that the client sends in the /simpleenroll API.

The call flow illustrates the RA returning a 202 Retry-After response to the initial EST /simpleenroll API. This may be appropriate if processing of the /simpleenroll request and ACME interactions takes some timme to complete.


Pledge
EST RA
ACME
MASA
NOTE: Pre-Authorization of "example.com" is complete
STEP 1: Pledge requests Voucher
POST /requestvoucher
POST /requestvoucher
200 OK Voucher
200 OK Voucher
STEP 2: Pledge enrolls against RA
GET /csrattrs
200 OK
SEQUENCE {AttrOrOID}
SAN OID:
"pledge.example.com"
POST /simpleenroll
PCSK#10 CSR
"pledge.example.com"
202 Retry-After
STEP 3: RA places ACME order
POST /newOrder
"pledge.example.com"
201 status=ready
POST /finalize
PKCS#10 CSR

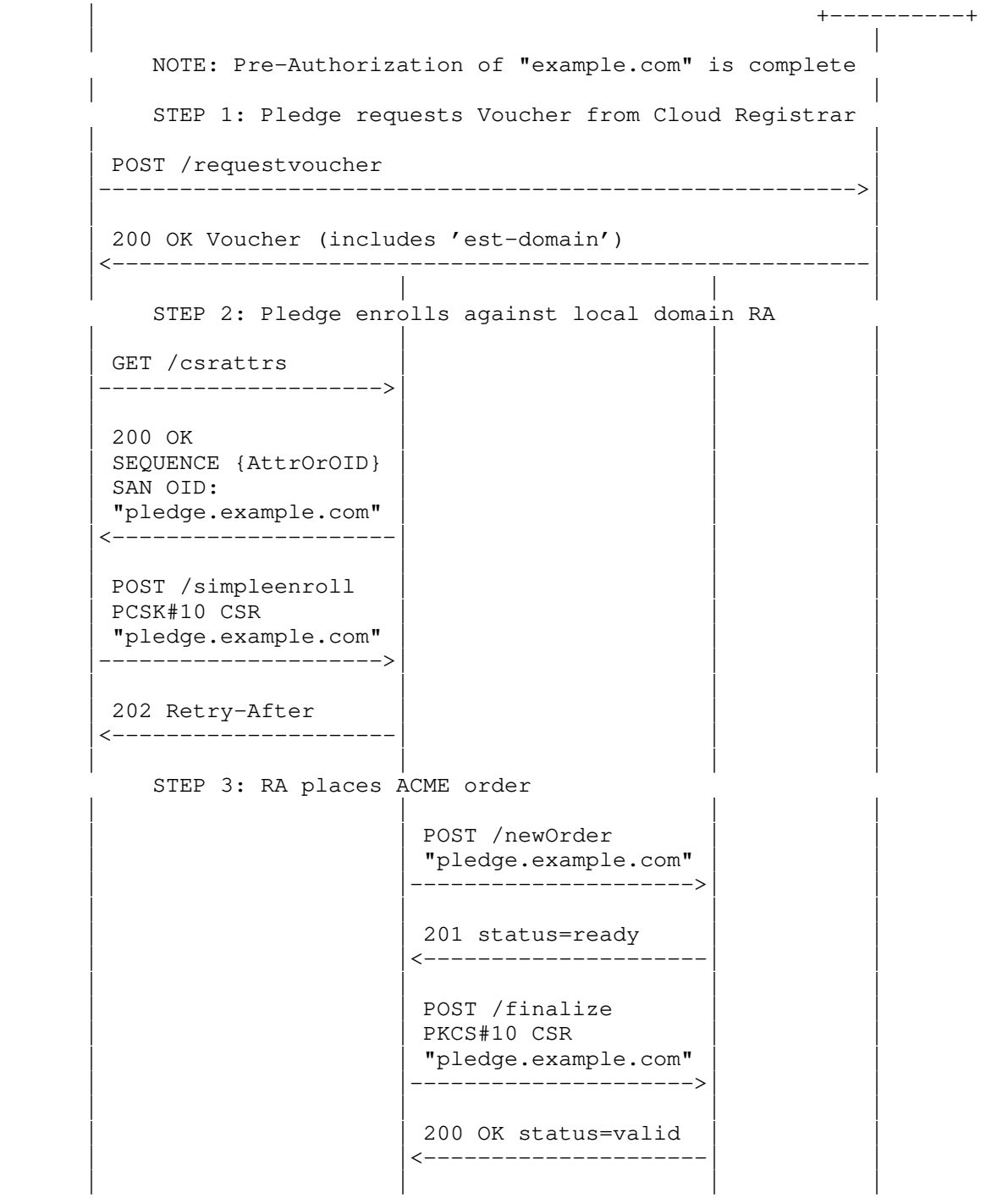


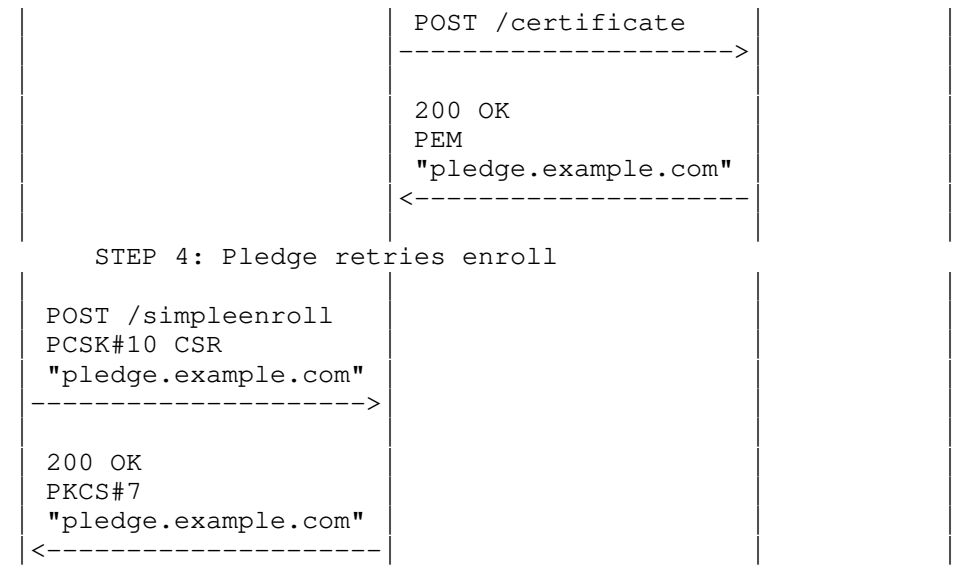
5. ACME Integration with BRSKI Default Cloud Registrar

BRSKI Cloud Registrar [I-D.friel-anima-brski-cloud] specifies the behaviour of a BRSKI Cloud Registrar, and how a pledge can interact with a BRSKI Cloud Registrar when bootstrapping. Similar to the local domain registrar BRSKI flow, ACME can be easily integrated with a cloud registrar bootstrap flow.

BRSKI cloud registrar is flexible and allows for multiple different local domain discovery and redirect scenarios. In the example illustrated here, the extension to [RFC8366] Vouchers which is defined in [I-D.friel-anima-brski-cloud], and allows the specification of a bootstrap EST domain, is leveraged. This extension allows the cloud registrar to specify the local domain RA that the pledge should connect to for the purposes of EST enrollment.

Similar to the sectiosn above, the client calls EST /csrattrs API before calling the EST /simpleenroll API.


NOTE: Pre-Authorization of "example.com" is complete
STEP 1: Pledge requests Voucher from Cloud Registrar
POST /requestvoucher
200 OK Voucher (includes 'est-domain')
STEP 2: Pledge enrolls against local domain RA
GET /csrattrs
200 OK
SEQUENCE {AttrOrOID}
SAN OID:
"pledge.example.com"
POST /simpleenroll
PCSK#10 CSR
"pledge.example.com"
202 Retry-After
STEP 3: RA places ACME order
POST /newOrder
"pledge.example.com"
201 status=ready
POST /finalize
PKCS#10 CSR
"pledge.example.com"
200 OK status=valid



6. ACME Integration with TEAP

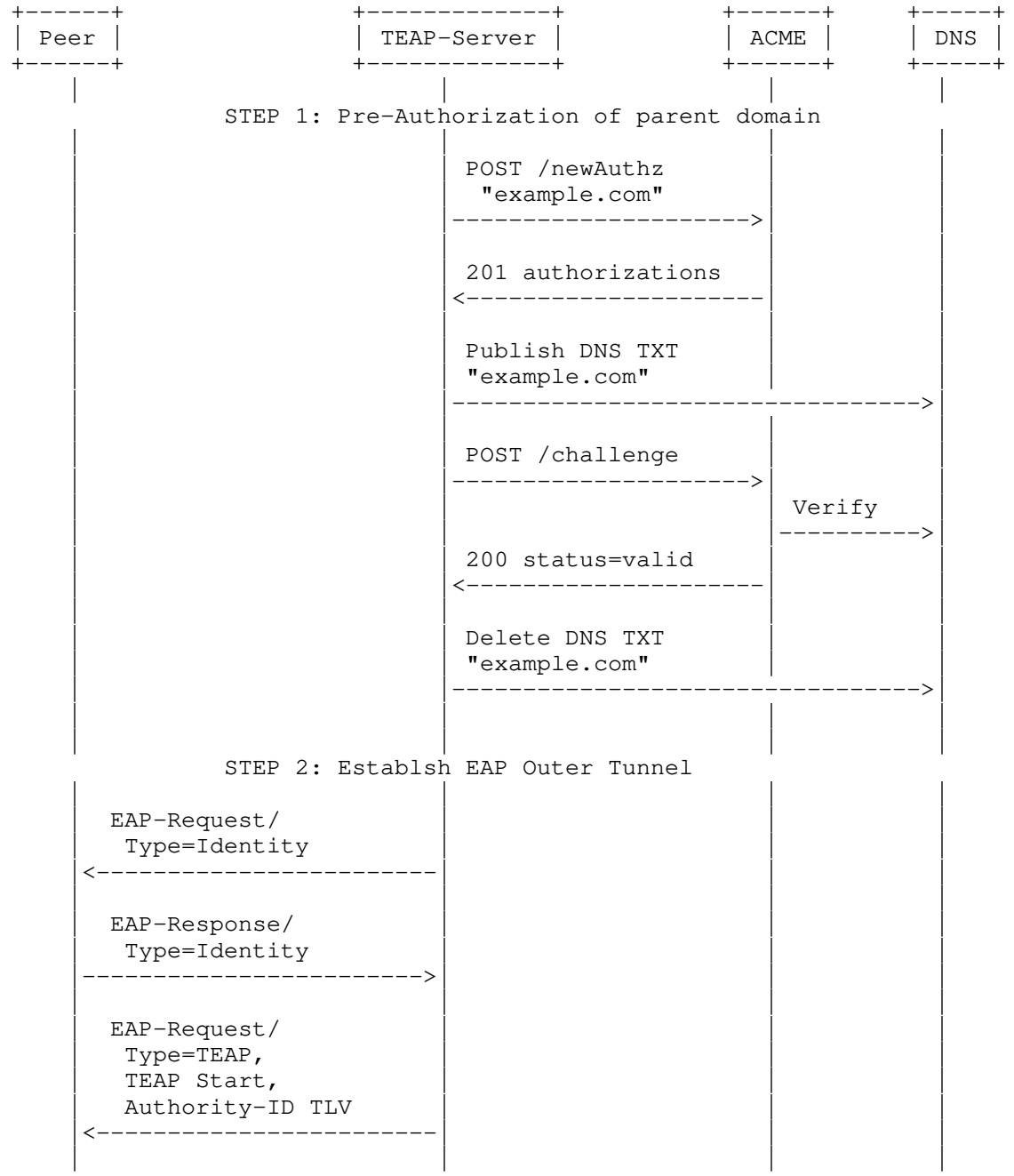
TEAP [RFC7170] defines a tunnel-based EAP method that enables secure communication between a peer and a server by using TLS to establish a mutually authenticated tunnel. TEAP enables certificate provisioning within the tunnel. TEAP Update and Extensions for Bootstrapping [I-D.lear-eap-teap-brski] defines extensions to TEAP that includes additional TLVs for certificate enrollment and BRSKI handling inside the TEAP tunnel. Neither TEAP [RFC7170] or TEAP Update and Extensions for Bootstrapping [I-D.lear-eap-teap-brski] define how the TEAP server communicates with the CA.

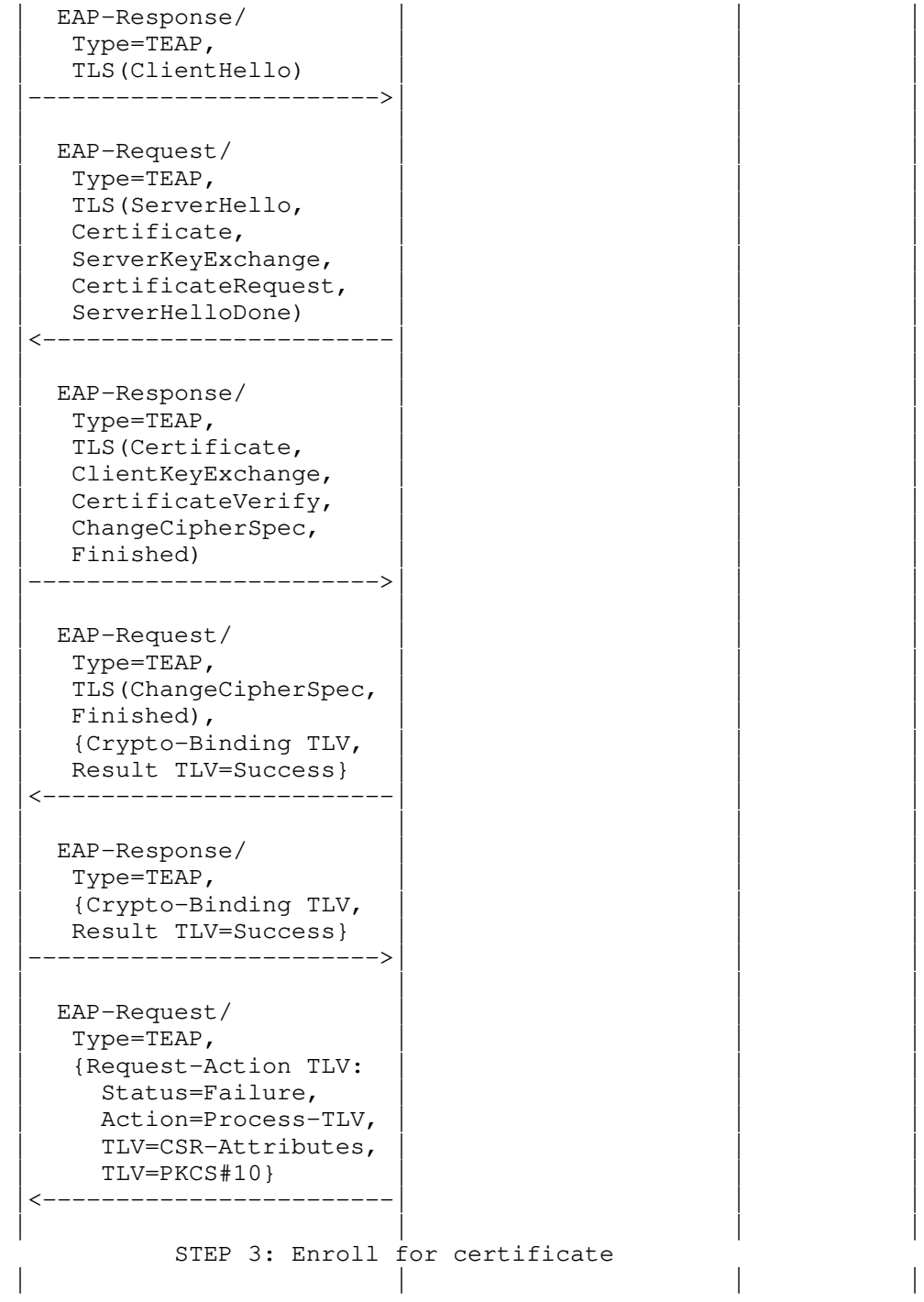
This section outlines how ACME could be used for communication between the TEAP server and the CA. The example call flow leverages [I-D.friel-acme-subdomains] and shows the TEAP server proving ownership of a parent domain, with individual client certificates being subdomains under that parent domain.

The example illustrates the TEAP server sending a Request-Action TLV including a CSR-Attributes TLV instructing the peer to send a CSR-Attributes TLV to the server. This enables the server to indicate what fields the peer should include in the CSR that the peer sends in the PKCS#10 TLV. For example, the TEAP server could instruct the peer what Subject or SAN entries to include in its CSR.

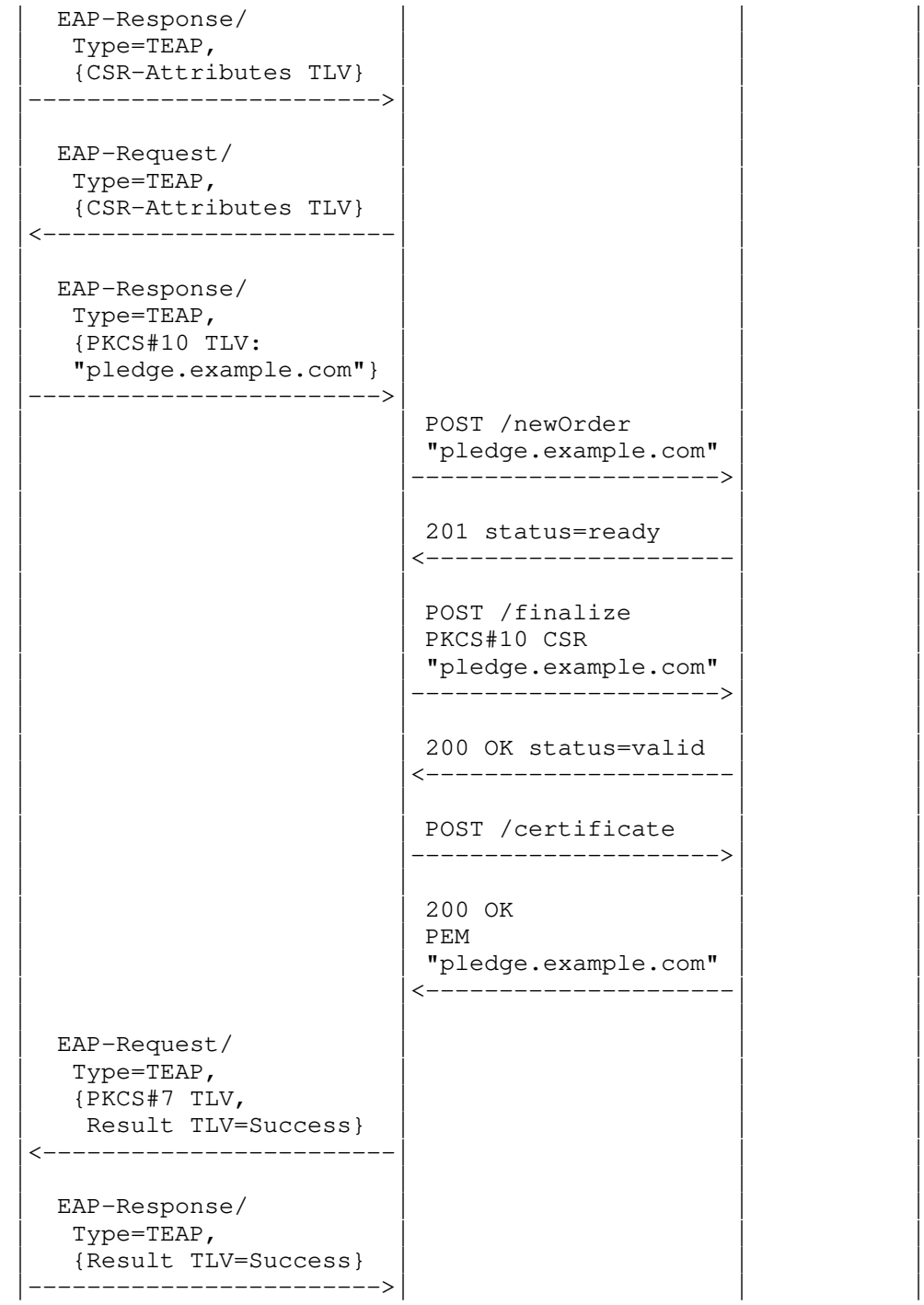
Althought not explicitly illustrated in this call flow, the Peer and TEAP Server could exchange BRSKI TLVs, and a BRSKI integration and voucher exchange with a MASA server could take place over TEAP.

Whether a BRSKI TLV exchange takes place or not does not impact the ACME specific message exchanges.





STEP 3: Enroll for certificate


EAP-Response/
Type=TEAP,
{CSR-Attributes TLV}
EAP-Request/
Type=TEAP,
{CSR-Attributes TLV}
EAP-Response/
Type=TEAP,
{PKCS#10 TLV:
"pledge.example.com"}
POST /newOrder
"pledge.example.com"
201 status=ready
POST /finalize
PKCS#10 CSR
"pledge.example.com"
200 OK status=valid
POST /certificate
200 OK
PEM
"pledge.example.com"
EAP-Request/
Type=TEAP,
{PKCS#7 TLV,
Result TLV=Success}
EAP-Response/
Type=TEAP,
{Result TLV=Success}

7. IANA Considerations

This document does not make any requests to IANA.

8. Security Considerations

This draft is informational and makes no changes to the referenced specifications. All security considerations from these referenced documents are applicable here:

- EST [RFC7030]
- BRSKI [I-D.ietf-anima-bootstrapping-keyinfra]
- BRSKI Default Cloud Registrar [I-D.friel-anima-brski-cloud]
- TEAP [RFC7170] and TEAP Update and Extensions for Bootstrapping [I-D.lear-eap-teap-brski]

Additionally, all Security Considerations in ACME in the following areas are equally applicable to ACME Integrations.

The integration mechanisms proposed here will primarily use the DNS-01 challenge documented in [RFC8555] section 8.4. The security considerations in RFC8555 says:

The DNS is a common point of vulnerability for all of these challenges. An entity that can provision false DNS records for a domain can attack the DNS challenge directly and can provision false A/AAAA records to direct the ACME server to send its HTTP validation query to a remote server of the attacker's choosing.

It is expected that the TEAP-EAP server/EST Registrar will perform DNS dynamic updates to a DNS primary server using [RFC3007] Dynamic updates, secured with with either SIG(0), or TSIG keys.

A major source of vulnerability is the disclosure of these DNS key records. An attacker that has access to them, can provision their own certificates into the the name space of the entity.

For many uses, this may allow the attacker to get access to some enterprise resource. When used to provision, for instance, a (SIP) phone system this would permit an attacker to impersonate a

legitimate phone. Not only does this allow for redirection of phone calls, but possibly also toll fraud.

Operators should consider restricting the integration server such that it can only update the DNS records for a specific zone or zones where ACME is required for client certificate enrolment automation. For example, if all IoT devices in an organisation enrol using EST against an EST RA, and all IoT devices will be issued certificates in a subdomain under iot.example.com, then the integration server could be issued a credential that only allows updating of DNS records in a zone that includes domains in the iot.example.com namespace, but does not allow updating of DNS records under any other example.com DNS namespace.

When performing challenge fulfilment via writing files to HTTP webservers, write access should only be granted to a specific set of servers, and only to a specific set of directories for storage of challenge files.

8.1. Denial of Service against ACME infrastructure

The intermdiate node (the TEAP-EAP server, or the EST Registrar) should cache the resulting certificates such that if the communication with the pledge is lost, subsequent attempts to enroll will result in the cache certificate being returned.

As many ACME servers have per-day, per-IP and per-subjectAltName limits, it is prudent not to request identical certificates too often. This could be due to operator or installer error, with multiple configuration resets occuring within a short period of time.

The cache should be keyed by the complete contents of the Certificate Signing Request, and should not persist beyond the notAfter date in the certificate.

This means that if the private/public keypair changes on the pledge, then a new certificate will be issued. If the the requested SubjectAltName changes, then a new certificate will be requested.

In a case where a device is simply factory reset, and enrolls again, then the same certificate can be returned.

9. Informative References

[I-D.friel-acme-subdomains]
Friel, O., Barnes, R., Hollebeek, T., and M. Richardson, "ACME for Subdomains", draft-friel-acme-subdomains-03 (work in progress), October 2020.

[I-D.friel-anima-brski-cloud]
Friel, O., Shekh-Yusef, R., and M. Richardson, "BRSKI Cloud Registrar", draft-friel-anima-brski-cloud-03 (work in progress), September 2020.

[I-D.ietf-anima-bootstrapping-keyinfra]
Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructures (BRSKI)", draft-ietf-anima-bootstrapping-keyinfra-45 (work in progress), November 2020.

[I-D.lear-eap-teap-brski]
Lear, E., Friel, O., Cam-Winget, N., and D. Harkins, "TEAP Update and Extensions for Bootstrapping", draft-lear-eap-teap-brski-05 (work in progress), November 2019.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3007] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, DOI 10.17487/RFC3007, November 2000, <https://www.rfc-editor.org/info/rfc3007>.

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[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>.

[RFC8555] Barnes, R., Hoffman-Andrews, J., McCarney, D., and J. Kasten, "Automatic Certificate Management Environment (ACME)", RFC 8555, DOI 10.17487/RFC8555, March 2019, <https://www.rfc-editor.org/info/rfc8555>.

Authors' Addresses

Owen Friel
Cisco

Email: ofriel@cisco.com

Richard Barnes
Cisco

Email: rlb@ipv.sx

Rifaat Shekh-Yusef
Auth0

Email: rifaat.s.ietf@gmail.com

Michael Richardson
Sandelman Software Works

Email: mcr+ietf@sandelman.ca