Workgroup: SUIT
Internet-Draft: draft-ietf-suit-mti-05
Published: 12 February 2024
Intended Status: Standards Track
Expires: 15 August 2024
Authors: B. Moran
Arm Limited
Ø. Rønningstad
Nordic Semiconductor
A. Tsukamoto
ALAXALA Networks Corp.

Mandatory-to-Implement Algorithms for Authors and Recipients of Software Update for the Internet of Things manifests

Abstract

This document specifies algorithm profiles for SUIT manifest parsers and authors to ensure better interoperability. These profiles apply specifically to a constrained node software update use case.

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 15 August 2024.

Table of Contents

1. Introduction

Mandatory algorithms may change over time due to an evolving threat landscape. Algorithms are grouped into algorithm profiles to account for this. Profiles may be deprecated over time. SUIT will define five choices of MTI profile specifically for constrained node software update. These profiles are:

- One Symmetric MTI profile
- Two "Current" Constrained Asymmetric MTI profiles
- Two "Current" AEAD Asymmetric MTI profiles
- One "Future" Constrained Asymmetric MTI profile

At least one MTI algorithm in each category MUST be FIPS qualified.

Because SUIT presents an asymmetric communication profile, with powerful/complex manifest authors and constrained manifest recipients, the requirements for Recipients and Authors are different.

Recipients MAY choose which MTI profile they wish to implement. It is RECOMMENDED that they implement the "Future" Asymmetric MTI profile. Recipients MAY implement any number of other profiles.

Authors MUST implement all MTI profiles. Authors MAY implement any number of other profiles.

AEAD is preferred over un-authenticated encryption. Where possible an AEAD profile SHOULD be selected. Certain constrained IoT applications require streaming decryption, which necessitates a non-AEAD ecryption algorithm. If the application is not a constrained device, the two AEAD profiles are RECOMMENDED.

Other use-cases of SUIT MAY define their own MTI algorithms.

2. Algorithms

The algorithms that form a part of the profiles defined in this document are grouped into:

- *Digest Algorithms
- *Authentication Algorithms
- *Key Exchange Algorithms
- *Encryption Algorithms

3. Profiles

Recognized profiles are defined below.

3.1. Symmetric MTI profile: suit-sha256-hmac-a128kw-a128ctr

Algorithm Type	Algorithm	COSE Key
Digest	SHA-256	-16
Authentication	HMAC-256	5
Key Exchange	A128KW Key Wrap	-3
Encryption	A128CTR	-65534

Table 1

3.2. Current Constrained Asymmetric MTI Profile 1: suit-sha256-es256-ecdh-a128ctr

Algorithm Type	Algorithm	COSE Key
Digest	SHA-256	-16
Authentication	ES256	-7
Key Exchange	ECDH-ES + A128KW	-29
Encryption	A128CTR	-65534

Table 2

3.3. Current Constrained Asymmetric MTI Profile 2: suit-sha256-eddsa-ecdh-a128ctr

Algorithm Type	Algorithm	COSE Key
Digest	SHA-256	-16
Authentication	EDDSA	-8
Key Exchange	ECDH-ES + A128KW	-29
Encryption	A128CTR	-65534

Table 3

3.4. Current AEAD Asymmetric MTI Profile 1: suit-sha256-es256-ecdh-a128gcm

Algorithm Type	Algorithm	COSE Key
Digest	SHA-256	-16
Authentication	ES256	-7
Key Exchange	ECDH-ES + A128KW	-29
Encryption	A128GCM	1

Table 4

3.5. Current AEAD Asymmetric MTI Profile 2: suit-sha256-eddsa-ecdh-chacha-poly

Algorithm Type	Algorithm	COSE Key
Digest	SHA-256	-16
Authentication	EDDSA	-8
Key Exchange	ECDH-ES + A128KW	-29
Encryption	ChaCha20/Poly1305	24

Table 5

3.6. Future Constrained Asymmetric MTI Profile 1: suit-sha256-hsslms-a256kw-a256ctr

Algorithm Type	Algorithm	COSE Key
Digest	SHA-256	-16
Authentication	HSS-LMS	-46
Key Exchange	A256KW	-5
Encryption	A256CTR	-65532

Table 6

4. Reporting Profiles

When using reverse-direction communication, particularly data structures that are designed for reporting of update capabilities, status, progress, or success, the same profile as the is used on the SUIT manifest SHOULD be used. There are cases where this is not

possible, such as suit-sha256-hsslms-a256kw-a256ctr. In this case, the closest equivalent profile SHOULD be used, for example suit-sha256-es256-ecdh-a128ctr.

5. Security Considerations

For the avoidance of doubt, there are scenarios where payload or manifest encryption are not required. In these scenarios, the encryption element of the selected profile is simply not used.

AES-CTR mode is specified, see [RFC9459]. All of the AES-CTR security considerations in [RFC9459] apply. A non-AEAD encryption mode is specified in this draft due to the following mitigating circumstances:

* Streaming decryption must be supported. Therefore, there is no difference between AEAD and plaintext hash verification.

* Out-of-order decryption must be supported. Therefore, we must use a stream cipher that supports random access.

* There are no chosen plaintext attacks: the plaintext is authenticated prior to encryption.

* Content Encryption Keys MUST be used to encrypt only once. See [I-D.ietf-suit-firmware-encryption].

As a result of these mitigating circumstances, AES-CTR is the most appropriate cipher for typical software/firmware delivery scenarios.

6. IANA Considerations

IANA is requested to create a page for COSE Algorithm Profiles within the category for Software Update for the Internet of Things (SUIT)

IANA is also requested to create a registry for COSE Alforithm Profiles within this page. The initial content of the registry is:

Profile	Status	Digest	Auth	Key Exchange	Encryption	Descriptor Array	Reference
suit-sha256-hmac-a128kw-a128ctr	MANDATORY	-16	5	-3	-65534	[-16, 5, -3, -65534]	Section 3.1
suit-sha256-es256-	MANDATORY	-16	-7	-29	-65534	[-16, -7, -29, -65534]	Section 3.2

Profile	Status	Digest	Auth	Key Exchange	Encryption	Descriptor Array	Reference
ecdh-a128ctr							
suit-sha256-eddsa-ecdh-a128ctr	MANDATORY	-16	-8	-29	-65534	[-16, -8, -29, -65534]	Section 3.3
suit-sha256-es256-ecdh-a128gcm	MANDATORY	-16	-7	-29	1	[-16, -7, -29, 1]	Section 3.4
suit-sha256-eddsa-ecdh-chacha-poly	MANDATORY	-16	-8	-29	24	[-16, -8, -29, 24]	Section 3.5
suit-sha256-hsslms-a256kw-a256ctr	MANDATORY	-16	-46	-5	-65532	[-16, -46, -5, -65532]	Section 3.6

Table 7

New entries to this registry require standards action.

7. References

7.1. Normative References

[I-D.ietf-suit-manifest] Moran, B., Tschofenig, H., Birkholz, H., Zandberg, K., and O. Rønningstad, "A Concise Binary Object Representation (CBOR)-based Serialization Format for the Software Updates for Internet of Things (SUIT) Manifest", Work in Progress, Internet-Draft, draft-ietf-suit-manifest-25, 5 February 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-suit-manifest-25>.

[RFC8152] Schaad, J., "CBOR Object Signing and Encryption (COSE)", RFC 8152, DOI 10.17487/RFC8152, July 2017, <https://www.rfc-editor.org/rfc/rfc8152>.

[RFC8778] Housley, R., "Use of the HSS/LMS Hash-Based Signature Algorithm with CBOR Object Signing and Encryption

(COSE)", RFC 8778, DOI 10.17487/RFC8778, April 2020, <https://www.rfc-editor.org/rfc/rfc8778>.

[RFC9052] Schaad, J., "CBOR Object Signing and Encryption (COSE): Structures and Process", STD 96, RFC 9052, DOI 10.17487/RFC9052, August 2022, <https://www.rfc-editor.org/rfc/rfc9052>.

[RFC9459] Housley, R. and H. Tschofenig, "CBOR Object Signing and Encryption (COSE): AES-CTR and AES-CBC", RFC 9459, DOI 10.17487/RFC9459, September 2023, <https://www.rfc-editor.org/rfc/rfc9459>.

7.2. Informative References

[I-D.ietf-suit-firmware-encryption]
Tschofenig, H., Housley, R., Moran, B., Brown, D., and K. Takayama, "Encrypted Payloads in SUIT Manifests", Work in Progress, Internet-Draft, draft-ietf-suit-firmware-encryption-18, 23 October 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-suit-firmware-encryption-18>.

[IANA-COSE] "CBOR Object Signing and Encryption (COSE)", 2022, <https://www.iana.org/assignments/cose/cose.xhtml>.

Appendix A. A. Full CDDL

The following CDDL creates a subset of COSE for use with SUIT. Both tagged and untagged messages are defined. SUIT only uses tagged COSE messages, but untagged messages are also defined for use in protocols that share a ciphersuite with SUIT.

To be valid, the following CDDL MUST have the COSE CDDL appended to it. The COSE CDDL can be obtained by following the directions in [RFC9052], Section 1.4.

```
SUIT_COSE_tool_tweak /= suit-sha256-hmac-a128kw-a128ctr
SUIT_COSE_tool_tweak /= suit-sha256-es256-ecdh-a128ctr
SUIT_COSE_tool_tweak /= suit-sha256-eddsa-ecdh-a128ctr
SUIT_COSE_tool_tweak /= suit-sha256-eddsa-ecdh-chacha-poly
SUIT_COSE_tool_tweak /= suit-sha256-hsslms-a256kw-a256ctr
SUIT_COSE_tool_tweak /= SUIT_COSE_Profiles

SUIT_COSE_Profiles /= SUIT_COSE_Profile_HMAC_A128KW_A128CTR
SUIT_COSE_Profiles /= SUIT_COSE_Profile_ES256_ECDH_A128CTR
SUIT_COSE_Profiles /= SUIT_COSE_Profile_EDDSA_ECDH_A128CTR
SUIT_COSE_Profiles /= SUIT_COSE_Profile_ES256_ECDH_A128GCM
SUIT_COSE_Profiles /= SUIT_COSE_Profile_EDDSA_ECDH_CHACHA20_POLY1304
SUIT_COSE_Profiles /= SUIT_COSE_Profile_HSSLMS_A256KW_A256CTR

suit-sha256-hmac-a128kw-a128ctr = [-16, 5, -3, -65534]
suit-sha256-es256-ecdh-a128ctr = [-16, -7, -25, -65534]
suit-sha256-eddsa-ecdh-a128ctr = [-16, -8, -25, -65534]
suit-sha256-es256-ecdh-a128gcm = [-16, -7, -25, 1]
suit-sha256-eddsa-ecdh-chacha-poly = [-16, -8, -25, 24]
suit-sha256-hsslms-a256kw-a256ctr = [-16, -46, -5, -65532]

SUIT_COSE_Profile_HMAC_A128KW_A128CTR = SUIT_COSE_Profile<5, -65534> .an
SUIT_COSE_Profile_ES256_ECDH_A128CTR = SUIT_COSE_Profile<-7,-65534> .and
SUIT_COSE_Profile_EDDSA_ECDH_A128CTR = SUIT_COSE_Profile<-8,-65534> .and
SUIT_COSE_Profile_ES256_ECDH_A128GCM = SUIT_COSE_Profile<-7,1> .and COSE
SUIT_COSE_Profile_EDDSA_ECDH_CHACHA20_POLY1304 = SUIT_COSE_Profile<-8,24
SUIT_COSE_Profile_HSSLMS_A256KW_A256CTR = SUIT_COSE_Profile<-46,-65532>

SUIT_COSE_Profile<authid, encid> = SUIT_COSE_Messages<authid,encid>

SUIT_COSE_Messages<authid, encid> = SUIT_COSE_Untagged_Message<authid, e
    SUIT_COSE_Tagged_Message<authid, encid>

SUIT_COSE_Untagged_Message<authid, encid> = SUIT_COSE_Sign<authid> /
    SUIT_COSE_Sign1<authid> / SUIT_COSE_Encrypt<encid> /
    SUIT_COSE_Encrypt0<encid> / SUIT_COSE_Mac<authid> /
    SUIT_COSE_Mac0<authid>

SUIT_COSE_Tagged_Message<authid, encid> = SUIT_COSE_Sign_Tagged<authid>
    SUIT_COSE_Sign1_Tagged<authid> / SUIT_COSE_Encrypt_Tagged<encid> /
    SUIT_COSE_Encrypt0_Tagged<encid> / SUIT_COSE_Mac_Tagged<authid> /
    SUIT_COSE_Mac0_Tagged<authid>

; Note: This is not the same definition as is used in COSE.
; It restricts a COSE header definition further without
; repeating the COSE definition. It should be merged
; with COSE by using the CDDL .and operator.
SUIT_COSE_Profile_Headers<algid> = (
    protected : bstr .cbor SUIT_COSE_alg_map<algid>,
    unprotected : SUIT_COSE_header_map
```

```
)
SUIT_COSE_alg_map<algid> = {
    1 => algid,
    * int => any
}

SUIT_COSE_header_map = {
    * int => any
}

SUIT_COSE_Sign_Tagged<authid> = #6.98(SUIT_COSE_Sign<authid>)


SUIT_COSE_Sign<authid> = [
    SUIT_COSE_Profile_Headers<authid>,
    payload : bstr / nil,
    signatures : [+ SUIT_COSE_Signature<authid>]
]


SUIT_COSE_Signature<authid> =  [
    SUIT_COSE_Profile_Headers<authid>,
    signature : bstr
]


SUIT_COSE_Sign1_Tagged<authid> = #6.18(SUIT_COSE_Sign1<authid>)


SUIT_COSE_Sign1<authid> = [
    SUIT_COSE_Profile_Headers<authid>,
    payload : bstr / nil,
    signature : bstr
]


SUIT_COSE_Encrypt_Tagged<encid> = #6.96(SUIT_COSE_Encrypt<encid>)


SUIT_COSE_Encrypt<encid> = [
    SUIT_COSE_Profile_Headers<encid>,
    ciphertext : bstr / nil,
    recipients : [+SUIT_COSE_recipient<encid>]
]


SUIT_COSE_recipient<encid> = [
    SUIT_COSE_Profile_Headers<encid>,
    ciphertext : bstr / nil,
    ? recipients : [+SUIT_COSE_recipient<encid>]
```

```
]


SUIT_COSE_Encrypt0_Tagged<encid> = #6.16(SUIT_COSE_Encrypt0<encid>)


SUIT_COSE_Encrypt0<encid> = [
    SUIT_COSE_Profile_Headers<encid>,
    ciphertext : bstr / nil,
]


SUIT_COSE_Mac_Tagged<authid> = #6.97(SUIT_COSE_Mac<authid>)


SUIT_COSE_Mac<authid> = [
   SUIT_COSE_Profile_Headers<authid>,
   payload : bstr / nil,
   tag : bstr,
   recipients :[+SUIT_COSE_recipient<authid>]
]


SUIT_COSE_Mac0_Tagged<authid> = #6.17(SUIT_COSE_Mac0<authid>)


SUIT_COSE_Mac0<authid> = [
   SUIT_COSE_Profile_Headers<authid>,
   payload : bstr / nil,
   tag : bstr,
]
```

Authors' Addresses

Brendan Moran
Arm Limited

Email: brendan.moran.ietf@gmail.com

Øyvind Rønningstad
Nordic Semiconductor

Email: oyvind.ronningstad@gmail.com

Akira Tsukamoto
ALAXALA Networks Corp.

Email: akira.tsukamoto@alaxala.com