HTTP
Internet-Draft
Intended status: Informational
Expires: 23 April 2026

J. Hoyland

C. Patton
Cloudflare

20 October 2025

HTTP Signature Component for TLS Channel Binding
draft-hoypat-httpbis-message-signatures-ekm-00

Abstract

A derived component is specified for HTTP Message Signatures that binds the signature to the underlying secure channel (TLS over TCP or QUIC), thereby ensuring a signed message transmitted over one channel cannot be retransmitted over another. The component consists of key material exported from TLS.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://cjpatton.github.io/draft-hoypat-httpbis-message-signatures-ekm/draft-hoypat-httpbis-message-signatures-ekm.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-hoypat-httpbis-message-signatures-ekm/.

Discussion of this document takes place on the HTTP Working Group mailing list (mailto:ietf-http-wg@w3.org), which is archived at https://lists.w3.org/Archives/Public/ietf-http-wg/.

Source for this draft and an issue tracker can be found at https://github.com/cjpatton/draft-hoypat-httpbis-message-signatures-ekm.

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/.

Copyright Notice

Table of Contents

1. Introduction

HTTP Message Signatures [RFC9421] allow various components of an HTTP message to the authenticated by the sender either using a digital signature or a message authentication code (MAC). The exact set of components to be signed may very depending upon the application:

1. the components that need to be signed depend on the security considerations of the application; and

2. some components of the message may not available at the time of signing or verification.

To accommodate these limitations, HTTP Message Signatures defines a number of HTTP Message Components (Section 2 of [RFC9421]) and specifies rules for transforming components into the input to the

signature algorithm (Section 3.1 of [RFC9421]). The value of most components are extracted directly from the bytes of the HTTP message; others are derived from the message through a well-specified process.

All components are derived from the HTTP messages themselves. Consequentially, an on-path attacker with access to the HTTP messages transmitted between the client and server can replay a signed message at will. This is described in Section 7.2.2 of [RFC9421].

The nonce parameter provides some defense against replay attacks, but this mechanism is not applicable in all deployment scenarios. For example, it is common for two TLS servers to be authoritative for the same DNS name. (This setup is commonly referred to as "multi-CDN".) In this scenario, the first server can intercept a signed request from a client, then replay that request to the second server, thereby impersonating the client.

The goal of this document is to make replay protection more robust. A new derived component is defined for HTTP Message Signatures whose value is set to key material exported from TLS as defined in Section 7.5 of [RFC8446]. This binds the signed message to the underlying TLS channel, thereby ensuring the signature is never accepted outside of that channel.

> OPEN ISSUE Would it be better do define exported key material as a signature parameter instead of as a derived component?

2. Conventions and Definitions

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.

3. The @ekm Derived Component

A new derived component is defined with the name @ekm.

The contents of this component are the output of a call to the exporter interface of the underlying TLS connection as defined in Section 7.5 of [RFC8446], encoded in base64 [RFC4648]. The label parameter of the exporter function is set to "http-sig-ekm" and the context value is the version of TLS. For TLS 1.3 (i.e., [RFC8446]) its value is 0x0304.

TLS 1.3 or later is REQUIRED. This derived component is not compatible with HTTP messages sent in plaintext or over TLS 1.2 and below.

NOTE We could in principal specify this for TLS 1.2, if we need to.

If the signer and verifier do not agree on the value of @ekm, then the signature will not verify. If the signer and verifier share a TLS connection between them, then they will compute the same value. If they do not share a direct TLS connection, it is possible to architect a system such that the verifier does not directly call the exporter interface, but is simply provided its output on a trusted channel. This behaviour works, but requires the verifier and caller to trust each other.

OPEN ISSUE How do we negotiate usage of @ekm? Both the signer and verifier need to support the new derived component into generate message signatures that use it, but the signer might not know if the verifier uses it.

4. Security Considerations

TODO Define a channel binding and say why it prevents replays between CDNs.

5. IANA Considerations

TODO Update the "HTTP Signature Derived Component Names" registry.

6. 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/rfc/rfc2119>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/rfc/rfc4648>.

[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/rfc/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/rfc/rfc8446>.

[RFC9421] Backman, A., Ed., Richer, J., Ed., and M. Sporny, "HTTP Message Signatures", RFC 9421, DOI 10.17487/RFC9421, February 2024, <https://www.rfc-editor.org/rfc/rfc9421>.

Acknowledgments

TODO acknowledge.

Authors' Addresses

Jonathan Hoyland
Email: jonathan.hoyland@gmail.com

Christopher Patton
Cloudflare
Email: chrispatton+ietf@gmail.com