Workgroup: Privacy Pass
Internet-Draft:
draft-hendrickson-privacypass-expiration-
extension-01
Published: 7 August 2023
Intended Status: Standards Track
Expires: 8 February 2024
Authors: S. Hendrickson C. A. Wood
Google Cloudflare, Inc.

Privacy Pass Token Expiration Extension

Abstract

This document describes an extension for Privacy Pass that allows tokens to encode expiration information.

About This Document

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

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-hendrickson-privacypass-expiration-extension/.

Discussion of this document takes place on the Privacy Pass Working Group mailing list (mailto:privacy-pass@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/privacy-pass/. Subscribe at https://www.ietf.org/mailman/listinfo/privacy-pass/.

Source for this draft and an issue tracker can be found at https://github.com/chris-wood/draft-hendrickson-privacypass-expiration-extension.

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 8 February 2024.

Copyright Notice

Table of Contents

1. Introduction

Some Privacy Pass token types support binding additional information to the tokens, often referred to as public metadata. [AUTH-EXTENSIONS] describes an extension parameter to the basic PrivateToken HTTP authentication scheme [AUTH-SCHEME] for supplying this metadata alongside a token. [EXTENDED-ISSUANCE] describes variants of the basic Privacy Pass issuance protocols [BASIC-ISSUANCE] that support issuing tokens with public metadata. However, there are no existing extensions defined to make use of these protocol extensions.

This document describes an extension for Privacy Pass that allows tokens to encode expiration information. The use case and deployment considerations, especially with respect to the resulting privacy impact, are also discussed.

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. Expiration Extension

The expiration extension is an extension used to convey the expiration for an issued token. It is useful for Privacy Pass deployments that make use of cached tokens, i.e., those that are not bound to a specific TokenChallenge redemption context, without having to frequently rotate issuing public keys.

For example, consider a Privacy Pass deployment wherein Clients use cached tokens that are valid for one hour. Clients could pre-fetch these tokens each hour and the Issuer and Origin could rotate the verification key every hour to force expiration. Alternatively, Clients could pre-fetch tokens for the entire day all at once, including an expiration timestamp in each token to indicate the time window for which the token is valid.

The value of this extension is an ExpirationTimestamp, defined as follows.

```
struct {
   uint64 timestamp_precision;
   uint64 timestamp;
} ExpirationTimestmap;
```

The ExpirationTimestmap fields are defined as follows:

- "timestamp_precision" is an 8-octet integer, in network byte order, representing the granularity of the timestamp, i.e., the target to which the timestamp is rounded for loss of precision.

- "timestamp" is an 8-octet integer, in network byte order, representing the expiration timestamp. The expiration timestamp is the UNIX time in seconds at which a token expires.

As an example, an ExpirationTimestamp structure with the following value would be interpreted as an expiration timestamp of 1688583600, i.e., July 05, 2023 at 19:00:00 GMT+0000, which is the timestamp rounded to the nearest hour (timestamp_precision = 3600).

```
struct {
   uint64 timestamp_precision = 3600;
   uint64 timestamp = 1688583600;
} ExpirationTimestmap;
```

4. Privacy Considerations

This extension intentionally adds more information to a token that might not otherwise be visibile to Attester, Issuer, or Origin. As such, how this information is chosen can have an impact on Origin-Client, Issuer-Client, Attester-Origin, or redemption context unlinkability as defined in Section 3.2 of [ARCHITECTURE]. Mitigating risk of privacy violation requires that the extension be constructed in a way that does not induce anonymity set partitioning, as described in Section 6.1 of [ARCHITECTURE].

The best way to achieve this in practice is for Clients to use the same limited sets of information in the extension. Consistency can be achieved in a variety of ways. For example, Client implementations might insist that all Clients use the same deterministic function for computing the expiration timestamp, e.g., some function F(current time). This function would round the current timestamp, resulting in a loss of precision but overall less unique value. One way to implement this function would by rounding the timestamp to the nearest hour, day, or week. Of course, this does not account for clock skew, which occurs with some non-neglgiible probability in practice [CLOCK-SKEW].

An alternative implementation strategy for consistency is to run some sort of consistency check to ensure that the Client uses a value that is consistent with other Clients. Several consistency mechanisms exist; see [CONSISTENCY] for more information. Such an explicit consistency check would depend less upon the Client's current clock and thus be more robust at the cost of additional work.

Orthogonal to the mechanism used to ensure consistency, it is also important that Clients choose expiration timestamps that are shared by other Clients. Consider, for example, a scenario where two Clients consistently choose expiration timestamps per the recommendation above, but only one Client ever requests a token within a given expiration window. Despite the consistency check in place, the actual value of the timestamp is still unique to one of the Clients.

The means by which implementations ensure that some minimum number of Clients share the same expiration timestamp is a deployment-specific challenge. For example, in the Split Origin, Attester, and Issuer deployments as described in Section 4.4 of [ARCHITECTURE], the Attester is positioned to ensure that Clients do not choose consistent yet unique values. General purpose approaches to ensure that some minimum number of Clients share the same expiration timestamp are outside the scope of this document; indeed, this

problem is not unique to Privacy Pass and is common to other privacy-related protocols such as Oblivious HTTP [OHTTP].

5. Security Considerations

Use of the expiration extension risks revealing additional information to parties that see the extension, including the Attester, Issuer, and Origin. Section 4 discusses specific privacy implications for use of this extension that aim to mitigate exposure of information that can unintentionally partition the Client anonymity set and lead to Origin-Client, Issuer-Client, Attester-Origin, or redemption context unlinkability as defined in Section 3.2 of [ARCHITECTURE]. General information regarding the use of extensions and their possible impact on Client privacy can be found in Section 3.4.3 of [ARCHITECTURE] and Section 6.1 of [ARCHITECTURE].

6. IANA Considerations

This document registers the following entry into the "Privacy Pass PrivateToken Extensions" registry.

*Expiration extension

-Type: 0x0001

-Name: Expiration

-Value: ExpirationTimestamp value as defined in Section 3

-Reference: This document

-Notes: None

7. References

7.1. Normative References

[ARCHITECTURE] Davidson, A., Iyengar, J., and C. A. Wood, "The Privacy Pass Architecture", Work in Progress, Internet-Draft, draft-ietf-privacypass-architecture-13, 15 June 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-privacypass-architecture-13>.

[AUTH-EXTENSIONS] Hendrickson, S. and C. A. Wood, "The PrivateToken HTTP Authentication Scheme Extensions Parameter", Work in Progress, Internet-Draft, draft-wood-privacypass-auth-scheme-extensions-00, 10 July 2023, <https://datatracker.ietf.org/doc/html/draft-wood-privacypass-auth-scheme-extensions-00>.

[AUTH-SCHEME]
Pauly, T., Valdez, S., and C. A. Wood, "The Privacy Pass HTTP Authentication Scheme", Work in Progress, Internet-Draft, draft-ietf-privacypass-auth-scheme-11, 23 June 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-privacypass-auth-scheme-11>.

[BASIC-ISSUANCE] Celi, S., Davidson, A., Valdez, S., and C. A. Wood, "Privacy Pass Issuance Protocol", Work in Progress, Internet-Draft, draft-ietf-privacypass-protocol-11, 26 June 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-privacypass-protocol-11>.

[CONSISTENCY] Davidson, A., Finkel, M., Thomson, M., and C. A. Wood, "Key Consistency and Discovery", Work in Progress, Internet-Draft, draft-ietf-privacypass-key-consistency-01, 10 July 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-privacypass-key-consistency-01>.

[EXTENDED-ISSUANCE] "*** BROKEN REFERENCE ***".

[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>.

[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>.

7.2. Informative References

[CLOCK-SKEW]
Acer, M., Stark, E., Felt, A., Fahl, S., Bhargava, R., Dev, B., Braithwaite, M., Sleevi, R., and P. Tabriz, "Where the Wild Warnings Are: Root Causes of Chrome HTTPS Certificate Errors", ACM, Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, DOI 10.1145/3133956.3134007, October 2017, <https://doi.org/10.1145/3133956.3134007>.

[OHTTP] "*** BROKEN REFERENCE ***".

Acknowledgments

This document received input and feedback from Jim Laskey.

Authors' Addresses

Scott Hendrickson
Google

Email: scott@shendrickson.com

Christopher A. Wood
Cloudflare, Inc.

Email: caw@heapingbits.net