ACE Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 25, 2021

F. Palombini
Ericsson AB

September 21, 2020

Identifier Negotiation for the OSCORE Profile of ACE
draft-palombini-ace-oscore-profile-id-00

Abstract

This document defines a mechanism to negotiate OSCORE security material identifiers for the OSCORE profile of ACE.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

In the OSCORE profile, the client and resource server receive the OSCORE Sender and Recipient Identifiers from the AS. This has some limitations, especially if the OSCORE profile is used in conjuction with other mechamisms that also derive identifiers, in which case either collisions would happen, or longer identifiers need to be used as a result. This document describes a way to negotiate the identifiers so that collisions does not happen even if other authentication mechanisms are used.

1.1. 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.

Readers are expected to be familiar with the terms and concepts described in [I-D.ietf-ace-oauth-authz] [I-D.ietf-ace-oscore-profile], such as Authorization Server (AS) and Resource Server (RS).

Readers are expected to be familiar with the terms and concepts described in [RFC8613], especially on the use of Sender, Recipient and Context Identifiers.

2. Background

TODO: introduce OSCORE Sender and Recipient Identifiers and how they are used in OSCORE.

The OSCORE profile specifies that the AS assigns and sends the OSCORE Sender and Recipient Identifiers to both Client and RS, together with the rest of the input material to derive the complete OSCORE Security Context. That is done by including these identifiers in the Access Token and Access Information response to the Client. The access token containing these identifiers is also forwarded to the RS by the Client.

```
             C                            RS                   AS
             |                            |                     |
             | ----- POST /token  ----------------------------> |
             |                            |                     |
             | <---------------------------- Access Token ----- |
             |                           + Access Information   |
             | ---- POST /authz-info ---> |                     |
             |     (access_token, N1)     |                     |
             |                            |                     |
             | <--- 2.01 Created (N2) --- |                     |
             |                            |                     |
           /Sec Context             /Sec Context                |
             Derivation/              Derivation/               |
             |                            |                     |
             | ---- OSCORE Request -----> |                     |
             |                            |                     |
             |                    /proof-of-possession          |
             |                    Sec Context storage/          |
             |                            |                     |
             | <--- OSCORE Response ----- |                     |
             |                            |                     |
          /proof-of-possession            |                     |
          Sec Context storage/            |                     |
             |                            |                     |
             | ---- OSCORE Request -----> |                     |
             |           ...              |                     |
```

Figure 1: OSCORE Profile Overview

This works with a number of requirements: the OSCORE profile states that if other authentication mechanisms are used to set up OSCORE between the same client and RS, that do not rely on an AS assigning identifiers, collisions may happen and need to be mitigated. Such mitigation mechanism also need to be used if a different AS (not sinchronized with the first AS) or authentication protocol is used to

set up OSCORE between the same RS and other clients. A mitigation example would be to use distinct namespaces of context identifiers for different authentication mechanisms or authentication servers. Another solution would be to use longer random identifiers. A third possible solution, acceptable if collisions are not expected to be numerous, would be to rely on trial and error of security contexts when receiving a message.

These solutions have the drawback of requiring longer identifiers to be used in general, which leads to larger message sizes, or additional processing on the RS.

This document specifies a different mechanism to assign identifiers that works on top of the current OSCORE profile, and that allows to set up identifiers without collisions, even when other authentication mechanisms or non-syncrhonized AS are used.

3. Identifiers Negotiation

This section details the message exchange.

3.1. C-to-AS:

3.2. C-to-RS: POST to authz-info endpoint

The client generates its own Recipient Id for the OSCORE Security Context that it is establishing with the RS. By generating its own Recipient Id, the client makes sure that it does not collide with any other Recipient Identifiers stored in memory. The client posts it together with what is described in Section 4.1 of [I-D.ietf-ace-oscore-profile]. The Client includes the Recipient Id in the POST to authz-info request, as a ace_client_recipientid parameter, as registered in Section 5.1 and Section 5.2.

When receiving the POST to authz-info request including the ace_client_recipientid parameter, the RS MUST set its own Sender Identifier to the value of the ace_client_recipientid and discard any ServerId present in the access token.

3.3. RS-to-C: 2.01 (Created)

The RS generates its own Recipient Id for the OSCORE Security Context that it is establishing with the client. The Recipient Id MUST be different than the ace_client_recipientid received from the client. By generating its own Recipient Id, the RS makes sure that it does not collide with any other Recipient Identifiers stored in memory. The RS sends it to the Client together with what is described in Section 4.2 of [I-D.ietf-ace-oscore-profile]. The RS includes the

Recipient Id in the 2.01 (Created) response, as a ace_server_recipientid parameeter, as registered in Section 5.1 and Section 5.2.

When receiving the response including the ace_server_recipientid parameter, the Client MUST set its own Sender Identifier to the value of the ace_server_recipientid and discard any ClientId present in the access token.

3.4. Not Supported

If the RS does not support this specification, and the client sends its Recipient Id in the ace_client_recipientid, the server will not recognize the parameter and either respond with an error response or discard the parameter.

If the RS replies with an error response or if the RS replies with a 2.01 (Created) not including the ace_server_recipientid parameter the Client MUST assume the server uses the identifiers in the token and do the same.

TODO: so it is possible for anybody in the middle to revert back to OSCORE profile, without this addition, and therefore create collisions without identifiers.

4. Security Considerations

TODO

5. IANA Considerations

This document has the following actions for IANA.

5.1. OAuth Parameters Registry

The following registrations are done for the OAuth ParametersRegistry following the procedure specified in section 11.2 of [RFC6749]:

o Parameter name: ace_client_recipientid o Parameter usage location: client-rs request o Change Controller: IESG o Specification Document(s): [[This specification]]

o Parameter name: ace_server_recipientid o Parameter usage location: rs-client response o Change Controller: IESG o Specification Document(s): [[This specification]]

5.2. OAuth Parameters CBOR Mappings Registry

The following registrations are done for the OAuth Parameters CBOR Mappings Registry following the procedure specified in section 8.9 of [I-D.ietf-ace-oauth-authz]:

* Name: ace_client_recipientid
* CBOR Key: TBD (range -256 to 255)
* Value Type: byte string
* Reference: \[\[This specification\]\]

* Name: ace_server_recipientid
* CBOR Key: TBD (range -256 to 255)
* Value Type: byte string
* Reference: \[\[This specification\]\]

Acknowledgments

This document was started from comments and discussion with the following individuals: John Mattsson, Jim Schaad, Goeran Selander.

7. Normative References

[I-D.ietf-ace-oauth-authz]
Seitz, L., Selander, G., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "Authentication and Authorization for Constrained Environments (ACE) using the OAuth 2.0 Framework (ACE-OAuth)", draft-ietf-ace-oauth-authz-35 (work in progress), June 2020.

[I-D.ietf-ace-oscore-profile]
Palombini, F., Seitz, L., Selander, G., and M. Gunnarsson, "OSCORE profile of the Authentication and Authorization for Constrained Environments Framework", draft-ietf-ace-oscore-profile-11 (work in progress), June 2020.

[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>.

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

[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>.

[RFC8613] Selander, G., Mattsson, J., Palombini, F., and L. Seitz, "Object Security for Constrained RESTful Environments (OSCORE)", RFC 8613, DOI 10.17487/RFC8613, July 2019, <https://www.rfc-editor.org/info/rfc8613>.

Author's Address

Francesca Palombini
Ericsson AB
Torshamnsgatan 23
Kista SE-16440 Stockholm
Sweden

Email: francesca.palombini@ericsson.com