ACME
Internet-Draft
Intended status: Standards Track
Expires: 9 September 2020

Y. Sheffer
Intuit
D. Lopez
A. Pastor Perales
Telefonica I+D
T. Fossati
ARM

8 March 2020

An ACME Profile for Generating Delegated STAR Certificates
draft-ietf-acme-star-delegation-03

Abstract

This memo proposes a profile of the ACME protocol that allows the owner of an identifier (e.g., a domain name) to delegate to a third party access to a certificate associated with said identifier. A primary use case is that of a CDN (the third party) terminating TLS sessions on behalf of a content provider (the owner of a domain name). The presented mechanism allows the owner of the identifier to retain control over the delegation and revoke it at any time by cancelling the associated STAR certificate renewal with the ACME CA. Another key property of this mechanism is it does not require any modification to the deployed TLS ecosystem.

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 9 September 2020.

Table of Contents

1. Introduction

This document is a companion document to [I-D.ietf-acme-star]. To avoid duplication, we give here a bare-bones description of the motivation for this solution. For more details and further use cases, please refer to the introductory sections of [I-D.ietf-acme-star].

An Identifier Owner (IdO), that we can associate in the primary use case to a content provider (also referred to as Domain Name Owner, DNO), has agreements in place with one or more NDC (Name Delegation Consumer) to use and attest its identity. In the primary use case, we consider a CDN provider contracted to serve the IdO content over HTTPS. The CDN terminates the HTTPS connection at one of its edge cache servers and needs to present its clients (browsers, mobile apps, set-top-boxes) a certificate whose name matches the authority of the URL that is requested, i.e., that of the IdO. Understandably, most IdOs balk at sharing their long-term private keys with another organization and, equally, delegates would rather not have to handle other parties' long-term secrets.

Other relevant use cases are discussed in Section 4.

This document describes a profile of the ACME protocol [RFC8555] that allows the NDC to request the IdO, acting as a profiled ACME server, a certificate for a delegated identity - i.e., one belonging to the IdO. The IdO then uses the ACME protocol (with the extensions described in [I-D.ietf-acme-star]) to request issuance of a STAR certificate for the same delegated identity. The generated short-term certificate is automatically renewed by the ACME Certification Authority (CA), periodically fetched by the NDC and used to terminate HTTPS connections in lieu of the IdO. The IdO can end the delegation at any time by simply instructing the CA to stop the automatic renewal and letting the certificate expire shortly thereafter.

In case the delegated identity is a domain name, this document also provides a way for the NDC to inform the IdO about the CNAME mappings that need to be installed in the IdO's DNS zone to enable the aliasing of the delegated name, thus allowing the complete name delegation workflow to be handled using a single interface.

1.1. Terminology

IdO Identifier Owner, the owner of an identifier (e.g., a domain name) that needs to be delegated.
DNO Domain Name Owner, a specific kind of IdO whose identifier is a domain name
NDC Name Delegation Consumer, the entity to which the domain name is delegated for a limited time. This is a CDN in the primary use case (in fact, readers may note the symmetry of the two acronyms).
CDN Content Delivery Network, a widely distributed network that serves the domain's web content to a wide audience at high performance.
STAR Short-Term, Automatically Renewed X.509 certificates.
ACME The IETF Automated Certificate Management Environment, a certificate management protocol.
CA A Certificate Authority that implements the ACME protocol.

1.2. Conventions used in this document

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.

2. Protocol Flow

This section presents the protocol flow. For completeness, we include the ACME profile proposed in this draft as well as the extended ACME protocol described in [I-D.ietf-acme-star].

2.1. Preconditions

The protocol assumes the following preconditions are met:

* The IdO exposes an ACME server interface to the NDC(s) comprising the account management interface;
* The NDC has registered an ACME account with the IdO;
* NDC and IdO have agreed on a "CSR template" to use, including at a minimum: subject name (e.g., "somesite.example.com"), requested algorithms and key length, key usage, extensions (e.g., TNAuthList). The NDC is required to use this template for every CSR created under the same delegation;
* IdO has registered an ACME account with the Certificate Authority (CA)

Note that even if the IdO implements the ACME server role, it is not acting as a CA: in fact, from the point of view of the certificate issuance process, the IdO only works as a "policing" forwarder of the

NDC's key-pair and is responsible for completing the identity verification process towards the ACME CA.

2.2. Overview

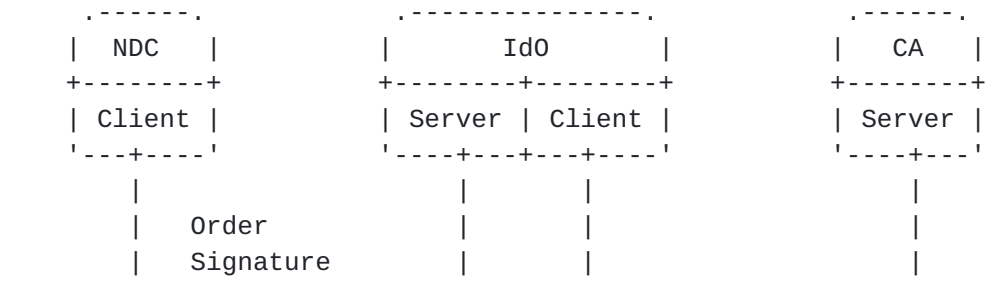
The interaction between the NDC and the IdO is governed by the profiled ACME workflow detailed in Section 2.3. The interaction between the IdO and the CA is ruled by ACME STAR [I-D.ietf-acme-star] as well as any other ACME extension that applies (e.g., [I-D.ietf-acme-authority-token-tnauthlist] for STIR).

The outline of the combined protocol is as follow (Figure 1):

* NDC sends an Order for the delegated identifier to IdO;
* IdO creates an Order resource in state "ready" with a "finalize" URL;
* NDC immediately sends a finalize request (which includes the CSR) to the IdO;
* IdO verifies the CSR according to the agreed upon CSR template;
* If the CSR verification fails, the Order is moved to an "invalid" state and everything stops;
* If the CSR verification is successful, IdO moves the Order to state "processing", and sends an Order' (using its own account) for the delegated identifier to the ACME STAR CA;
* If the ACME STAR protocol fails, Order' moves to "invalid" and the same state is reflected in the NDC Order;
* If the ACME STAR run is successful (i.e., Order' is "valid"), IdO copies the "star-certificate" URL from Order' to Order and moves its state to "valid".

The NDC can now download, install and use the short-term certificate bearing the name delegated by the IdO. This sequence of actions is repeated until the STAR certificate expires or the IdO decides to cancel the automatic renewal process with the ACME STAR CA.

Note that, because the identity validation is suppressed, the NDC sends the finalize request, including the CSR, to the IdO immediately after the Order has been acknowledged. The IdO must buffer a (valid) CSR until the Validation phase completes successfully.



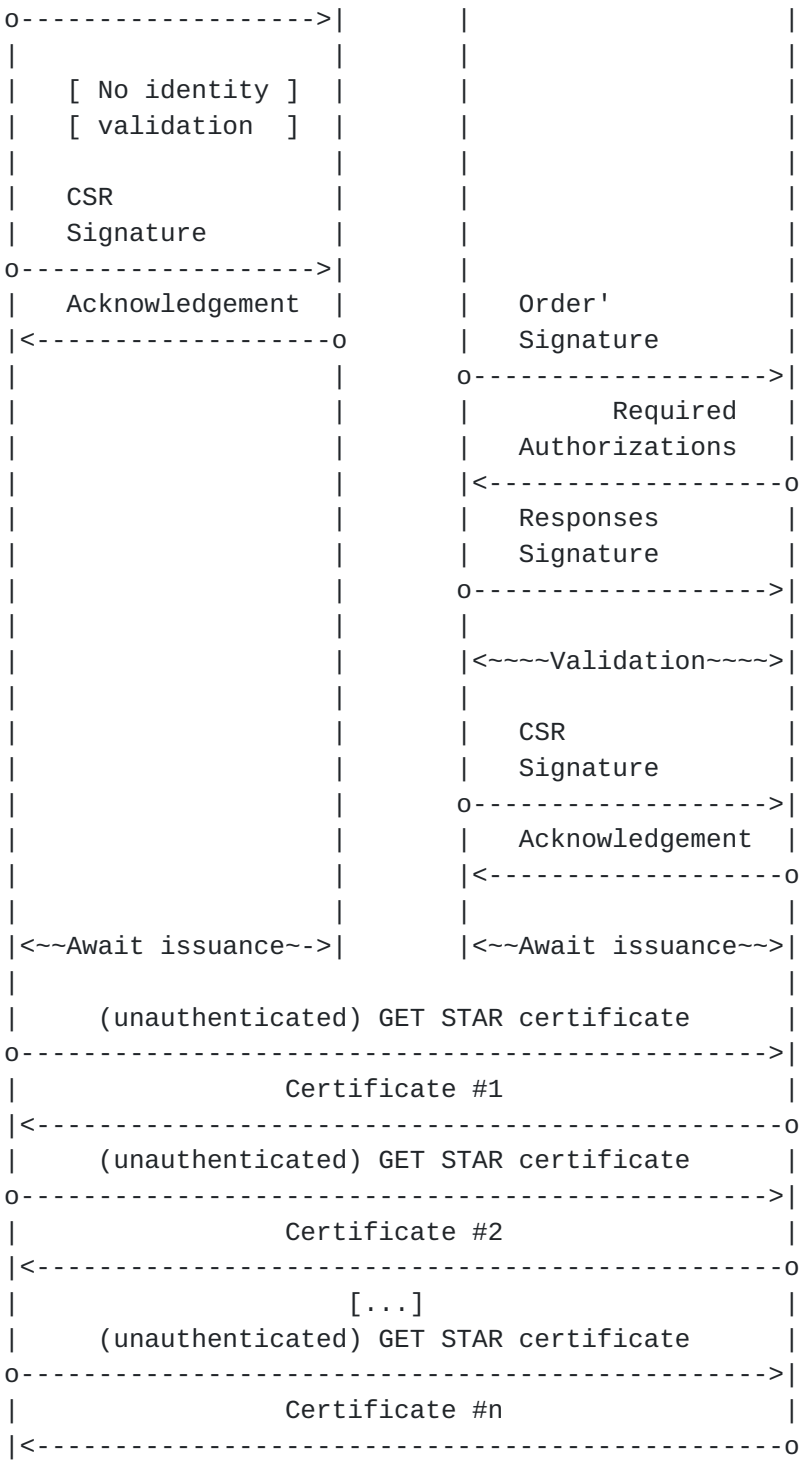


Figure 1: End to end STAR delegation flow

2.3. Delegated Identity Profile

2.3.1. Order Object on the NDC-IdO side

The Order object created by the NDC:

* MUST contain identifiers with the new "delegated" field set to true;
* MUST NOT contain the notBefore and notAfter fields;
* MAY contain an "auto-renewal" object and inside it, any of the fields listed in Section 3.1.1 of [I-D.ietf-acme-star];
* In case the identifier type is "dns", it MAY contain a "cname" field with the alias of the identifier in the NDC domain. This field is used by the IdO to create the DNS aliasing needed to redirect the resolvers to the delegated entity.

```
POST /acme/new-order HTTP/1.1
Host: acme.dno.example
Content-Type: application/jose+json

{
  "protected": base64url({
    "alg": "ES256",
    "kid": "https://acme.dno.example/acme/acct/evOfKhNU60wg",
    "nonce": "5XJ1L3lEkMG7tR6pA00clA",
    "url": "https://acme.dno.example/acme/new-order"
  }),
  "payload": base64url({
    "identifiers": [
      {
        "type": "dns",
        "value": "abc.ndc.dno.example.",
        "delegated": true,
        "cname": "abc.ndc.example."
      }
    ],
  }),
  "signature": "H6ZXtGjTZyUnPeKn...wEA4TklBdh3e454g"
}
```

The Order object that is created on the IdO:

* MUST start in the "ready" state;
* MUST contain an "authorizations" array with zero elements;
* MUST NOT contain the "notBefore" and "notAfter" fields.

```
{
  "status": "ready",
  "expires": "2016-01-01T00:00:00Z",

  "identifiers": [
   {
     "type": "dns",
     "value": "abc.ndc.dno.example.",
     "delegated": true,
     "cname": "abc.ndc.example."
   }
  ],

  "authorizations": [],

  "finalize": "https://acme.dno.example/acme/order/TO8rfgo/finalize"
}
```

The IdO SHOULD copy the "auto-renewal" object (if it exists) from the NDC request into the related STAR request to the ACME CA.

When the validation of the identifiers has been successfully completed and the certificate has been issued by the CA, the IdO:

* MUST move its Order resource status to "valid";
* MUST copy the "star-certificate" field from the STAR Order;

The latter indirectly includes (via the NotBefore and NotAfter HTTP headers) the renewal timers needed by the NDC to inform its certificate reload logic.

```
{
  "status": "valid",
  "expires": "2016-01-01T00:00:00Z",

  "identifiers": [
   {
     "type": "dns",
     "value": "abc.ndc.dno.example.",
     "delegated": true,
     "cname": "abc.ndc.example."
   }
  ],

  "authorizations": [],

  "finalize": "https://acme.dno.example/acme/order/TO8rfgo/finalize",

  "star-certificate": "https://acme.ca.example/acme/order/yTr23sSDg9"
}
```

If an "identifier" object of type "dns" was included, the IdO MUST validate the specified CNAME at this point in the flow. The NDC and IdO may have a pre-established list of valid CNAME values. At the minimum, the IdO MUST verify that both DNS names are syntactically valid.

Following this validation, the IdO can add the CNAME records to its zone:

```
abc.ndc.dno.example. CNAME abc.ndc.example.
```

2.3.2. Order Object on the IdO-CA side

When sending the Order to the ACME CA, the IdO SHOULD strip the "delegated" and "cname" attributes sent by the NDC (Section 2.3.1). The IdO MUST add the necessary STAR extensions to the Order. In addition, to allow the NDC to download the certificate using unauthenticated GET, the IdO MUST add the allow-certificate-get attribute and set it to true. This implies that an "auto-renewal" object must be included in the Order.

2.3.3. Capability Discovery

In order to help a client to discover support for this profile, the directory object of an ACME server MUST contain the following attribute inside the "auto-renewal" object in the "meta" field:

* delegation-enabled: boolean flag indicating support for the profile specified in this memo. An ACME server that supports this delegation profile MUST include this key, and MUST set it to true.

2.3.4. On Cancellation

It is worth noting that cancellation of the ACME STAR certificate is a prerogative of the IdO. The NDC does not own the relevant account key on the ACME CA, therefore it can't issue a cancellation request for the STAR cert. Potentially, since it holds the STAR certificate's private key, it could request the revocation of a single STAR certificate. However, STAR explicitly disables the revokeCert interface.

2.4. Proxy Behavior

There are cases where the ACME Delegation flow should be proxied, such as the use case described in Section 4.1.2. This section describes the behavior of such proxies.

An ACME Delegation server can decide, on a per-identity case, whether to act as a proxy into another ACME Delegation server, or to behave as an IdO and obtain a certificate directly. The determining factor is whether the server can successfully be authorized by the ACME Server for the identity associated with the certificate request.

The identities supported by each server and the disposition for each of them are preconfigured.

Following is the proxy's behavior for each of the messages exchanged in the ACME Delegation process:

* New-order request:
  - The complete "identifiers" object MUST be copied as-is.
  - Similarly, the "auto-renewal" object MUST be copied as-is.
* New-order response:
  - The "status", "expires", "authorizations", "identifiers" and "auto-renewal" attributes/objects MUST be copied as-is.
  - The "finalize" URL is rewritten, so that the "finalize" request will be made to the proxy.
  - Similarly, the Location header is rewritten.
* Get Order response:
  - The "status", "expires", "authorizations", "identifiers" and "auto-renewal" attributes/objects MUST be copied as-is.
  - Similarly, the "star-certificate" URL MUST be copied as-is.
  - The "finalize" URL is rewritten, so that the "finalize" request will be made to the proxy.
  - The "Location" header must be rewritten.

* Finalize request:
  - The CSR MUST be copied as-is.
* Finalize response:
  - Both the Location header and the "finalize" URLs are rewritten.

We note that all the above messages are authenticated, and therefore each proxy must be able to authenticate any subordinate server.

3. CSR Template

The CSR template is used to express and constrain the shape of the CSR that the NDC uses to request the certificate. The CSR is used for every certificate created under the same delegation. Its validation by the IdO is a critical element in the security of the whole delegation mechanism.

Instead of defining every possible CSR attribute, this document takes a minimalist approach by declaring only the minimum attribute set and deferring the registration of further, more specific, attributes to future documents.

3.1. Template Syntax

The template is a JSON document. Each field denotes one of:

* A mandatory field, where the template specifies the literal value of that field. This is denoted by a literal string, such as "client1.ndc.dno.example.com".
* A mandatory field, where the content of the field is defined by the client. This is denoted by "**".
* An optional field, where the client decides whether the field is included in the CSR and what its value is. This is denoted by "*".

The NDC MUST NOT include in the CSR any fields that are not specified in the template, and in particular MUST NOT add any extensions unless those were previously negotiated out of band with the IdO.

The mapping between X.509 CSR fields and the template will be defined in a future revision of this document.

When the CSR is received by the IdO, it MUST verify that the CSR is consistent with the template that the IdO sent earlier. The IdO MAY enforce additional constraints, e.g. by restricting field lengths.

3.2. Example

The CSR template in Figure 2 represents one possible CSR template governing the delegation exchanges provided in the rest of this document.

```
{
  "keyTypes": [
    {
      "PublicKeyType": "RSA",
      "PublicKeyLength": 4096,
      "SignatureType": "sha256WithRSAEncryption"
    }
  ],
  "subject": {
    "country": "CA",
    "stateOrProvince": "**",
    "locality": "**",
    "commonName": "**"
  },
  "extensions": {
    "subjectAltName": {
      "DNS": [
        "client1.ndc.dno.example"
      ],
      "IP": [
        "1.2.3.4",
        "13::17"
      ]
    },
    "keyUsage": [
      "digitalSignature"
    ],
    "extendedKeyUsage": [
      "serverAuth",
      "timeStamping"
    ]
  }
}
```

Figure 2: Example CSR template

The template syntax is defined in Appendix B.

4. Further Use Cases

4.1. CDNI

[I-D.ietf-cdni-interfaces-https-delegation] discusses several solutions addressing different delegation requirements for the CDNI (CDN Interconnection) environment. This section discusses two of the stated requirements in the context of the STAR delegation workflow.

4.1.1. Multiple Parallel Delegates

In some cases the content owner (IdO) would like to delegate authority over a web site to multiple NDCs (CDNs). This could happen if the IdO has agreements in place with different regional CDNs for different geographical regions, or if a "backup" CDN is used to handle overflow traffic by temporarily altering some of the CNAME mappings in place. The STAR delegation flow enables this use case naturally, since each CDN can authenticate separately to the IdO (via its own separate account) specifying its CSR, and the IdO is free to allow or deny each certificate request according to its own policy.

4.1.2. Chained Delegation

In other cases, a content owner (IdO) delegates some domains to a large CDN (uCDN), which in turn delegates to a smaller regional CDN, dCDN. The DNO has a contractual relationship with uCDN, and uCDN has a similar relationship with dCDN. However IdO may not even know about dCDN.

If needed, the STAR protocol can be chained to support this use case: uCDN could forward requests from dCDN to DNO, and forward responses back to dCDN. Whether such proxying is allowed is governed by policy and contracts between the parties.

A mechanism is necessary at the interface between uCDN and dCDN by which the uCDN can advertise:

* The namespace that is made available to the dCDN to mint its delegated names;
* The policy for creating the key material (allowed algorithms, minimum key lengths, key usage, etc.) that the dCDN needs to satisfy.

Note that such mechanism is provided by the CSR template.

4.1.2.1. Two-Level Delegation in CDNI

A User Agent (browser or set-top-box) wants to fetch the video resource at the following URI: "https://video.cp.example/movie". Redirection between Content Provider, upstream, and downstream CDNs is arranged as a CNAME-based aliasing chain as illustrated in Figure 3.

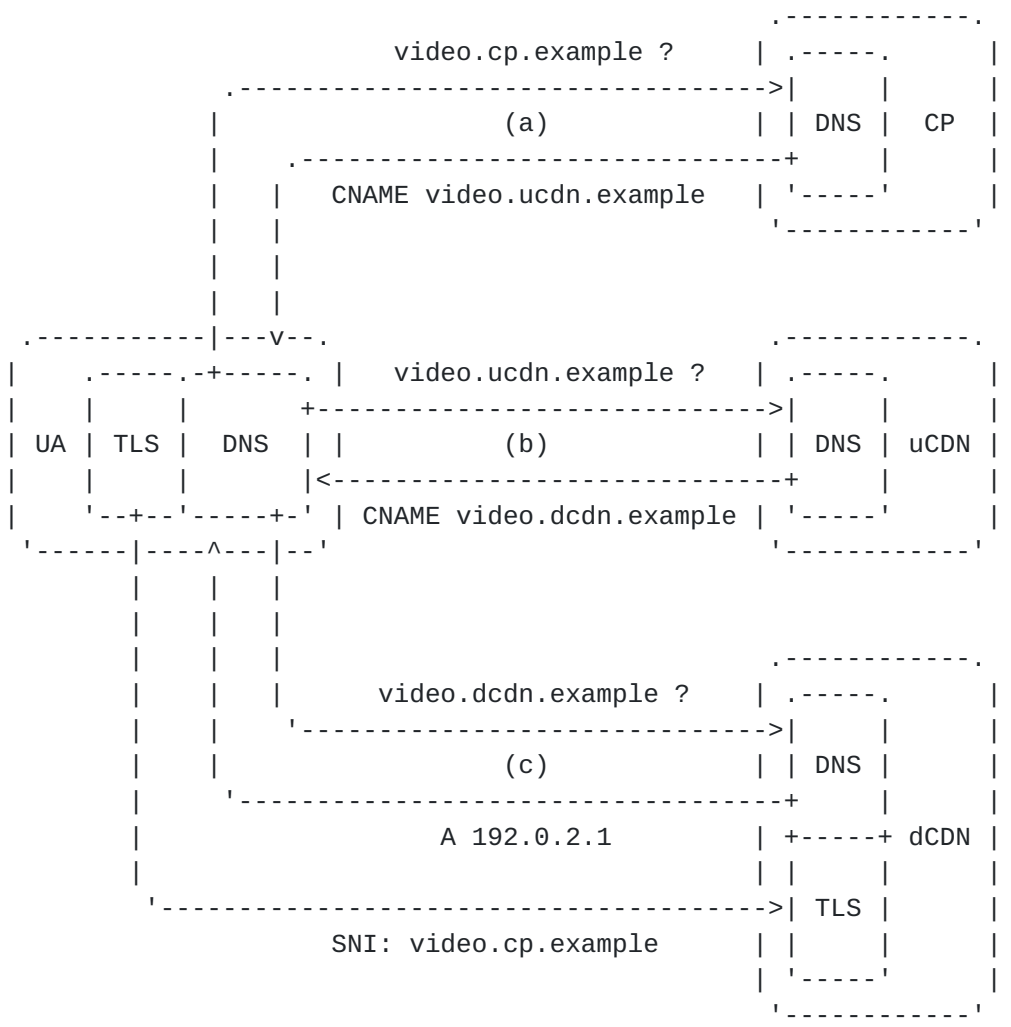


Figure 3: DNS Redirection

Unlike HTTP based redirection, where the original URL is supplanted by the one found in the Location header of the 302 response, DNS redirection is completely transparent to the User Agent. As a result, the TLS connection to the dCDN edge is done with an SNI equal to the "host" in the original URI - in the example, "video.cp.example". So, in order to successfully complete the handshake, the landing dCDN node has to be configured with a

certificate whose SAN matches "video.cp.example", i.e., a Content Provider's name.

Figure 4 illustrates the cascaded delegation flow that allows dCDN to obtain a STAR certificate that bears a name belonging to the Content Provider with a private key that is only known to the dCDN.

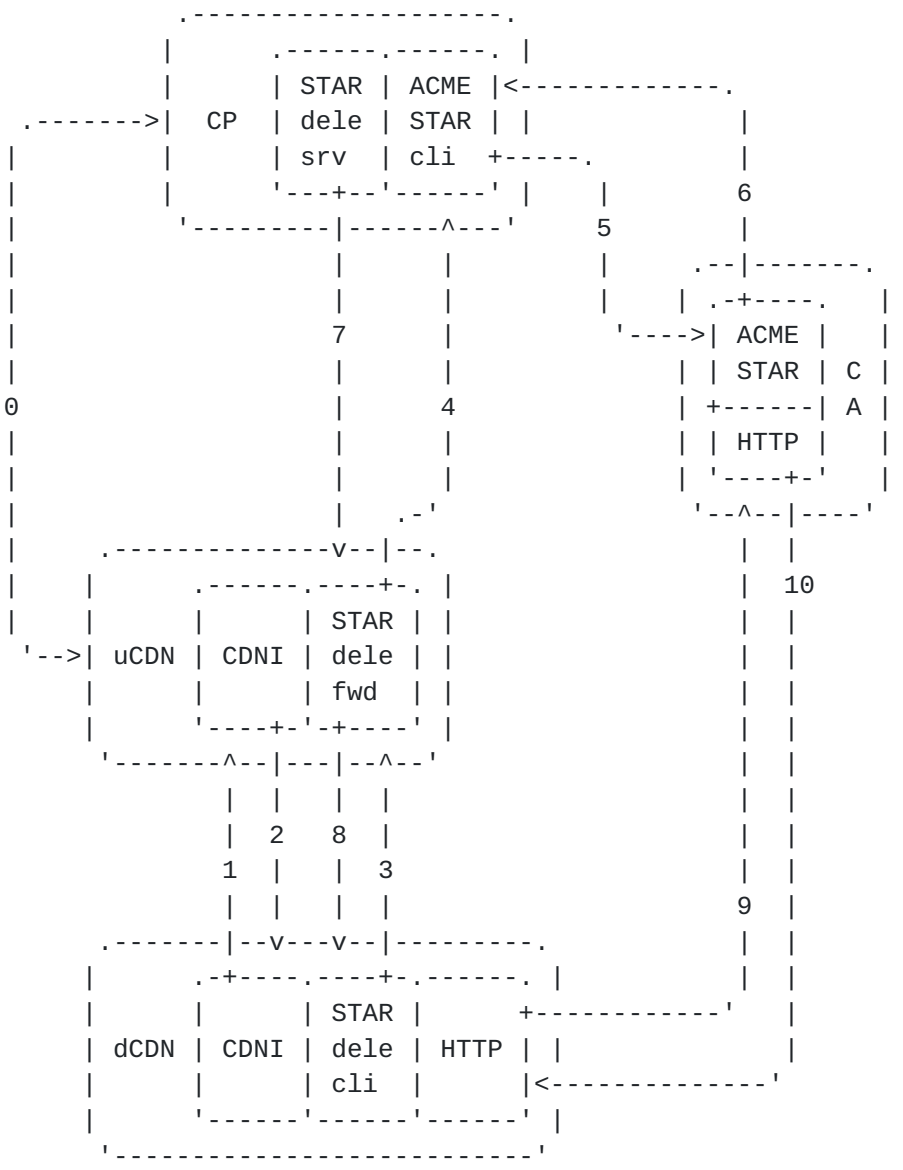


Figure 4: Two levels delegation in CDNI

TBD bootstrap, see https://github.com/yaronf/I-D/issues/47

1. dCDN requests CDNI path metadata to uCDN;
2. uCDN replies with, among other CDNI things, the STAR delegation

configuration, which includes the delegated Content Provider's name;
3. dCDN creates a key-pair and the CSR with the delegated name. It then places an order for the delegated name to uCDN;
4. uCDN forwards the received order to the Content Provider (CP);
5. CP creates an order for a STAR certificate and sends it to the ACME CA. The order also requests unauthenticated access to the certificate resource;
6. After all authorizations complete successfully, the STAR certificate is issued;
7. CP notifies uCDN that the STAR cert is available at the order's star-certificate URL;
8. uCDN forwards the information to dCDN. At this point the ACME signalling is complete;
9. dCDN requests the STAR cert using unauthenticated GET from the ACME CA;
10. the CA returns the certificate. Now dCDN is fully configured to handle HTTPS traffic in-lieu of the Content Provider.

Note that 9. and 10. repeat until the delegation expires or is terminated.

4.2. STIR

As a second use case, we consider the delegation of credentials in the STIR ecosystem [I-D.ietf-stir-cert-delegation].

In the STIR "delegated" mode, a service provider SP2 - the NDC - needs to sign PASSPorT's [RFC8225] for telephone numbers (e.g., TN=+123) belonging to another service provider, SP1 - the IdO. In order to do that, SP2 needs a STIR certificate, and private key, that includes TN=+123 in the TNAuthList [RFC8226] cert extension.

In details (Figure 5):

1. SP1 and SP2 agree on the configuration of the delegation - in particular, the CSR template that applies;
2. SP2 generates a private/public key-pair and sends a CSR to SP1 requesting creation of a certificate with: SP1 name, SP2 public key, and a TNAuthList extension with the list of TNs that SP1 delegates to SP2. (Note that the CSR sent by SP2 to SP1 needs to be validated against the CSR template agreed upon in step 1.);
3. SP1 sends an Order for the CSR to the ACME STAR CA;
4. Subsequently, after the required TNAuthList authorizations are successfully completed, the ACME STAR CA moves the Order to a "valid" state; at the same time the star-certificate endpoint is populated.

5. The Order contents are forwarded from SP1 to SP2 by means of the paired "delegation" Order.
6. SP2 dereferences the star-certificate URL in the Order to fetch the rolling STAR certificate bearing the delegated identifiers.

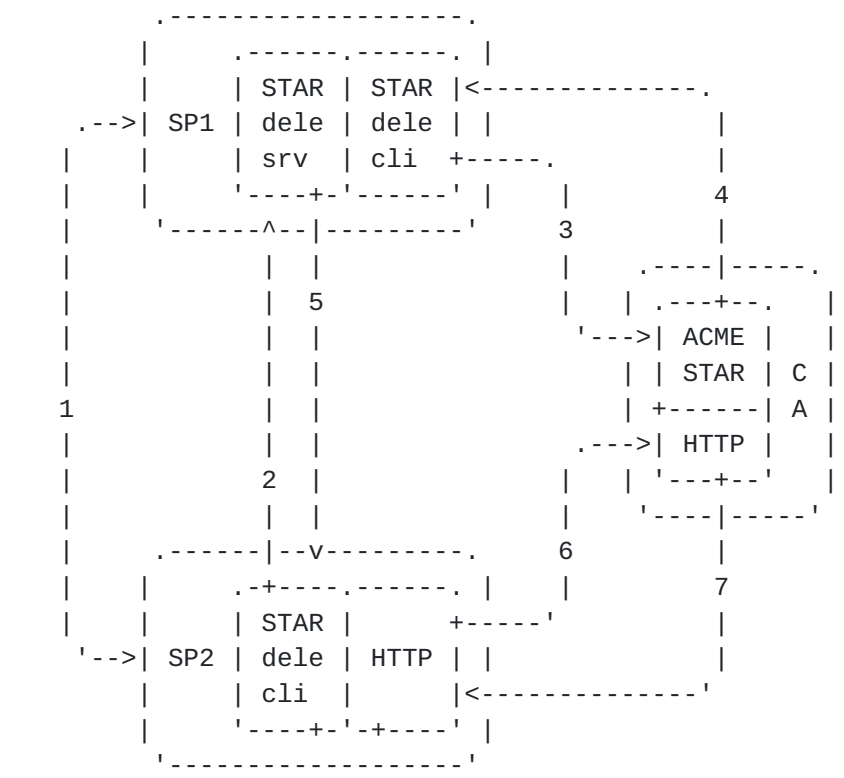


Figure 5: Delegation in STIR

As shown, the STAR delegation profile described in this document applies straightforwardly, the only extra requirement being the ability to instruct the NDC about the allowed TNAuthList values. This can be achieved by a simple extension to the CSR template.

5. IANA Considerations

[[RFC Editor: please replace XXXX below by the RFC number.]]

5.1. New Fields in the "auto-renewal" Object within a Directory Metadata Object

This document adds the following entries to the ACME Directory Metadata Fields:

Field Name	Field Type	Reference
delegation-enabled	boolean	RFC XXXX

Table 1

5.2. New Fields for Identifiers

* "delegated", TBD.

5.3. CSR Template Registry

TODO

6. Security Considerations

6.1. Restricting CDNs to the Delegation Mechanism

When a web site is delegated to a CDN, the CDN can in principle modify the web site at will, create and remove pages. This means that a malicious or breached CDN can pass the ACME (as well as common non-ACME) HTTPS-based validation challenges and generate a certificate for the site. This is true regardless of whether the CNAME mechanisms defined in the current document is used or not.

In some cases, this is the desired behavior: the domain owner trusts the CDN to have full control of the cryptographic credentials for the site. The current document however assumes that the domain owner only wants to delegate restricted control, and wishes to retain the capability to cancel the CDN's credentials at a short notice.

Following is the proposed solution where the IdO wishes to ensure that a rogue CDN cannot issue unauthorized certificates:

* The domain owner makes sure that the CDN cannot modify the DNS records for the domain. The domain owner should ensure it is the only entity authorized to modify the DNS zone. Typically, it establishes a CNAME resource record from a subdomain into a CDN-managed domain.
* The domain owner uses a CAA record [RFC6844] to restrict certificate issuance for the domain to specific CAs that comply with ACME and are known to implement [RFC8657].
* The domain owner uses the ACME-specific CAA mechanism [RFC8657] to restrict issuance to a specific account key which is controlled by it, and MUST require "dns-01" as the sole validation method.

We note that the above solution may need to be tweaked depending on the exact capabilities and authorisation flows supported by the selected CAs.

6.2. TBC

* CSR validation
* CNAME mappings
* Composition with ACME STAR
* Composition with other ACME extensions
* Channel security

7. Acknowledgments

This work is partially supported by the European Commission under Horizon 2020 grant agreement no. 688421 Measurement and Architecture for a Middleboxed Internet (MAMI). This support does not imply endorsement.

8. References

8.1. Normative References

[I-D.handrews-json-schema]
Wright, A., Andrews, H., Hutton, B., and G. Dennis, "JSON Schema: A Media Type for Describing JSON Documents", Work in Progress, Internet-Draft, draft-handrews-json-schema-02, 17 September 2019, <http://www.ietf.org/internet-drafts/draft-handrews-json-schema-02.txt>.

[I-D.ietf-acme-star]
Sheffer, Y., Lopez, D., Dios, O., Pastor, A., and T. Fossati, "Support for Short-Term, Automatically-Renewed (STAR) Certificates in Automated Certificate Management Environment (ACME)", Work in Progress, Internet-Draft, draft-ietf-acme-star-11, 24 October 2019, <http://www.ietf.org/internet-drafts/draft-ietf-acme-star-11.txt>.

[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>.

[RFC6844] Hallam-Baker, P. and R. Stradling, "DNS Certification Authority Authorization (CAA) Resource Record", RFC 6844,

DOI 10.17487/RFC6844, January 2013,
<https://www.rfc-editor.org/info/rfc6844>.

[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>.

[RFC8657] Landau, H., "Certification Authority Authorization (CAA) Record Extensions for Account URI and Automatic Certificate Management Environment (ACME) Method Binding", RFC 8657, DOI 10.17487/RFC8657, November 2019, <https://www.rfc-editor.org/info/rfc8657>.

8.2. Informative References

[I-D.ietf-acme-authority-token-tnauthlist]
Wendt, C., Hancock, D., Barnes, M., and J. Peterson, "TNAuthList profile of ACME Authority Token", Work in Progress, Internet-Draft, draft-ietf-acme-authority-token-tnauthlist-05, 4 November 2019, <http://www.ietf.org/internet-drafts/draft-ietf-acme-authority-token-tnauthlist-05.txt>.

[I-D.ietf-cdni-interfaces-https-delegation]
Fieau, F., Emile, S., and S. Mishra, "CDNI extensions for HTTPS delegation", Work in Progress, Internet-Draft, draft-ietf-cdni-interfaces-https-delegation-02, 4 November 2019, <http://www.ietf.org/internet-drafts/draft-ietf-cdni-interfaces-https-delegation-02.txt>.

[I-D.ietf-stir-cert-delegation]
Peterson, J., "STIR Certificate Delegation", Work in Progress, Internet-Draft, draft-ietf-stir-cert-delegation-01, 4 November 2019, <http://www.ietf.org/internet-drafts/draft-ietf-stir-cert-delegation-01.txt>.

[RFC8225] Wendt, C. and J. Peterson, "PASSporT: Personal Assertion Token", RFC 8225, DOI 10.17487/RFC8225, February 2018, <https://www.rfc-editor.org/info/rfc8225>.

[RFC8226] Peterson, J. and S. Turner, "Secure Telephone Identity Credentials: Certificates", RFC 8226, DOI 10.17487/RFC8226, February 2018, <https://www.rfc-editor.org/info/rfc8226>.

Appendix A. Document History

[[Note to RFC Editor: please remove before publication.]]

A.1. draft-ietf-acme-star-delegation-03

* Consistency with the latest changes in the base ACME STAR document, e.g. star-delegation-enabled capability renamed and moved.
* Proxy use cases (recursive delegation) and the definition of proxy behavior.
* More detailed analysis of the CDNI and STIR use cases, including sequence diagrams.

A.2. draft-ietf-acme-star-delegation-02

* Security considerations: review by Ryan Sleevi.
* CSR template simplified: instead of being a JSON Schema document itself, it is now a simple JSON document which validates to a JSON Schema.

A.3. draft-ietf-acme-star-delegation-01

* Refinement of the CDNI use case.
* Addition of the CSR template (partial, more work required).
* Further security considerations (work in progress).

A.4. draft-ietf-acme-star-delegation-00

* Republished as a working group draft.

A.5. draft-sheffer-acme-star-delegation-01

* Added security considerations about disallowing CDNs from issuing certificates for a delegated domain.

A.6. draft-sheffer-acme-star-delegation-00

* Initial version, some text extracted from draft-sheffer-acme-star-requests-02

Appendix B. CSR Template Schema

Following is a JSON Schema definition of the CSR template. The syntax used is that of draft 7 of JSON Schema, which may not be the latest version of the corresponding Internet Draft [I-D.handrews-json-schema] at the time of publication.

While the CSR template must follow the syntax defined here, neither the IdO nor the NDC are expected to validate it at run-time.

```
{
  "title": "JSON Schema for the STAR Delegation CSR template",
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "http://ietf.org/acme/drafts/star-delegation/csr-template",
  "$def": {
    "distinguished-name": {
      "$id": "#distinguished-name",
      "type": "object",
      "properties": {
        "country": {
          "type": "string"
        },
        "stateOrProvince": {
          "type": "string"
        },
        "locality": {
          "type": "string"
        },
        "organization": {
          "type": "string"
        },
        "organizationalUnit": {
          "type": "string"
        },
        "emailAddress": {
          "type": "string"
        },
        "commonName": {
          "type": "string"
        }
      },
      "additionalProperties": false
    },
    "rsaKeyType": {
      "$id": "#rsaKeyType",
      "type": "object",
      "properties": {
        "PublicKeyType": {
```

```
            "type": "string",
            "const": "RSA"
          },
          "PublicKeyLength": {
            "type": "integer"
          },
          "SignatureType": {
            "type": "string",
            "enum": [
              "sha256WithRSAEncryption"
            ]
          }
        },
        "additionalProperties": false
      },
      "ecKeyTYpe": {
        "$id": "#ecKeyType",
        "type": "object",
        "properties": {
          "PublicKeyType": {
            "type": "string",
            "const": "ecPublicKey"
          },
          "Curve": {
            "type": "string",
            "enum": [
              "secp521r1"
            ]
          },
          "SignatureType": {
            "type": "string",
            "enum": [
              "ecdsa-with-SHA256"
            ]
          }
        },
        "additionalProperties": false
      }
    },
    "type": "object",
    "properties": {
      "keyTypes": {
        "type": "array",
        "items": {
          "oneOf": [
            {
              "$ref": "#rsaKeyType"
            },
```

```
            {
              "$ref": "#ecKeyType"
            }
          ]
        }
      },
      "subject": {
        "$ref": "#distinguished-name"
      },
      "extensions": {
        "type": "object",
        "properties": {
          "keyUsage": {
            "type": "array",
            "items": {
              "type": "string",
              "enum": [
                "digitalSignature",
                "nonRepudiation",
                "keyEncipherment",
                "dataEncipherment",
                "keyAgreement",
                "keyCertSign",
                "cRLSign",
                "encipherOnly",
                "decipherOnly"
              ]
            }
          },
          "extendedKeyUsage": {
            "type": "array",
            "items": {
              "type": "string",
              "enum": [
                "serverAuth",
                "clientAuth",
                "codeSigning",
                "emailProtection",
                "timeStamping",
                "OCSPSigning"
              ]
            }
          },
          "subjectAltName": {
            "type": "object",
            "properties": {
              "DNS": {
                "type": "array",
```

```
                "items": {
                  "type": "string",
                  "format": "hostname"
                }
              },
              "IP": {
                "type": "array",
                "items": {
                  "oneOf": [
                    {
                      "type": "string",
                      "format": "ipv4"
                    },
                    {
                      "type": "string",
                      "format": "ipv6"
                    }
                  ]
                }
              },
              "Email": {
                "type": "array",
                "items": {
                  "type": "string",
                  "format": "email"
                }
              }
            },
            "additionalProperties": false
          }
        },
        "additionalProperties": false
      }
    },
    "additionalProperties": false
  }
```

Authors' Addresses

Yaron Sheffer
Intuit

Email: yaronf.ietf@gmail.com

Diego Lopez
Telefonica I+D

Email: diego.r.lopez@telefonica.com

Antonio Agustin Pastor Perales
Telefonica I+D

Email: antonio.pastorperales@telefonica.com

Thomas Fossati
ARM

Email: thomas.fossati@arm.com