TLS
Internet-Draft
Intended status: Informational
Expires: December 19, 2021

N. Cam-Winget
Cisco Systems
J. Visoky
ODVA

June 17, 2021

TLS 1.3 Authentication and Integrity only Cipher Suites
draft-camwinget-tls-ts13-macciphersuites-12

Abstract

This document defines the use of HMAC-only cipher suites for TLS 1.3, which provides server and optionally mutual authentication and data authenticity, but not data confidentiality. Cipher suites with these properties are not of general applicability, but there are use cases, specifically in Internet of Things (IoT) and constrained environments, that do not require confidentiality of exchanged messages while still requiring integrity protection, server authentication, and optional client authentication. This document gives examples of such use cases, with the caveat that prior to using these integrity-only cipher suites, a threat model for the situation at hand is needed, and a threat analysis must be performed within that model to determine whether the use of integrity-only cipher suites is appropriate. The approach described in this document is not endorsed by the IETF and does not have IETF consensus, but is presented here to enable interoperable implementation of a reduced security mechanism that provides authentication and message integrity without supporting confidentiality.

Table of Contents

1. Introduction 2
2. Terminology 3
3. Applicability Statement 3
4. Cryptographic Negotiation Using Integrity only Cipher Suites 6
5. Record Payload Protection for Integrity only Cipher Suites . 6
6. Key Schedule when using Integrity only Cipher Suites 8
7. Error Alerts 8
8. IANA Considerations 8
9. Security and Privacy Considerations 9
10. Acknowledgements 10
11. References 10
11.1. Normative References 10
11.2. Informative Reference 11
Authors' Addresses 11

1. Introduction

There are several use cases in which communications privacy is not strictly needed, although authenticity of the communications transport is still very important. For example, within the Industrial Automation space there could be TCP or UDP communications which command a robotic arm to move a certain distance at a certain speed. Without authenticity guarantees, an attacker could modify the packets to change the movement of the robotic arm, potentially causing physical damage. However, the motion control commands are not always considered to be sensitive information and thus there is no requirement to provide confidentiality. Another Internet of Things (IoT) example with no strong requirement for confidentiality is the reporting of weather information; however, message authenticity is required to ensure integrity of the message.

There is no requirement to encrypt messages in environments where the information is not confidential; such as when there is no intellectual property associated with the processes, or where the threat model does not indicate any outsider attacks (such as in a closed system). Note however, this situation will not apply equally to all use cases (for example, a robotic arm might be used in one case for a process that does not involve any intellectual property, but in another case used in a different process that does contain intellectual property). Therefore, it is important that a user or system developer carefully examine both the sensitivity of the data and the system threat model to determine the need for encryption before deploying equipment and security protections.

Besides having a strong need for authenticity and no need for confidentiality, many of these systems also have a strong requirement for low latency. Furthermore, several classes of IoT device (industrial or otherwise) have limited processing capability. However, these IoT systems still gain great benefit from leveraging TLS 1.3 for secure communications. Given the reduced need for confidentiality, TLS 1.3 [RFC8446] cipher suites that maintain data integrity without confidentiality are described in this document. These cipher suites are not meant for general use as they do not meet the confidentiality and privacy goals of TLS. They should only be used in cases where confidentiality and privacy is not a concern and there are constraints on using cipher suites that provide the full set of security properties. The approach described in this document is not endorsed by the IETF and does not have IETF consensus, but is presented here to enable interoperable implementation of a reduced security mechanism that provides authentication and message integrity with supporting confidentiality.

2. Terminology

This document adopts the conventions for normative language to provide clarity of instructions to the implementer. The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD 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.

3. Applicability Statement

The two HMAC SHA [RFC6234] based cipher suites defined in this document are intended for a small limited set of applications where confidentiality requirements are relaxed and the need to minimize the number of cryptographic algorithms is prioritized. This section describes some of those applicable use cases.

Use cases in the industrial automation industry, while requiring data integrity, often do not require confidential communications. Mainly, information communicated to unmanned machines to execute repetitive tasks does not convey private information. For example, there could be a system with a robotic arm that paints the body of a car. This equipment is used within a car manufacturing plant, and is just one piece of equipment in a multi-step manufacturing process. The movements of this robotic arm are likely not a trade secret or sensitive intellectual property, although some portions of the manufacturing of the car might very well contain sensitive intellectual property. Even the mixture for the paint itself might be confidential, but the mixing is done by a completely different piece of equipment and therefore communication to/from that equipment can be encrypted without requiring the communication to/from the robotic arm to be encrypted. Modern manufacturing often has segmented equipment with different levels of risk on intellectual property, although nearly every communication interaction has strong data authenticity requirements.

Another use case which is closely related is that of fine-grained time updates. Motion systems often rely on time synchronization to ensure proper execution. Time updates are essentially public; there is no threat from an attacker knowing the time update information. This should make intuitive sense to those not familiar with these applications; rarely if ever does time information present a serious attack surface dealing with privacy. However, the authenticity is still quite important. The consequences of maliciously modified time data can vary from mere denial of service to actual physical damage, depending on the particular situation and attacker capability. As these time synchronization updates are very fine-grained, it is again important for latency to be very low.

A third use case deals with data related to alarms. Industrial control sensing equipment can be configured to send alarm information when it meets certain conditions, for example, temperature goes above or below a given threshold. Often times this data is used to detect certain out-of-tolerance conditions, allowing an operator or automated system to take corrective action. Once again, in many systems the reading of this data doesn't grant the attacker information that can be exploited, it is generally just information regarding the physical state of the system. At the same time, being able to modify this data would allow an attacker to either trigger alarms falsely or to cover up evidence of an attack that might allow for detection of their malicious activity. Furthermore, sensors are often low powered devices that might struggle to process encrypted and authenticated data. These sensors might be very cost sensitive such that there is not enough processing power for data encryption. Sending data that is just authenticated but not encrypted eases the

burden placed on these devices, yet still allows the data to be protected against any tampering threats. A user can always choose to pay more for a sensor with encryption capability, but for some, data authenticity will be sufficient.

A fourth use case considers the protection of commands in the railway industry. In railway control systems, no confidentiality requirements are applied for the command exchange between an interlocking controller and a railway equipment controller (for instance, a railway point controller of a tram track where the position of the controlled point is publicly available). However, protecting integrity and authenticity of those commands is vital, otherwise, an adversary could change the target position of the point by modifying the commands, which consequently could lead to the derailment of a passing train. Furthermore, requirements for providing blackbox recording of the safety related network traffic can only be fulfilled through using authenticity-only ciphers, to be able to provide the safety related commands to a third party, which is responsible for the analysis after an accident.

The fifth use case deals with data related to civil aviation airplanes and ground communication. Pilots can send and receive messages to/from ground control such as airplane route-of-flight update, weather information, controller and pilot communication, and airline back office communication. Similarly, the Aviation Traffic Control (ATC) use air to ground communication to receive the surveillance data that relies on (is dependent on) downlink reports from an airplane's avionics. This communication occurs automatically in accordance with contracts established between the ATC ground system and the airplane's avionics. Reports can be sent whenever specific events occur, or specific time intervals are reached. In many systems the reading of this data doesn't grant the attacker information that can be exploited, it is generally just information regarding the airplane states, controller pilot communication, meteorological information etc. At the same time, being able to modify this data would allow an attacker to either put aircraft in the wrong flight trajectory or to provide false information to ground control that might delay flights and damage properties or harm life. Sending data that is not encrypted but is authenticated, allows the data to be protected against any tampering threats. Data authenticity is sufficient for the air traffic, weather and surveillance information exchange between airplanes and the ground systems.

The above use cases describe the requirements where confidentiality is not needed and/or interferes with other requirements. Some of these use cases are based on devices that come with a small runtime memory footprint and reduced processing power therefore the need to

minimize the number of cryptographic algorithms used is a priority. Despite this, it is noted that memory, performance, and processing power implications of any given algorithm or set of algorithms is highly dependent on hardware and software architecture. Therefore, although these cipher suites may provide performance benefits, they will not necessarily provide these benefits in all cases on all platforms. Furthermore, in some use cases third party inspection of data is specifically needed, which is also supported through the lack of confidentiality mechanisms.

4. Cryptographic Negotiation Using Integrity only Cipher Suites

The cryptographic negotiation as specified in [RFC8446] Section 4.1.1 remains the same, with the inclusion of the following cipher suites:

TLS_SHA256_SHA256 {0xC0, 0xB4}

TLS_SHA384_SHA384 {0xC0, 0xB5}

As defined in [RFC8446], TLS 1.3 cipher suites denote the AEAD algorithm for record protection and the hash algorithm to use with the HKDF. These cipher suites are defined in a similar way, but using the HMAC authentication tag to model the AEAD interface, as only an HMAC is provided for record protection (without encryption). These cipher suites allow the use of SHA-256 or SHA-384 as the Hashed Message Authentication Code (HMAC) for data integrity protection as well as its use for HMAC-based Key Derivation Function (HKDF). The authentication mechanisms remain unchanged with the intent to only update the cipher suites to relax the need for confidentiality.

Given that these cipher suites do not support confidentiality, they MUST NOT be used with authentication and key exchange methods that rely on confidentiality.

5. Record Payload Protection for Integrity only Cipher Suites

The record payload protection as defined in [RFC8446] is retained in modified form when integrity only cipher suites are used. Note that due to the purposeful use of hash algorithms, instead of AEAD algorithms, the confidentiality protection of the record payload is not provided. This section describes the mapping of record payload structures when integrity only cipher suites are employed.

Given that there is no encryption to be done at the record layer, the operations "Protect" and "Unprotect" take the place of "AEAD-Encrypt" and "AEAD-Decrypt", respectively, as referenced in [RFC8446]

As integrity protection is provided over the full record, the encrypted_record in the TLSCiphertext along with the additional_data input to protected_data (termed AEADEncrypted data in [RFC8446]) as defined in Section 5.2 of [RFC8446] remain the same. The TLSCiphertext.length for the integrity cipher suites will be:

TLS_SHA256_SHA256: TLSCiphertext.length = TLSPlaintext.length + 1 (type field) + length_of_padding + 32 (HMAC) = TLSInnerPlaintext_length + 32 (HMAC)

TLS_SHA384_SHA384: TLSCiphertext.length = TLSPlaintext.length + 1 (type field) + length_of_padding + 48 (HMAC) = TLSInnerPlaintext_length + 48 (HMAC)

Note that TLSInnerPlaintext_length is not defined as an explicit field in [RFC8446]; this refers to the length of the encoded TLSInnterPlaintext structure

The resulting protected_record is the concatenation of the TLSInnerPlaintext with the resulting HMAC. Note this analogous to the "encrypted_record" of [RFC8446], although it is referred to as a "protected_record" because of the lack of confidentiality via encryption. With this mapping, the record validation order as defined in Section 5.2 of [RFC8446] remains the same. That is, encrypted_record field of TLSCiphertext is set to:

encrypted_record = TLS13-HMAC-Protected = TLSInnerPlaintext || HMAC(write_key, nonce || additional_data || TLSInnerPlaintext)

Here "nonce" refers to the per-record nonce described in section 5.3 of [RFC8446].

For DTLS 1.3, the DTLSCiphertext is set to:

encrypted_record = DTLS13-HMAC-Protected = DTLSInnerPlaintext || HMAC(write_key, nonce || additional_data || DTLSInnerPlaintext)

The DTLS "nonce" refers to the per-record nonce described in section 4.2.2 of [DTLS13].

The Protect and Unprotect operations provide the integrity protection using HMAC SHA-256 or HMAC SHA-384 as described in [RFC6234].

Due to the lack of encryption of the plaintext, record padding does not provide any obfuscation as to the plaintext size, although it can be optionally included.

6. Key Schedule when using Integrity only Cipher Suites

The key derivation process for Integrity only Cipher Suites remains the same as defined in [RFC8446]. The only difference is that the keys used to protect the tunnel apply to the negotiated HMAC SHA-256 or HMAC SHA-384 ciphers. Note that the traffic key material (client_write_key, client_write_iv, server_write_key and server_write_iv) MUST be calculated as per RFC 8446, section 7.3. The key lengths and IVs for these cipher suites are according to the hash output lengths. In other words, the following key lengths and IV lengths SHALL be:

Cipher Suite	Key Length	IV Length
TLS_SHA256_SHA256	32 Bytes	32 Bytes
TLS_SHA384_SHA384	48 Bytes	48 Bytes

7. Error Alerts

The error alerts as defined by [RFC8446] remains the same, in particular:

bad_record_mac: This alert can also occur for a record whose message authentication code can not be validated. Since these cipher suites do not involve record encryption this alert will only occur when the HMAC fails to verify.

decrypt_error: This alert as described in [RFC8446] Section 6.2 occurs when the signature or message authentication code can not be validated. Note that this error is only sent during the handshake, not for an error in validating record HMACs.

8. IANA Considerations

IANA has granted registration the following specifically for this document within the TLS Cipher Suites Registry:

TLS_SHA256_SHA256 {0xC0, 0xB4} cipher suite and TLS_SHA384_SHA384 {0xC0, 0xB5} cipher suite.

Note that both of these cipher suites are registered with the DTLS-OK column set to Y and the Recommended column set to N

No further IANA action is requested by this document.

9. Security and Privacy Considerations

In general, except for confidentiality and privacy, the security considerations detailed in [RFC8446] and in [RFC5246] apply to this document. Furthermore, as the cipher suites described in this document do not provide any confidentiality, it is important that they only be used in cases where there are no confidentiality or privacy requirements and concerns; and the runtime memory requirements can accommodate support for authenticity-only cryptographic constructs.

With the lack of data encryption specified in this specification, no confidentiality or privacy is provided for the data transported via the TLS session. That is, the record layer is not encrypted when using these cipher suite, and the handshake also is not encrypted. To highlight the loss of privacy, the information carried in the TLS handshake, which includes both the Server and Client certificates, while integrity protected, will be sent unencrypted. Similarly, other TLS extensions that may be carried in the Server's EncryptedExtensions message will only be integrity protected without provisions for confidentiality. Furthermore, with this lack of confidentiality, any private PSK data MUST NOT be sent in the handshake while using these cipher suites. However, as PSKs may be loaded externally, these cipher suites can be used with the 0-RTT handshake defined in [RFC8446], with the same security implications discussed there applied.

Application protocols which build on TLS or DTLS for protection (e.g. HTTP) may have implicit assumptions of data confidentiality. Any assumption of data confidentiality is invalidated by the use of these cipher suites, as no data confidentiality is provided. This applies to any data sent over the application-data channel (e.g. bearer tokens in HTTP), as data requiring confidentiality MUST NOT be sent using these cipher suites.

Limits on key usage for AEAD-based ciphers are described in [RFC8446]. However, as the cipher suites discussed here are not AEAD, those same limits do not apply. The general security properties of HMACs discussed in [RFC2104] and [BCK1] apply. Additionally, security considerations on the algorithm's strength based on the HMAC key length and trunction length further described in [RFC4868] also apply. Until further cryptanalysis demonstrate limitations on key usage for HMACs, general practice for key usage recommends that implementations place limits on the lifetime of the HMAC keys and invoke a key update as described in [RFC8446] prior to reaching this limit.

DTLS 1.3 defines a mechanism for encrypting the DTLS record sequence numbers. However, as these cipher suites do not utilize encryption, the record sequence numbers are sent unencrypted. That is, the procedure for DTLS record sequence number protection is to apply no protection for these cipher suites.

Given the lack of confidentiality, these cipher suites MUST NOT be enabled by default. As these cipher suites are meant to serve the IoT market, it is important that any IoT endpoint that uses them be explicitly configured with a policy of non-confidential communications.

10. Acknowledgements

The authors would like to acknowledge the work done by Industrial Communications Standards Groups (such as ODVA) as the motivation for this document. We would also like to thank Steffen Fries for providing a fourth use case and Madhu Niraula for a fifth use case. In addition, we are grateful for the advice and feedback from Joe Salowey, Blake Anderson, David McGrew, Clement Zeller, and Peter Wu.

11. References

11.1. Normative References

[BCK1] Bellare, M., Canetti, R., and H. Krawczyk, "Keyed Hash Functions and Message Authentication", <https://cseweb.ucsd.edu/~mihir/papers/kmd5.pdf>.

[DTLS13] IETF Internet Drafts editor, "https://tools.ietf.org/id/draft-ietf-tls-dtls13-38.txt".

[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>.

[RFC4868] Kelly, S. and S. Frankel, "Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec", RFC 4868, DOI 10.17487/RFC4868, May 2007, <https://www.rfc-editor.org/info/rfc4868>.

[RFC6234] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <https://www.rfc-editor.org/info/rfc6234>.

[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>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

11.2. Informative Reference

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

Authors' Addresses

Nancy Cam-Winget
Cisco Systems
3550 Cisco Way
San Jose, CA 95134
USA

Email: ncamwing@cisco.com

Jack Visoky
ODVA
1 Allen Bradley Dr
Mayfield Heights, OH 44124
USA

Email: jmvisoky@ra.rockwell.com