Delay-Tolerant Networking
Internet-Draft
Intended status: Standards Track
Expires: August 7, 2020

E. Birrane
JHU/APL

February 4, 2020

BPSec Interoperability Security Contexts
draft-ietf-dtn-bpsec-interop-sc-01

Abstract

This document defines an integrity security context and a confidentiality security context suitable for testing the interoperability of Bundle Protocol Security (BPSec) implementations.

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 August 7, 2020.

Table of Contents

1. Introduction

The Bundle Protocol Security (BPSec) [I-D.ietf-dtn-bpsec] specification provides inter-bundle integrity and confidentiality features for networks deploying the Bundle Protocol (BP) [I-D.ietf-dtn-bpbis]. BPSec defines a set of BP extension blocks to carry security information produced under the auspices of some security context, but does not define a mandatory set of security contexts.

This document defines two security contexts (one for integrity services and one for confidentiality services) for populating BPSec Block Integrity Blocks (BIBs) and Block Confidentiality Blocks (BCBs).

The intent of these security context definitions is to provide a mechanism for interoperability testing. There is no claim that these contexts are suitable for operational deployment in any particular networking scenario. Further, there is no requirement that these contexts be used in any operational network deployments.

These contexts generate information that MUST be encoded using the CBOR specification documented in [RFC7049].

2. Requirements Language

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 [RFC2119].

3. Security Context BIB-IOP-HMAC256-SHA256

3.1. Overview

The BIB-IOP-HMAC256-SHA256 security context provides a keyed hash over a security target. This security context uses the SHA-256 secure hash algorithm discussed in [RFC4634] combined with the HMAC keyed hash discussed in [RFC2104]. This combination is based on the HMAC 256/256 algorithm defined in [RFC8152] Table 7: HMAC Algorithm Values.

The HMAC shall use a truncation length of 256 bits.

The BIB-IOP-HMAC256-SHA256 security context shall have the Security Context ID specified in Section 5.1.

3.2. Key Considerations

HMAC keys used with this context MUST be symmetric and 256 bits in length.

It is assumed that the node performing the integrity verification knows the HMAC key used to create the original keyed hash.

BIB-IOP-HMAC256-SHA256 provides no explicit requirements on the configuration, storage, or exchange of HMAC keys.

3.3. Canonicalization Algorithms

BIB-IOP-HMAC256-SHA256 uses the canonicalization algorithms defined in [I-D.ietf-dtn-bpsec] with the following exceptions.

The keyed hash MUST be calculated over the single, definite-length CBOR byte string representing the security target's block-type-specific-data field. All other fields of the security target (such as the block type code, block number, block processing control flags, or any CRC information) MUST NOT be included in the calculation.

3.4. Processing

3.4.1. Keyed Hash Generation

During keyed hash generation, the plain text of the security target and a HMAC key (given by local security policy) are input to the HMAC/SHA algorithm.

Upon successful hash generation, the output of the HMAC MUST be added as the security result for the security target.

Problems encountered in the keyed hash generation MUST be processed in accordance with local security policy.

3.4.2. Keyed Hash Verification

During keyed hash verification, the plain text of the security target and a HMAC key (given by local security policy) are input to the HMAC/SHA algorithm. The resulting HMAC output MUST be compared to the expected HMAC output encoded in the security results for the security target.

If the calculated HMAC and expected HMAC are identical, the verification MUST be considered a success. Otherwise, the verification MUST be considered a failure and processed according to local security policy.

3.5. Security Context Parameter Definitions

BIB-IOP-HMAC256-SHA256 defines no security context parameters.

3.6. Security Context Result Definitions

BIB-IOP-HMAC256-SHA256 defines the following security results.

BIB-IOP-HMAC256-SHA256 Security Results

Result Id	Result Name	CBOR Encoding Type	Description
1	Expected HMAC	byte string	The output of the HMAC calculation at the security source.

Table 1

4. Security Context BCB-IOP-AES-GCM-256

4.1. Overview

The BCB-IOP-AES-GCM-256 security context generates cipher text to replace the plain text in the block-type-specific-data field of its security target. BCB-IOP-AES-GCM-256 uses the Advanced Encryption Standard (AES) cipher operating in Galois/Counter Mode (GCM) [AES-GCM]. This formulation is based on the A256GCM algorithm defined in [RFC8152] Table 9: Algorithm Value for AES-GCM.

The BCB-IOP-AES-GCM-256 security context shall have the Security Context ID specified in Section 5.1.

4.2. Key Considerations

Keys used with this specification MUST be symmetric and 256 bits in length.

It is assumed that the node performing the decryption knows the symmetric key used for encryption.

BCB-IOP-AES-GCM-256 provides no explicit requirements on the configuration, storage, or exchange of keys.

4.3. Canonicalization Algorithms

BCB-IOP-AES-GCM-256 uses the canonicalization algorithms defined in [I-D.ietf-dtn-bpsec] with the following exceptions.

The plain text used during encryption MUST be calculated as the single, definite-length CBOR byte string representing the block-type-specific-data field excluding the CBOR byte string identifying byte and optional CBOR byte string length field.

For example, consider the following two CBOR byte strings and the plain text that would be extracted from them.

CBOR byte string Examples

CBOR Byte String	CBOR Encoding	Plain Text
0x18ED	0x18	0xED
0xC24CDEADBEEFDEADBEEFDEADBEEF	0xC24C	0xDEADBEEFDEADBEEFDEADBEEF

Table 2

Similarly, the cipher text used during decryption MUST be calculated

as the single, definite-length CBOR byte string representing the block-type-specific-data field excluding the CBOR byte string identifying byte and optional CBOR byte string length field.

All other fields of the security target (such as the block type code, block number, block processing control flags, or any CRC information) MUST NOT be considered as part of encryption or decryption.

4.4. Processing

4.4.1. Encryption

During encryption, the plain text of the security target MUST be input to the AES/GCM cipher with a unique Initialization Vector (IV) and an appropriate key (given by local security policy).

Upon successful encryption, the cipher text produced by AES/GCM (which will have the same length as the plain text provided to it) MUST replace the bytes used to define the plain text in the target block's block-type-specific-data field.

The IV input to the cipher MUST be added as a security parameter for the security target. Because replaying an IV in counter mode voids the confidentiality of all messages encrypted with said IV, this context also requires a unique IV for every encryption performed with the same key. This means the same key and IV combination MUST NOT be used more than once.

The authentication tag calculated by the AES/GCM cipher MUST be added as a security result for the security target.

Problems encountered in the encryption MUST be processed in accordance with local security policy.

NOTE: Because the cipher text and the plain text have the same length, the encoding information for the CBOR byte string (the CBOR byte string identifying byte and optional CBOR byte string length field) MUST remain unchanged in the target block's block-type-specific-data field. This allows for the replacement of plain text with cipher text without any additional consideration of block-type-specific-data field processing.

4.4.2. Decryption

During decryption, the cipher text of the security target MUST be input to the AES/GCM cipher with the IV present in the security parameters, an appropriate key (given by local security policy), and the authentication tag present in the security results.

The plain text produced by AES/GCM (which will have the same length as the cipher text provided to it) MUST replace the bytes used to define the cipher text in the target block's block-type-specific-data field.

If the cipher text fails to authenticate, if any needed parameters are missing, or if there are other problems in the decryption then the decryption MUST be treated as failed and processed in accordance with local security policy.

Upon successful decryption, the recovered plain text MUST replace the bytes used to define the cipher text in the target block's block-type-specific-data field.

NOTE: Because the cipher text and the plain text have the same length, the encoding information for the CBOR byte string (the CBOR byte string identifying byte and optional CBOR byte string length field) MUST remain unchanged in the target block's block-type-specific-data field. This allows for the replacement of cipher text with plain text without any additional consideration of block-type-specific-data field processing.

4.5. Security Context Parameter Definitions

BCB-IOP-AES-GCM-256 defines the following security context parameters.

Parm Id	Parm Name	CBOR Encoding Type	Description
1	Initialization Vector	byte string	The initialization vector. A value with a length, prior to CBOR encoding, between 8-16 bytes. 12 bytes is recommended.

Table 3

4.6. Security Result Definitions

BCB-IOP-AES-GCM-256 defines the following security results.

NOTE: Cipher text is not a security result as it is stored in the target block. When operating in GCM mode, AES produces cipher text of the same size as its plain text and, therefore, no additional logic is required to handle padding or overflow.

BCB-IOP-AES-GCM-256 Security Results

Result Id	Result Name	CBOR Encoding Type	Description
1	Authentication Tag	byte string	Output from the AES-GCM cipher. This value, prior to CBOR byte string encoding, MUST have a length of 16 bytes.

Table 4

5. IANA Considerations

5.1. Security Context Identifiers

This specification allocates two security context identifiers from the "BPSec Security Context Identifier" registry defined in [I-D.ietf-dtn-bpsec].

Additional Entries for the BPSec Security Context Identifiers Registry:

Value	Description	Reference
TBA	BIB-IOP-HMAC256-SHA256	This document
TBA	BCB-IOP-AES-GCM-256	This document

Table 5

6. Normative References

[AES-GCM] Dworkin, M., "NIST Special Publication 800-38D: Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC.", November 2007.

[I-D.ietf-dtn-bpbis] Burleigh, S., Fall, K., and E. Birrane, "Bundle Protocol Version 7", draft-ietf-dtn-bpbis-21 (work in progress), January 2020.

[I-D.ietf-dtn-bpsec] Birrane, E. and K. McKeever, "Bundle Protocol Security Specification", draft-ietf-dtn-bpsec-18 (work in progress), January 2020.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <https://www.rfc-editor.org/info/rfc2104>.

[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>.

[RFC4634] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and HMAC-SHA)", RFC 4634, DOI 10.17487/RFC4634, July 2006, <https://www.rfc-editor.org/info/rfc4634>.

[RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", RFC 7049, DOI 10.17487/RFC7049, October 2013, <https://www.rfc-editor.org/info/rfc7049>.

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

Appendix A. Acknowledgements

The following participants contributed useful review and analysis of these security contexts: Amy Alford and Sarah Heiner of the Johns Hopkins University Applied Physics Laboratory.

Author's Address

Edward J. Birrane, III
The Johns Hopkins University Applied
Physics Laboratory
11100 Johns Hopkins Rd.
Laurel, MD 20723
US

Phone: +1 443 778 7423
Email: Edward.Birrane@jhuapl.edu