CoRE Working Group
Internet-Draft
Updates: 8613 (if approved)
Intended status: Standards Track
Expires: August 23, 2021

R. Hoeglund
M. Tiloca
RISE AB

February 19, 2021

AEAD Key Usage Limits in OSCORE
draft-hoeglund-core-oscore-key-limits-00

Abstract

Object Security for Constrained RESTful Environments (OSCORE) uses AEAD algorithms to ensure confidentiality and integrity of exchanged messages. Due to known issues allowing forgery attacks against AEAD algorithms, limits should be followed on the number of times a specific key is used for encryption or decryption. This document defines how two peers using OSCORE must take these limits into account and what steps they must take to preserve the security of their communications. Therefore, this document updates RFC8613.

Table of Contents

1. Introduction

Object Security for Constrained RESTful Environments (OSCORE) [RFC8613] provides end-to-end protection of CoAP [RFC7252] messages at the application-layer, ensuring message confidentiality and integrity, replay protection, as well as binding of response to request between a sender and a recipient.

In particular, OSCORE uses AEAD algorithms to provide confidentiality and integrity of messages exchanged between two peers. Due to known issues allowing forgery attacks against AEAD algorithms, limits should be followed on the number of times a specific key is used to perform encryption or decryption [I-D.irtf-cfrg-aead-limits].

Should these limits be exceeded, an adversary may break the security properties of the AEAD algorithm, such as message confidentiality and integrity, e.g. by performing a message forgery attack. The original OSCORE specification [RFC8613] does not consider such limits.

This document updates [RFC8613] and defines when a peer must stop using an OSCORE Security Context shared with another peer, due to the reached key usage limits. When this happens, the two peers have to establish a new Security Context with new keying material, in order to continue their secure communication with OSCORE.

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 related to the CoAP [RFC7252] and OSCORE [RFC8613] protocols.

2. Problem Overview

The OSCORE security protocol [RFC8613] uses AEAD algorithms to provide integrity and confidentiality of messages, as exchanged between two peers sharing an OSCORE Security Context.

When processing messages with OSCORE, each peer should follow specific limits as to the number of times it uses a specific key. This applies separately to the Sender Key used to encrypt outgoing messages, and to the Recipient Key used to decrypt and verify incoming protected messages.

Exceeding these limits may allow an adversary to break the security properties of the AEAD algorithm, such as message confidentiality and integrity, e.g. by performing a message forgery attack.

The following refers to the two parameters 'q' and 'v' introduced in [I-D.irtf-cfrg-aead-limits], to use when deploying an AEAD algorithm.

o 'q': this parameter has as value the number of messages protected with a specific key, i.e. the number of times the AEAD algorithm has been invoked to encrypt data with that key.

o 'v': this parameter has as value the number of alleged forgery attempts that have been made against a specific key, i.e. the amount of failed decryptions that has been done with the AEAD algorithm for that key.

When a peer uses OSCORE:

o The key used to protect outgoing messages is its Sender Key, in its Sender Context.

o The key used to decrypt and verify incoming messages is its Recipient Key, in its Recipient Context.

Both keys are derived as part of the establishment of the OSCORE Security Context, as defined in Section 3.2 of [RFC8613].

As mentioned above, exceeding specific limits for the 'q' or 'v' value can weaken the security properties of the AEAD algorithm used, thus compromising secure communication requirements.

Therefore, in order to preserve the security of the used AEAD algorithm, OSCORE has to observe limits for the 'q' and 'v' values, throughout the lifetime of the used AEAD keys.

2.1. Limits for 'q' and 'v'

Recommendations for setting limits for the maximum 'q' and 'v' value are defined in [I-D.irtf-cfrg-aead-limits].

In particular, Figure 1 shows the limits given for AES-CCM-16-64-128, which is the mandatory to implement AEAD algorithm for OSCORE.

```
q <= sqrt((p * 2^126) / l^2)

v * 2^64 + (2l * (v + q))^2 <= p * 2^128
```

Figure 1: AES-CCM-16-64-128 limits

Considering the values p_q = 2^-60 and p_v = 2^-57 defined in [I-D.ietf-tls-dtls13], as well as l=1024, this gives the following values for the limits of 'q' and 'v'.

```
q <= sqrt(((2^-60) * 2^126) / 1024^2)

q <= 2^23

v * 2^64 + (2*1024 * (v + 2^23))^2 <= 2^-57 * 2^128

v <= 112
```

3. Additional Information in the Security Context

In addition to what defined in Section 3.1 of [RFC8613], the OSCORE Security Context MUST also include the following information.

The Sender Context is extended to include the following parameters.

- 'count_q': a non-negative integer counter, keeping track of the current 'q' value for the Sender Key. At any time, 'count_q' has as value the number of messages that have been encrypted using the

Sender Key. The value of 'count_q' is set to 0 when establishing the Sender Context.

- 'limit_q': a non-negative integer, which specifies the highest value that 'count_q' is allowed to reach, before stopping using the Sender Key to process outgoing messages.

  The value of 'limit_q' depends on the AEAD algorithm specified in the Common Context, considering the properties of that algorithm. The value of 'limit_q' is determined according to Section 3.

The Recipient Context is extended to include the following parameters.

- 'count_v': a non-negative integer counter, keeping track of the current 'v' value for the Recipient Key. At any time, 'count_v' has as value the number of failed decryptions occurred on incoming messages using the Recipient Key. The value of 'count_v' is set to 0 when establishing the Recipient Context.

- 'limit_v': a non-negative integer, which specifies the highest value that 'count_v' is allowed to reach, before stopping using the Recipient Key to process incoming messages.

  The value of 'limit_v' depends on the AEAD algorithm specified in the Common Context, considering the properties of that algorithm. The value of 'limit_v' is determined according to Section 3.

4. OSCORE Messages Processing

In order to keep track of the 'q' and 'v' values and ensure that AEAD keys are not used beyond reaching their limits, the processing of OSCORE messages is extended as defined in this section.

In particular, the processing of OSCORE messages follows the steps outlined in Section 8 of [RFC8613], with the additions defined below.

4.1. Protecting a Request or a Response

Before encrypting the COSE object using the Sender Key, the 'count_q' counter MUST be incremented.

If 'count_q' exceeds the 'limit_q' limit, the message processing MUST be aborted. From then on, the Sender Key MUST NOT be used to encrypt further messages.

4.2. Verifying a Request or a Response

If the decryption and verification of the COSE object using the Recipient Key fails, the 'count_v' counter MUST be incremented.

After 'count_v' has exceeded the 'limit_v' limit, incoming messages MUST NOT be decrypted and verified using the Recipient Key, and their processing MUST be aborted.

5. Methods for Rekeying OSCORE

Before the limit of 'q' or 'v' has been reached for an OSCORE Security Context, the two peers have to establish a new OSCORE Security Context, in order to continue using OSCORE for secure communication.

In practice, the two peers have to establish new Sender and Recipient Keys, as the keys actually used by the AEAD algorithm. When this happens, both peers reset their 'count_q' and 'count_v' values to 0 (see Section 3).

Currently, a number of ways exist to accomplish this.

o The two peers can run the procedure defined in Appendix B.2 of [RFC8613]. That is, the two peers exchange three or four messages, protected with temporary Security Contexts adding randomness to the ID Context.

   As a result, the two peers establish a new OSCORE Security Context with new ID Context, Sender Key and Recipient Key, while keeping the same OSCORE Master Secret and OSCORE Master Salt from the old OSCORE Security Context.

   This procedure does not require any additional components to what OSCORE already provides, and it does not provide perfect forward secrecy.

o The two peers can run the OSCORE profile [I-D.ietf-ace-oscore-profile] of the Authentication and Authorization for Constrained Environments (ACE) Framework [I-D.ietf-ace-oauth-authz].

   When a CoAP client uploads an Access Token to a CoAP server as an access credential, the two peers also exchange two nonces. Then, the two peers use the two nonces together with information provided by the ACE Authorization Server that issued the Access Token, in order to derive an OSCORE Security Context.

This procedure does not provide perfect forward secrecy.

- The two peers can run the EDHOC key exchange protocol based on Diffie-Hellman and defined in [I-D.ietf-lake-edhoc], in order to establish a pseudo-random key in a mutually authenticated way.

  Then, the two peers can use the established pseudo-random key to derive external application keys. This allows the two peers to securely derive especially an OSCORE Master Secret and an OSCORE Master Salt, from which an OSCORE Security Context can be established.

  This procedure additionally provides perfect forward secrecy.

Manually updating the OSCORE Security Context at the two peers should be a last resort option, and it might often be not practical or feasible.

It is RECOMMENDED that the peer initiating the rekeying procedure starts it before reaching the 'q' or 'v' limits. Otherwise, the AEAD keys possibly to be used during the rekeying procedure itself may already be or become invalid before the rekeying is completed, which may prevent a successful establishment of the new OSCORE Security Context altogether.

6. Security Considerations

This document mainly covers security considerations about using AEAD keys in OSCORE and their usage limits, in addition to the security considerations of [RFC8613].

Depending on the specific rekeying procedure used to establish a new OSCORE Security Context, the related security considerations also apply.

TODO: Add more considerations.

7. IANA Considerations

This document has no actions for IANA.

Acknowledgments

The authors sincerely thank Christian Amsuess, John Mattsson and Goeran Selander for the initial discussions that allowed shaping this document.

The work on this document has been partly supported by VINNOVA and the Celtic-Next project CRITISEC; and by the H2020 project SIFIS-Home (Grant agreement 952652).

9. References

9.1. Normative References

[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>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <https://www.rfc-editor.org/info/rfc7252>.

[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>.

9.2. Informative 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-36 (work in progress), November 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-15 (work in progress), January 2021.

[I-D.ietf-lake-edhoc] Selander, G., Mattsson, J., and F. Palombini, "Ephemeral Diffie-Hellman Over COSE (EDHOC)", draft-ietf-lake-edhoc-03 (work in progress), December 2020.

[I-D.ietf-tls-dtls13]
Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", draft-ietf-tls-dtls13-40 (work in progress), January 2021.

[I-D.irtf-cfrg-aead-limits]
Guenther, F., Thomson, M., and C. Wood, "Usage Limits on AEAD Algorithms", draft-irtf-cfrg-aead-limits-01 (work in progress), September 2020.

Authors' Addresses

Rikard Hoeglund
RISE AB
Isafjordsgatan 22
Kista SE-16440 Stockholm
Sweden

Email: rikard.hoglund@ri.se

Marco Tiloca
RISE AB
Isafjordsgatan 22
Kista SE-16440 Stockholm
Sweden

Email: marco.tiloca@ri.se