ACME Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 21, 2017

Y. Sheffer
Intuit
D. Lopez
O. Gonzalez de Dios
Telefonica I+D
T. Fossati
Nokia

April 19, 2017

Use of Short-Term, Automatically-Renewed (STAR) Certificates to Delegate Authority over Web Sites

draft-sheffer-acme-star-00

Abstract

This memo proposes two mechanisms that work in concert to allow a third party (e.g., a content delivery network) to terminate TLS sessions on behalf of a domain name owner (e.g., a content provider).

The proposed mechanisms are:

1. An extension to the ACME protocol to enable the issuance of short-term and automatically renewed certificates, and
2. A protocol that allows a domain name owner to delegate to a third party control over a certificate that bears its own name.

It should be noted that these are in fact independent building blocks that could be used separately to solve completely different problems.

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 http://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 October 21, 2017.

Table of Contents

1. A Solution for the HTTPS CDN Use Case

A content provider that we refer to as a Domain Name Owner (DNO), has agreements in place with one or more Content Delivery Networks (CDN) that are contracted to serve its content over HTTPS. The CDN terminates the HTTPS connection at one of its edge cache servers and needs to present its clients (browsers, set-top-boxes) a certificate whose name matches the authority of the URL that is requested, i.e. that of the DNO. However, many DNOs balk at sharing their long-term private keys with another organization and, equally, CDN providers would rather not have to handle other parties' long-term secrets. This problem has been discussed at the IETF under the LURK (limited use of remote keys) title.

This document proposes a solution to the above problem that involves the use of short-term certificates with a DNO's name on them, and a scheme for handling the naming delegation from the DNO to the CDN. The generated short-term credentials are automatically renewed by an ACME Certification Authority (CA) [I-D.ietf-acme-acme] and routinely rotated by the CDN on its edge cache servers. The DNO can end the delegation at any time by simply instructing the CA to stop the automatic renewal and let the certificate expire shortly after.

Using short-term certificates makes revocation cheap and effective [Topalovic] [I-D.iab-web-pki-problems] in case of key compromise or of termination of the delegation; seamless certificate issuance and renewal enable the level of workflow automation that is expected in today's cloud environments. Also, compared to other keyless-TLS solutions [I-D.cairns-tls-session-key-interface] [I-D.erb-lurk-rsalg], the proposed approach doesn't suffer from scalability issues or increase in connection setup latency, while requiring virtually no changes to existing COTS caching software used by the CDN.

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 [RFC2119].

TODO: glossary.

3. Protocol Flow

The protocol flow can be split into two: a STAR interface, used by CDN and DNO to agree on the name delegation, and the extended ACME interface, used by DNO to obtain the short-term and automatically

renewed certificate from the CA, which is eventually consumed by the CDN. The latter is also used to terminate the delegation, if so needed.

The following subsections describe the preconditions (Section 3.1), and the three main phases of the protocol:

- Bootstrap: the CDN requests from the DNO the delegation of a specific name and in turn DNO asks an ACME CA to create the corresponding short-term and auto-renewed (STAR) certificate (Section 3.2);
- Auto-renewal: the ACME CA periodically re-issues the short-term certificate and posts it to a public URL (Section 3.3);
- Termination: the DNO (indirectly) stops name delegation by explicitly requesting the ACME CA to discontinue the automatic renewal of the certificate (Section 3.4).

3.1. Preconditions

The protocol assumes the following preconditions are met:

- A mutually authenticated channel between CDN and DNO pre-exists. This is called "STAR channel" and all STAR protocol exchanges between CDN and DNO are run over it. It provides the guarantee that requests and responses are authentic [[_1: Note that, under this assumption, the key used to authenticate the CDN to the DNO becomes a critical asset for the security of the proposed protocol, and that certain interactions (e.g., CSR submission) might require a stronger authentication mechanism. For example, stacking a further authentication factor on top of CDN's STAR key would allow to distinguish an attacker that has only managed to successfully attack the CDN's STAR key from the legitimate CDN. --tf]].
- CDN and DNO have agreed on a "CSR template" to use, including at a minimum:

  - Subject name (e.g., "somesite.DNO.com"),
  - Validity (e.g., 24 to 72 hours),
  - Requested algorithms,
  - Key length,
  - Key usage.

  The CDN is required to use this template for every CSR created under the same delegation.
- DNO has registered through the ACME interface exposed by the Certificate Authority (CA) using the usual ACME registration procedure. The DNO shall, at the registration stage, query the ACME server for the supported STAR capabilities - for example: the

minimum validity period of the issued certificate, the maximum duration of the automatic renewal process (either as a maximum number of renewal events, or as its maximum absolute life-span).

3.2. Bootstrap

CDN (STAR Client) generates a key-pair, wraps it into a Certificate Signing Request (CSR) according to the agreed CSR template, and sends it to the DNO (STAR Proxy) over the pre-established STAR channel. The DNO uses the CDN identity provided on the STAR channel to look up the CSR template that applies to the requesting CDN and decides whether or not to accept the request. (TBD: This is probably a case that would require a further authentication stage over the one provided by the mutual-authenticated STAR channel?) Assuming everything is in order, it then "forwards" the CDN request to the ACME CA by means of the usual ACME application procedure. Specifically, DNO, in its role as an ACME client, requests the CA a STAR certificate, i.e., one that:

- Has a short validity (e.g., 24 to 72 hours);
- Is automatically renewed by the CA for a certain period of time;
- Is downloadable from a (highly available) public link without requiring any special authorization.

Other than that, the ACME protocol flows as normal between DNO and CA, in particular DNO is responsible for satisfying the requested ACME challenges until the CA is willing to issue the requested certificate. The DNO is given back a unique identifier for the issued STAR certificate to be used in subsequent interaction with the CA (e.g., if the certificate needs to be terminated.)

Concurrently, a 202 response has been sent back to the CDN with an endpoint to poll for completion of the certificate generation process.

The bootstrap phase ends when the DNO obtains the OK from the ACME CA and posts the certificate's URL to the "completion endpoint" where the CDN can retrieve it. The information that is passed on to the CDN at this stage also includes details about how much time before the certificate expires can the CDN expect the replacement to be ready.

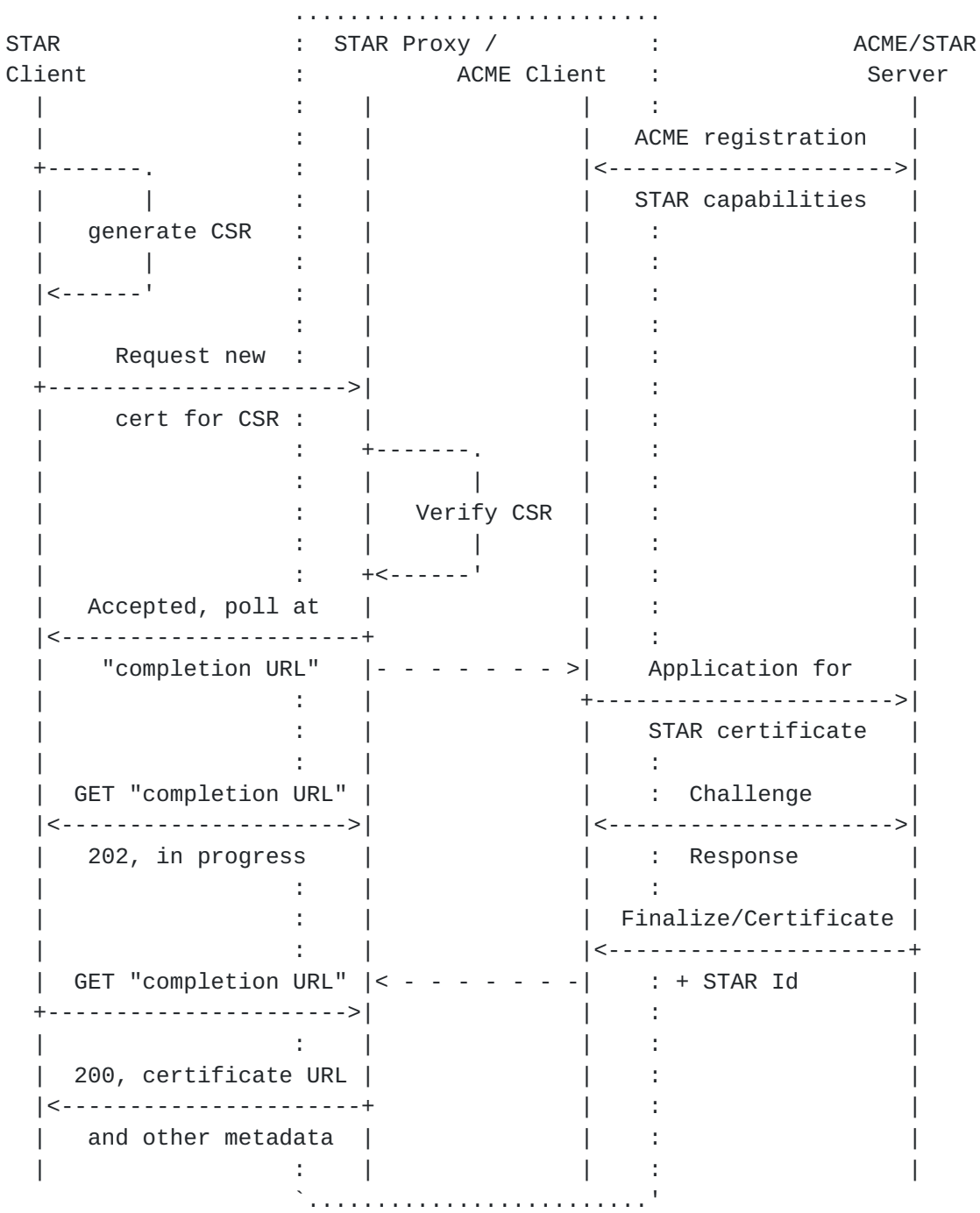

Figure 1: Bootstrap

3.3. Refresh

The CA automatically re-issues the certificate (using the same CSR) before it expires and publishes it to the URL that the CDN has come to know at the end of the bootstrap phase. The CDN downloads and installs it. This process goes on until either:

- DNO terminates the delegation, or
- Automatic renewal expires.

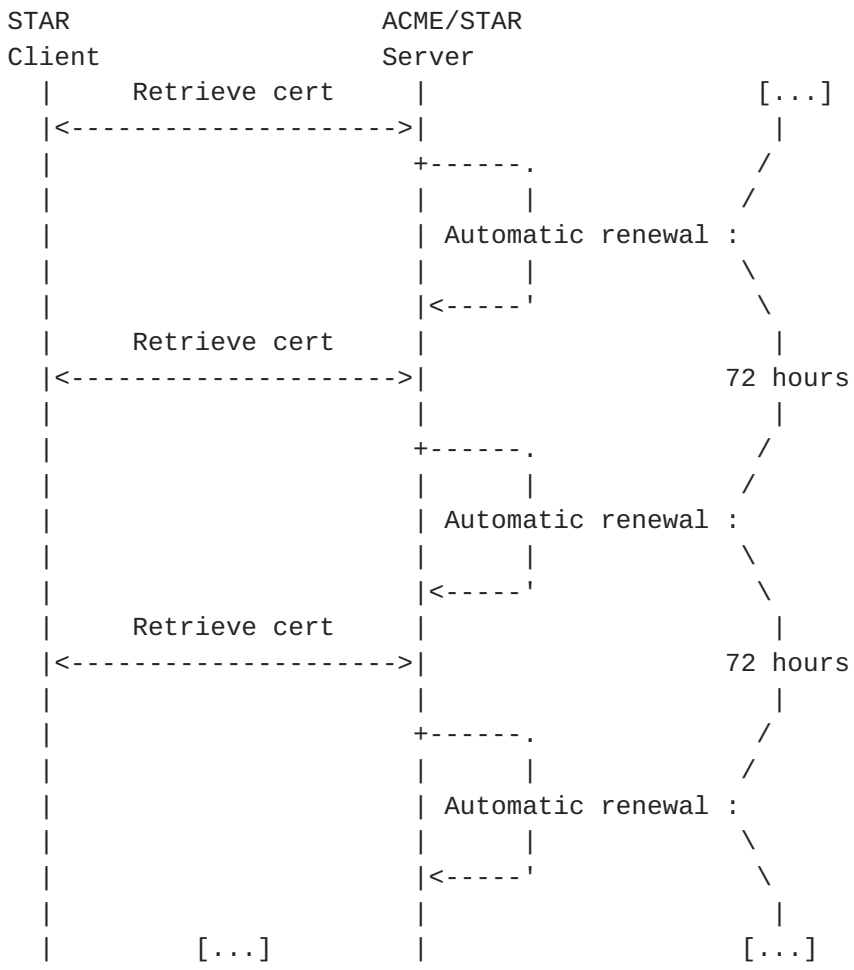

Figure 2: Auto renewal

3.4. Termination

DNO requests termination of the STAR certificate by including the previously obtained identifier in a STAR certificate termination request to the ACME interface. After CA receives and verifies the request, it shall:

- Cancel the automatic renewal process for the STAR certificate;
- Change the certificate publication resource to return an error indicating the termination of the delegation to external clients, including the CDN;

Note that it is not necessary to explicitly revoke the short-term certificate.

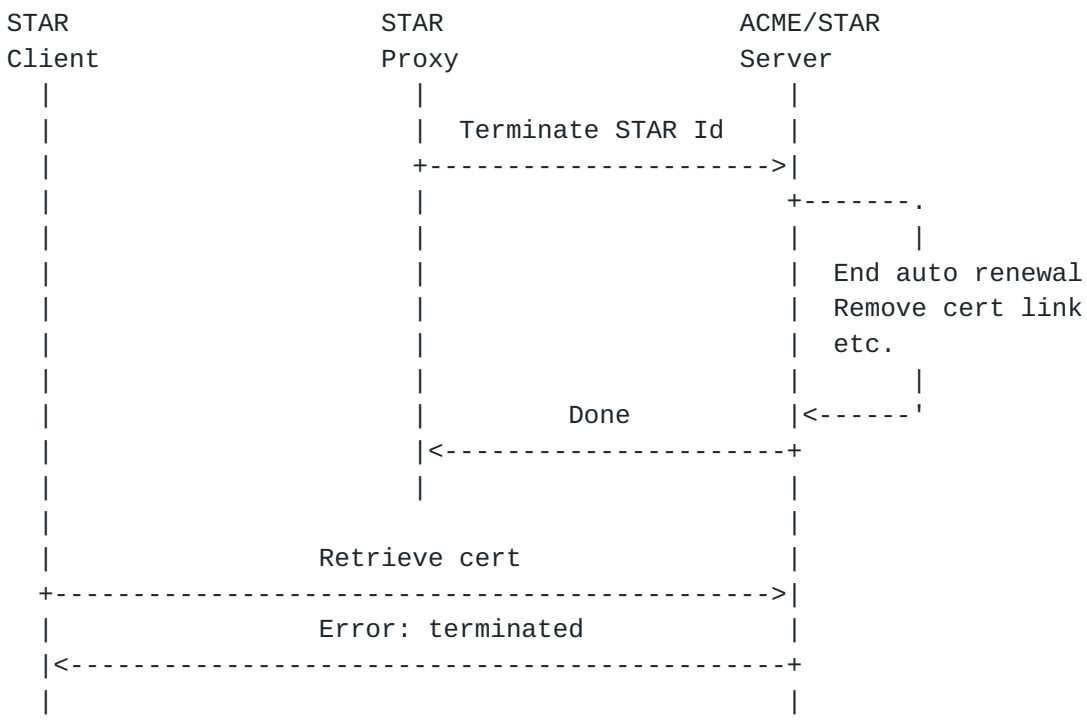

Figure 3: Termination

4. Protocol Details

This section describes the protocol's details. We start with the STAR API between the STAR Client and the STAR Proxy. Then we describe a few extensions to the ACME protocol running between the STAR Proxy and the ACME Server.

4.1. STAR API

This API allows the STAR Client to request a STAR certificate via the STAR Proxy, using a previously agreed-upon CSR template.

The API consists of a single resource, "registration". A new Registration is created with a POST and then the Registration instance is polled to obtain its details.

4.1.1. Creating a Registration

To create a registration, use:

```
POST /star/registration
Host: star-proxy.example.net
Content-Type: application/json

{
    "csr": "...", // CSR in PEM format
    "lifetime": 365 // requested registration lifetime in days,
                    // between 1 and 1095
}
```

Upon success, the call returns the new Registration resource.

```
HTTP/1.1 201 Created
Location: https://star-proxy.example.net/star/registration/567
```

4.1.2. Polling the Registration

The returned Registration can be polled until the information is available from the ACME server.

```
GET /star/registration/567
Host: star-proxy.example.net
```

In responding to poll requests while the validation is still in progress, the server MUST return a 200 (OK) response and MAY include a Retry-After header field to suggest a polling interval to the client. The Retry-After value MUST be expressed in seconds. If the Retry-After header is present, in order to avoid surprising interactions with heuristic expiration times, a max-age Cache-Control SHOULD also be present and set to a value slightly smaller than the Retry-After value.

```
HTTP/1.1 200 OK
Retry-After: 10
Cache-Control: max-age=9

{
    "status": "pending"
}
```

When the operation is successfully completed, the ACME Proxy returns:

```
HTTP/1.1 200 OK
Expires: Sun, 09 Sep 2018 14:09:00 GMT

{
    "status": "valid", // or "failed"
    "lifetime": 365, // lifetime of the registration in days,
                     //  possibly less than requested
    "certificates": "https://acme-server.example.org/certificates/A51A3"
}
```

The Expires header applies to the registration resource itself, and may be as small as a few minutes. It is unrelated to the order's lifetime which is measured in days or longer. The "certificates" attribute contains a URL of the certificate pull endpoint, see Section 4.4.

If the registration fails for any reason, the server returns a "200 OK" response, with the status as "failed" and a "reason" attribute containing a human readable error message.

4.2. Transport Security for the STAR Protocol Leg

Traffic between the STAR Client and the STAR Proxy MUST be protected with HTTPS. For interoperability, all implementations MUST support HTTP Basic Authentication [RFC7617]. However some deployments MAY prefer mutually- authenticated HTTPS or two-legged OAUTH.

4.3. ACME Extensions between Proxy and Server

We propose to extend the ACME protocol slightly, by allowing recurrent orders.

4.3.1. Extending the Order Resource

The Order resource is extended with the following attributes:

```
 {
     "recurrent": true,
     "recurrent-total-lifetime": 365, // requested lifetime of the
                                      // recurrent registration, in days
     "recurrent-certificate-validity": 7
        // requested validity of each certificate, in days
 }
```

These attributes are included in a POST message when creating the order, as part of the "payload" encoded object. They are returned when the order has been created, possibly with adjusted values.

4.3.2. Canceling a Recurrent Order

An important property of the recurrent order is that it can be cancelled by the domain name owner, with no need for certificate revocation. We use the DELETE message for that:

```
DELETE /acme/order/1 HTTP/1.1
Host: acme-server.example.org
```

Which returns:

```
HTTP/1.1 202 Deleted
```

The server MUST NOT issue any additional certificates for this Order, beyond the certificate that is available for collection at the time of deletion.

4.3.3. Indicating Support of Recurrent Orders

ACME supports sending arbitrary extensions when creating an Order, and as a result, there is no need to explicitly indicate support of this extension. The Proxy MUST verify that the "recurrent" attribute was understood, as indicated by the "recurrent" attribute included in the created Order. Since the standard ACME protocol does not allow to explicitly cancel a pending Order (the DELETE operation above is an extension), an unhappy Proxy will probably let the Order expire instead of following through with the authorization process.

4.4. Fetching the Certificates

The certificate is fetched from the certificate endpoint, as per [I-D.ietf-acme-acme], Sec. 7.4.2 "Downloading the Certificate". The server MUST include an Expires header that indicates expiry of the specific certificate. When the certificate expires, the client MAY assume that a newer certificate is already in place.

A certificate MUST be replaced by its successor at the latest 24 hours before its "Not After" time.

5. CDNI Use Cases

Members of the IETF CDNI (Content Delivery Network Interconnection) working group are interested in delegating authority over web content to CDNs. Their requirements are described in a draft [I-D.fieau-cdni-https-delegation] that compares several solutions. This section discusses two particular requirements in the context of the STAR protocol.

5.1. Multiple Parallel Delegates

In some cases the DNO would like to delegate authority over a web site to multiple CDNs. This could happen if the DNO has agreements in place with different regional CDNs for different geographical regions. STAR enables this use case naturally, since each CDN can authenticate separately to the DNO specifying its CSR, and the DNO is free to allow or deny each certificate request according to its own policy.

5.2. Chained Delegation

In other cases, a content owner (DNO) delegates some domains to a large CDN (CDN1), which in turn delegates to a smaller regional CDN, CDN2. The DNO has a contractual relationship with CDN1, and CDN1 has a similar relationship with CDN2. However DNO may not even know about CDN2.

The STAR protocol does not prevent this use case, although there is no special support for it. CDN1 can forward requests from CDN2 to DNO, and forward responses back to CDN2. Whether such proxying is allowed is governed by policy and contracts between the parties.

6. Security Considerations

- CDN's client certificate key is first order security asset and MUST be protected. Absent 2FA/MFA, an attacker that can compromise the key might be able to obtain certificates bearing DNO's identity.
- Consider collusion of two or more CDNs with contracts with the same DNO (?)

6.1. Restricting CDNs to the Delegation Mechanism

Currently there are no standard methods for the DNO to ensure that the CDN cannot issue a certificate through mechanisms other than the one described here, for the URLs under the CDN's control. For example, regardless of the STAR solution, a rogue CDN employee can use the ACME protocol (or proprietary mechanisms used by various CAs) to create a fake certificate for the DNO's content.

The best solution currently being worked on would consist of several related configuration steps:

- Make sure that the CDN cannot modify the DNS records for the domain. Typically this would mean that the content owner establishes a CNAME resource record from a subdomain into a CDN-managed domain.

- Restrict certificate issuance for the domain to specific CAs that comply with ACME. This assumes universal deployment of CAA [RFC6844] by CAs, which is not the case yet.
- Deploy ACME-specific methods to restrict issuance to a specific authorization key which is controlled by the content owner [I-D.landau-acme-caa], and/or to specific ACME authorization methods.

This solution is recommended in general, even if an alternative to the mechanism described here is used.

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.ietf-acme-acme]
Barnes, R., Hoffman-Andrews, J., and J. Kasten, "Automatic Certificate Management Environment (ACME)", draft-ietf-acme-acme-06 (work in progress), March 2017.

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

[RFC7617] Reschke, J., "The 'Basic' HTTP Authentication Scheme", RFC 7617, DOI 10.17487/RFC7617, September 2015, <http://www.rfc-editor.org/info/rfc7617>.

8.2. Informative References

[I-D.cairns-tls-session-key-interface]
Cairns, K., Mattsson, J., Skog, R., and D. Migault, "Session Key Interface (SKI) for TLS and DTLS", draft-cairns-tls-session-key-interface-01 (work in progress), October 2015.

[I-D.erb-lurk-rsalg]
Erb, S. and R. Salz, "A PFS-preserving protocol for LURK", draft-erb-lurk-rsalg-01 (work in progress), May 2016.

[I-D.fieau-cdni-https-delegation]
Fieau, F., Emile, S., and S. Mishra, "HTTPS delegation in CDNI", draft-fieau-cdni-https-delegation-01 (work in progress), March 2017.

[I-D.iab-web-pki-problems]
Housley, R. and K. O'Donoghue, "Improving the Public Key Infrastructure (PKI) for the World Wide Web", draft-iab-web-pki-problems-05 (work in progress), October 2016.

[I-D.landau-acme-caa]
Landau, H., "CA Account URI Binding for CAA Records", draft-landau-acme-caa-01 (work in progress), October 2016.

[RFC6844] Hallam-Baker, P. and R. Stradling, "DNS Certification Authority Authorization (CAA) Resource Record", RFC 6844, DOI 10.17487/RFC6844, January 2013, <http://www.rfc-editor.org/info/rfc6844>.

[Topalovic]
Topalovic, E., Saeta, B., Huang, L., Jackson, C., and D. Boneh, "Towards Short-Lived Certificates", 2012, <http://www.w2spconf.com/2012/papers/w2sp12-final9.pdf>.

Appendix A. Document History

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

A.1. draft-sheffer-acme-star-00

- Renamed draft to prevent confusion with other work in this space.
- Added an initial STAR protocol: a REST API.
- Discussion of CDNI use cases.

A.2. draft-sheffer-acme-star-lurk-00

- Initial version.

Authors' Addresses

Yaron Sheffer
Intuit

EMail: yaronf.ietf@gmail.com

Diego Lopez
Telefonica I+D

EMail: diego@telefonica.es

Oscar Gonzalez de Dios
Telefonica I+D

EMail: oscar.gonzalezdedios@telefonica.com

Thomas Fossati
Nokia

EMail: thomas.fossati@nokia.com