ACE Working Group
Internet-Draft
Intended status: Informational
Expires: October 7, 2016

S. Erdtman, Ed.

April 5, 2016

Certificate credentials for ACE framework
draft-erdtman-ace-certificate-credential-00

Abstract

This draft provides an example of how to extend the ACE framework [I-D.ietf-ace-oauth-authz], to use client and server certificates (x509), for mutual authentication. Certificate are used to establish the security context between the client and resource server. This draft is limited to transport layer security based on DTLS and it does not consider the mixed case where e.g. only the server is authenticated with a certificate.

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 7, 2016.

Table of Contents

1. Introduction

Certificates is the dominant way to secure TLS connections. TLS is mostly used to establish the identity of the Server, by connecting the DNS name to the server certificate. The client can optional be asked to provide its identity based on a certificate, but the common way is to establish the client/user identity on the application layer. In IoT space the limitation of devices makes the mixed solution with application layer and transport layer security complex. It is therefore common to do both client and server authentication on the same layer.

This draft details on how the authorisation server can be leveraged to provide the trust anchors between client and resources server when setting up a connection. The result is similar to DANE RFC 6698 [RFC6698], where the DNS server provides the trust anchor.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. x5t and x5t#256

The authorisation server is the common point in OAuth 2.0 with relation to both the client and the resource server. It needs to have a way to communicate the certificate (x509) trust anchors to the client and the resource server. Communication with the client is done with the Token endpoint where the client gets the token. The resource server can either get the trust anchor information as part of a self contained token or as a new attribute from the introspection endpoint.

For the transport of the two new attributes are defined, x5t and x5t#256. These are defined and registered in the appropriate IANA registry

The attributes are defined as in defined in RFC 7519 [RFC7519] a base64url encoded thumbprint of the x509 certificate. In this context the thumbprint is used to identify the client-, server-, issuer or root certificate of the server and the connecting client. In cases where the transport is CBOR based the encoding of these parameters is CBOR byte string, without the base64url encoding.

2.1. CBOR types

Validation of the trust chain MUST be done according to PKIX [TODO insert reference] both on client and server side. With the exception that the traversing of the certificate chain stops when a certificate with the matching thumbprint is found.

If the x5t value in a token is 88234efc198f455848fa728fbde3ce549be1e7b4, the server first validates the user certificate but does not stop there. It continues through the chain to the Issuer CA certificate where it finds a match to the thumbprint. With the match it does not continue up to the Root CA. The x5t can be any of the certificates in the chain. How the client and resource sever obtains the chain is out of scope for this specification.

- Root CA (x5t = c16aab9fe3288df0fb8fc1d24990a300b6b8f299)
  - Issuer CA (x5t = 88234efc198f455848fa728fbde3ce549be1e7b4)
    - Server/Client Cert (x5t = 10f7158b7813470820325004d4637f7287dc1f63)

Figure 1: Certificate chain example

2.2. CBOR types

When using CBOR encoding, values must be encoded with major type according to table.

Major Type	Key
2	x5t
2	x5t#256

Figure 2: CWT CBOR key values

3. IANA Considerations

This section contains registrations to the different registries where the parameters are be used.

3.1. Token endpoint

The x5t or x5t#256 parameter is included in the token request and returned in the token response. In the token response it is used to validate the server certificate provided in the DTLS handshake between client and resource server. In the token request it is to be included in the access token or the token introspection response. To aid the resource server in validating the client certificate in the DTLS handshake between client and resource server.

- Parameter name: "x5t"
- Parameter usage location: token response and token request
- Change Controller: IESG
- Specification Document(s): this document

- Parameter name: "x5t#256"
- Parameter usage location: token response and token request
- Change Controller: IESG
- Specification Document(s): this document

3.1.1. CBOR Mappings

When token response is CBOR encoded according the the ACE framework the following key values MUST be used.

TODO register values in ACE framework registry

- Claim name: "x5t"
- CBOR key value: X

- CBOR major type: 2
- Change Controller: IESG
- Specification Document(s): this document

- Claim name: "x5t#256"
- CBOR key value: Y
- CBOR major type: 2
- Change Controller: IESG
- Specification Document(s): this document

3.2. JWT and CWT

When the x5t or x5t#256 parameter is included in the token it is used to validate the client certificate provided in the DTLS handshake.

- Claim Name: "x5t"
- Claim Description: X.509 certificate SHA-1 thumbprint defined as in RFC 7519 [RFC7519] but used to validate the client certificate provided in the DTLS handshake.
- Change Controller: IESG
- Specification Document(s): this document

- Claim Name: "x5t#S256"
- Claim Description: X.509 certificate SHA-256 thumbprint defined as in RFC 7519 [RFC7519] but used to validate the client certificate provided in the DTLS handshake.
- Change Controller: IESG
- Specification Document(s): this document

3.2.1. CWT CBOR key registration

When encoded in a CWT following key values MUST be used.

TODO register key values in CWT registry

- Claim name: "x5t"
- CBOR key value: 8
- CBOR major type: 2
- Change Controller: IESG
- Specification Document(s): this document

- Claim name: "x5t#256"
- CBOR key value: 9
- CBOR major type: 2
- Change Controller: IESG
- Specification Document(s): this document

3.3. Token Introspection

When the x5t or x5t#256 parameter is returned in the introspection response it is used to validate the client certificate provided in the DTLS handshake.

- Name: "x5t"
- Description: X.509 certificate SHA-1 thumbprint defined as in RFC 7519 [RFC7519] but used to validate client certificate provided in the DTLS handshake.
- Change Controller: IESG
- Specification Document(s): this document

- Name: "x5t#S256"
- Description: X.509 certificate SHA-256 thumbprint defined as in RFC 7519 [RFC7519] but used to validate the client certificate provided in the DTLS handshake.
- Change Controller: IESG
- Specification Document(s): this document

3.3.1. CBOR Mappings

When token response is CBOR encoded according the the ACE framework the following key values MUST be used.

TODO register values in ACE framework registry

- Claim name: "x5t"
- CBOR key value: X
- CBOR major type: 2
- Change Controller: IESG
- Specification Document(s): this document

- Claim name: "x5t#256"
- CBOR key value: Y
- CBOR major type: 2
- Change Controller: IESG
- Specification Document(s): this document

4. Acknowledgements

TBD

5. Security Considerations

TBD

6. References

6.1. Normative References

[I-D.ietf-ace-oauth-authz]
Seitz, L., Selander, G., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "Authorization for the Internet of Things using OAuth 2.0", draft-ietf-ace-oauth-authz-01 (work in progress), February 2016.

[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>.

[RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, DOI 10.17487/RFC6749, October 2012, <http://www.rfc-editor.org/info/rfc6749>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <http://www.rfc-editor.org/info/rfc7519>.

[RFC7662] Richer, J., Ed., "OAuth 2.0 Token Introspection", RFC 7662, DOI 10.17487/RFC7662, October 2015, <http://www.rfc-editor.org/info/rfc7662>.

6.2. Informative References

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698, August 2012, <http://www.rfc-editor.org/info/rfc6698>.

Appendix A. Example

This sections provides a non normative examples of the flow and the different connections

TBD

Author's Address

Samuel Erdtman (editor)
SE

Phone: +46702691499
Email: samuel@erdtman.se