Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: November 9, 2019

N. Madden
ForgeRock

May 8, 2019

Public Key Authenticated Encryption for JOSE: ECDH-1PU draft-madden-jose-ecdh-1pu-00

Abstract

This document describes the ECDH-1PU public key authenticated encryption algorithm for JWE. The algorithm is similar to the existing ECDH-ES encryption algorithm, but adds an additional ECDH key agreement between static keys of the sender and recipient. This additional step allows the recipient to be assured of sender authenticity without requiring a nested signed-then-encrypted message structure. The mode is also a useful building block for constructing interactive handshake protocols on top of JOSE.

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 November 9, 2019.

Copyright Notice

Table of Contents

1. Introduction

JSON Object Signing and Encryption (JOSE) defines a number of encryption (JWE) [RFC7516] and digital signature (JWS) [RFC7515] algorithms. When symmetric cryptography is used, JWE provides authenticated encryption that ensures both confidentiality and sender authentication. However, for public key cryptography the existing JWE encryption algorithms provide only confidentiality and some level of ciphertext integrity. When sender authentication is required, users must resort to nested signed-then-encrypted structures, which increases the overhead and size of resulting messages. This document describes an alternative encryption algorithm called ECDH-1PU that provides public key authenticated encryption, allowing the benefits of authenticated encryption to be enjoyed for public key JWE as it currently is for symmetric cryptography.

ECDH-1PU is based on the One-Pass Unified Model for Elliptic Curve Diffie-Hellman key agreement described in [NIST.800-56A].

The advantages of public key authenticated encryption with ECDH-1PU compared to using nested signed-then-encrypted documents include the following:

- The resulting message size is more compact as an additional layer of headers and base64url-encoding is avoided.

- The same primitives are used for both confidentiality and authenticity, providing savings in code size for constrained environments.

- The generic composition of signatures and public key encryption involves a number of subtle details that are essential to security [PKAE]. Providing a dedicated algorithm for public key authenticated encryption reduces complexity for users of JOSE libraries.

- ECDH-1PU provides only authenticity and not the stronger security properties of non-repudiation or third-party verifiability. This can be an advantage in applications where privacy, anonymity, or plausible deniability are goals.

1.1. Requirements 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 [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Key Agreement with Elliptic Curve Diffie-Hellman Ephemeral-Static Static-Static (ECDH-1PU)

This section defines the specifics of key agreement with Elliptic Curve Diffie-Hellman Ephemeral-Static Static-Static, in combination with the one-step KDF, as defined in Section 5.8.2.1 of [NIST.800-56A] using the Concatenation Format of Section 5.8.2.1.1. This is identical to the ConcatKDF function used by the existing JWE ECDH-ES algorithm defined in Section 4.6 of [RFC7518]. As for ECDH-ES, the key agreement result can be used in one of two ways:

1. directly as the Content Encryption Key (CEK) for the "enc" algorithm, in the Direct Key Agreement mode, or

2. as a symmetric key used to wrap the CEK with the "A128KW", "A192KW", or "A256KW" algorithms, in the Key Agreement with Key Wrapping mode.

A new ephemeral public key value MUST be generated for each key agreement operation.

In Direct Key Agreement mode, the output of the KDF MUST be a key of the same length as that used by the "enc" algorithm. In this case, the empty octet sequence is used as the JWE Encrypted Key value. The

"alg" (algorithm) Header Parameter value "ECDH-1PU" is used in Direct Key Agreement mode.

In Key Agreement with Key Wrapping mode, the output of the KDF MUST be a key of the length needed for the specified key wrapping algorithm. In this case, the JWE Encrypted Key is the CEK wrapped with the agreed-upon key.

The following "alg" (algorithm) Header Parameter values are used to indicate the JWE Encrypted Key is the result of encrypting the CEK using the result of the key agreement algorithm as the key encryption key for the corresponding key wrapping algorithm:

"alg" Param Value	Key Management Algorithm
ECDH-1PU+A128KW	ECDH-1PU using Concat KDF and CEK wrapped with "A128KW"
ECDH-1PU+A192KW	ECDH-1PU using Concat KDF and CEK wrapped with "A192KW"
ECDH-1PU+A256KW	ECDH-1PU using Concat KDF and CEK wrapped with "A256KW"

2.1. Header Parameters used for ECDH Key Agreement

The "epk" (ephemeral public key), "apu" (Agreement PartyUInfo), and "apv" (Agreement PartyVInfo) header parameters are used in ECDH-1PU exactly as defined in Section 4.6.1 of [RFC7518].

When no other values are supplied, it is RECOMMENDED that the producer software initializes the "apu" header to the base64url-encoding of the SHA-256 hash of the concatenation of the sender's static public key and the ephemeral public key, and the "apv" header to the base64url-encoding of the SHA-256 hash of the recipient's static public key. This ensures that all keys involved in the key agreement are cryptographically bound to the derived keys.

2.2. Key Derivation for ECDH-1PU Key Agreement

The key derivation process derives the agreed-upon key from the shared secret Z established through the ECDH algorithm, per Section 6.2.1.2 of [NIST.800-56A]. For the NIST prime order curves "P-256", "P-384", and "P-521", the ECC CDH primitive for cofactor Diffie-Hellman defined in Section 5.7.1.2 of [NIST.800-56A] is used (taking note that the cofactor for all these curves is 1). For

curves "X25519" and "X448" the appropriate ECDH primitive from Section 5 of [RFC7748] is used.

Key derivation is performed using the one-step KDF, as defined in Section 5.8.1 and Section 5.8.2.1 of [NIST.800-56A] using the Concatenation Format of Section 5.8.2.1.1, where the Auxilary Function H is SHA-256. The KDF parameters are set as follows:

Z This is set to the representation of the shared secret Z as an octet sequence. As per Section 6.2.1.2 of [NIST.800-56A] Z is the concatenation of Ze and Zs, where Ze is the shared secret derived from applying the ECDH primitive to the sender's ephemeral private key and the recipient's static public key. Zs is the shared secret derived from applying the ECDH primitive to the sender's static private key and the recipient's static public key.

keydatalen This is set to the number of bits in the desired output key. For "ECDH-1PU", this is the length of the key used by the "enc" algorithm. For "ECDH-1PU+A128KW", "ECDH-1PU+A192KW", and "ECDH-1PU+A256KW", this is 128, 192, and 256, respectively.

AlgorithmID The AlgorithmID values is of the form Datalen || Data, where Data is a variable-length string of zero or more octets, and Datalen is a fixed-length, big-endian 32-bit counter that indicates the length (in octets) of Data. In the Direct Key Agreement case, Data is set to the octets of the ASCII representation of the "enc" Header Parameter value. In the Key Agreement with Key Wrapping case, Data is set to the octets of the ASCII representation of the "alg" (algorithm) Header Parameter value.

PartyUInfo The PartyUInfo value is of the form Datalen || Data, where Data is a variable-length string of zero or more octets, and Datalen is a fixed-length, big-endian 32-bit counter that indicates the length (in octets) of Data. If an "apu" (agreement PartyUInfo) Header Parameter is present, Data is set to the result of base64url decoding the "apu" value and Datalen is set to the number of octets in Data. Otherwise, Datalen is set to 0 and Data is set to the empty octet sequence.

PartyVInfo The PartyVInfo value is of the form Datalen || Data, where Data is a variable-length string of zero or more octets, and Datalen is a fixed-length, big-endian 32-bit counter that indicates the length (in octets) of Data. If an "apv" (agreement PartyVInfo) Header Parameter is present, Data is set to the result of base64url decoding the "apv" value and Datalen is set to the number of octets in Data. Otherwise, Datalen is set to 0 and Data is set to the empty octet sequence.

SuppPubInfo This is set to the keydatalen represented as a 32-bit big-endian integer.

SuppPrivInfo This is set to the empty octet sequence.

Applications need to specifiy how the "apu" and "apv" Header Parameters are used for that application. The "apu" and "apv" values MUST be distinct, when used. Applications wishing to conform to [NIST.800-56A] need to provide values that meet the requirements of that doucument, e.g., by using values that identify the producer and consumer.

3. Two-way interactive handshake

A party that has received a JWE encrypted with ECDH-1PU MAY reply to that message by creating a new JWE using ECDH-1PU, but using the ephemeral public key ("epk") from the first message as if it was the originating party's static public key. In this case, key agreement proceeds exactly as for Section 2, but with the originator's ephemeral public key used as the recipient (Party V) static public key. The "alg" (algorithm) Header Parameter in the response MUST be identical to the "alg" Header Parameter of the original message.

The value of the "apu" (Agreement PartyUInfo) Header Parameter value from the original message SHOULD be reflected as the "apv" (Agreement PartyVInfo) Header Parameter value in the new message. Applications need to specify how the new "apu" Header Parameter should be constructed.

If a "kid" claim was included in the ephemeral public key of the original message, then a "kid" Header Parameter with the same value MUST be included in the reply JWE.

After the initial message and a reply have been exchanged, the two parties may communicate using the derived key from the second message as the encryption key for any number of additional messages. When ECDH-1PU is used in Direct Key Agreement mode, then subsequent messages using the derived key MUST be encrypted using the "dir" (Direct) JWE algorithm. When used in Key Agreement with Key Wrapping mode, subsequent messages using the derived key MUST be encrypted using the associated Key Wrapping algorithm, as shown in the following table:

ECDH-1PU "alg" Param Value	Subsequent "alg" Param Value
ECDH-1PU+A128KW	A128KW
ECDH-1PU+A192KW	A192KW
ECDH-1PU+A256KW	A256KW

4. IANA considerations

This section registers JWE algorithms as per the registry established in [RFC7518].

4.1. ECDH-1PU

Algorithm Name: "ECDH-1PU"
Algorithm Description: ECDH One-Pass Unified Model using Concat KDF
Algorithm Usage Location(s): "alg"
JOSE Implementation Requirements: Optional
Change Controller: IESG
Specification Document(s): Section 2
Algorithm Analysis Document(s): [NIST.800-56A] (Section 7.3), [PKAE]

5. Security Considerations

The security considerations of [RFC7518] relevant to ECDH-ES also apply to this specification.

The security considerations of [NIST.800-56A] apply here.

When performing an ECDH key agreement between a static private key and any untrusted public key, care should be taken to ensure that the public key is a valid point on the same curve as the private key. Failure to do so may result in compromise of the static private key. For the NIST curves P-256, P-384, and P-521, appropriate validation routines are given in Section 5.6.2.3.3 of [NIST.800-56A]. For the curves used by X25519 and X448, consult the security considerations of [RFC7748].

The ECDH-1PU algorithm is vulnerable to Key Compromise Impersonation (KCI) attacks. If the long-term static private key of a party is compromised, then the attacker can not only impersonate that party to other parties, but also impersonate any other party when communicating with the compromised party. The second and any subsequent messages in the two-way interactive handshake described in Section 3 are not vulnerable to KCI. If resistance to KCI is desired

in a single message, then it is RECOMMENDED to use a nested JWS signature over the content.

When Key Agreement with Key Wrapping is used, with the same Content Encryption Key (CEK) reused for multiple recipients, any of those recipients can produce a new message that appears to come from the original sender and will be trusted by any of the other recipients. It is RECOMMENDED that a unique CEK is used for each recipient.

6. References

6.1. Normative References

[NIST.800-56A] Barker, E., Chen, L., Roginsky, A., Vassilev, A., and R. Davis, "Recommendation for Pair-Wise Key Establishment Using Discrete Logarithm Cryptography Revision 3.", NIST Special Publication 800-56A, April 2018.

[RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <https://www.rfc-editor.org/info/rfc7515>.

[RFC7516] Jones, M. and J. Hildebrand, "JSON Web Encryption (JWE)", RFC 7516, DOI 10.17487/RFC7516, May 2015, <https://www.rfc-editor.org/info/rfc7516>.

[RFC7518] Jones, M., "JSON Web Algorithms (JWA)", RFC 7518, DOI 10.17487/RFC7518, May 2015, <https://www.rfc-editor.org/info/rfc7518>.

[RFC7748] Langley, A., Hamburg, M., and S. Turner, "Elliptic Curves for Security", RFC 7748, DOI 10.17487/RFC7748, January 2016, <https://www.rfc-editor.org/info/rfc7748>.

[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>.

6.2. Informative References

[PKAE] An, J., "Authenticated Encryption in the Public-Key Setting: Security Notions and Analyses", IACR ePrint 2001/079, 2001.

Author's Address

Neil Madden
ForgeRock
Broad Quay House
Prince Street
Bristol BS1 4DJ
United Kingdom

Email: neil.madden@forgerock.com