

Privacy Pass
Internet-Draft
Intended status: Standards Track
Expires: 14 November 2026

S. Hendrickson
Google
C. A. Wood
13 May 2026

The PrivateToken HTTP Authentication Scheme Extensions Parameter
draft-ietf-privacypass-auth-scheme-extensions-03

Abstract

This document specifies new parameters for the "PrivateToken" HTTP authentication scheme, called extensions. The purpose of these extensions is to negotiate and carry public metadata for Privacy Pass protocols.

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://ietf-wg-privacypass.github.io/draft-ietf-privacypass-auth-scheme-extensions/draft-ietf-privacypass-auth-scheme-extensions.html>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-ietf-privacypass-auth-scheme-extensions/>.

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/ietf-wg-privacypass/draft-ietf-privacypass-auth-scheme-extensions>.

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 14 November 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
2. Conventions and Definitions	3
3. PrivateToken Extensions Parameter	3
4. PrivateToken Challenge Extension Parameter	4
5. Extensions Negotiation	5
6. Security Considerations	6
7. IANA Considerations	6
8. References	7
8.1. Normative References	7
8.2. Informative References	7
Authors' Addresses	8

1. Introduction

The primary Token structure in the "PrivateToken" HTTP authentication scheme [AUTHSCHEME] is composed as follows:

```
struct {
    uint16_t token_type;
    uint8_t nonce[32];
    uint8_t challenge_digest[32];
    uint8_t token_key_id[Nid];
    uint8_t authenticator[Nk];
} Token;
```

Functionally, this structure conveys a single bit of information from the issuance protocol: whether or not the token is valid (as indicated by a valid authenticator value). This structure does not admit any additional information to flow from the issuance protocol, including, for example, public metadata that is incorporated into the issuance protocol.

This document specifies a new parameter for the "PrivateToken" HTTP authentication scheme for carrying extensions. This extensions parameter is cryptographically bound to the Token structure via the Privacy Pass issuance protocol.

This document additionally defines an optional extension negotiation scheme which allows origins to indicate what extension types they expect, and allows clients to respond with the extensions appropriate for their context.

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. PrivateToken Extensions Parameter

As defined in Section 2.2 of [AUTHSCHEME], the "PrivateToken" authentication scheme defines one parameter, "token", which contains the base64url-encoded Token struct. This document defines a new parameter, "extensions," which contains the base64url-encoded representation of the following Extensions structure. This document follows the default padding behavior described in Section 3.2 of [RFC4648], so the base64url value MUST include padding.

```
struct {
    ExtensionType extension_type;
    opaque extension_data<0..2^16-1>;
} Extension;

enum {
    reserved(0),
    (65535)
} ExtensionType;

struct {
    Extension extensions<0..2^16-1>;
} Extensions;
```

The contents of Extensions are a list of Extension values, each of which is a type-length-value structure whose semantics are determined by the type. The type and length of each extension are 2-octet integers, in network byte order. The length of the extensions list is also a 2-octet integer, in network byte order.

Clients, Issuers, and Origins all agree on the content and encoding of this Extensions structure, i.e., they agree on the same type-length-value list. The list MUST be ordered by ExtensionType value, from 0 to 65535. Extension types MAY be repeated. The value of the Extensions structure is used as-is when verifying the value of the corresponding "token" parameter in the "PrivateToken" authentication header. As an example, Clients presenting this extension parameter to origins would use an Authorization header field like the following:

```
Authorization: PrivateToken token="abc...", extensions="def..."
```

Since the new parameters (extensions, extension-set) are defined as containing base64url-encoded data, they will contain characters that require double-quotes. Therefore, parameter values for "token" and "extensions" MUST be enclosed in double-quotes.

Future documents may specify extensions to be included in this structure. Registration details for these extensions are in Section 7.

Each Privacy Pass issuance protocol, identified by a token type, specifies the structure of the PrivateToken value to be used. Extensions are bound to the resulting tokens via the issuance protocol. In particular, the value of an Extensions structure is provided as metadata for the issuance protocol. Issuance protocols are specified in [PUBLIC-ISSUANCE].

4. PrivateToken Challenge Extension Parameter

As defined in Section 2.1 of [AUTHSCHEME], the "PrivateToken" authentication scheme defines two parameters, "challenge" which contains the base64url-encoded TokenChallenge struct, and a "token-key" which contains the base64url-encoded public key used for this challenge. This document defines two OPTIONAL new parameters, "extension-set," which contains the base64url-encoded representation of the following ExtensionSet structure, and "extensions" which contain the base64url-encoded representation of the Extensions structure defined in Section 3. This document follows the default padding behavior described in Section 3.2 of [RFC4648], so the base64url value MUST include padding.

```
struct {
    enum { false(0), true(1) } Bool;
    Bool is_required;
    ExtensionType extension_type;
} ExtensionEntry;

enum {
    reserved(0),
    (65535)
} ExtensionType;

struct {
    ExtensionEntry extension_types<0..2^16-1>;
} ExtensionSet;
```

The contents of ExtensionSet is a list of ExtensionEntry structs containing extensions (defined in Section 3), each of which is a type-length-value structure whose semantics are determined by the type, and a bit marking whether the extension is required or optional. The type and length of each ExtensionType is a 2-octet integer, in network byte order. The length of the extension_types list is also a 2-octet integer, in network byte order.

ExtensionTypes are to be defined outside of this document.

The extensions parameter is to be used for pre-populated extension structs the origin suggests to the client.

When presented with an ExtensionSet, a client should expect to be rejected if not providing required extensions. A client MAY provide optional extensions. A client MAY use the pre-populated extension provided by the origin, or craft its own.

WWW-Authenticate:

PrivateToken challenge="abc...", token-key="123...", extension-set="%00%01%00%02..." extensions="def..."

5. Extensions Negotiation

In some Privacy Pass deployments, the set of extensions may be well known to Clients and Origins and thus do not require negotiation. In this case, no extension-set or extensions are provided by the origin in the PrivateToken. In other settings, negotiation may be required. However, negotiation can raise privacy risks, by partitioning Clients by their chosen provided extensions risking Origin-Client unlinkability. Some of these risks may be mitigated if all Clients in a given redemption context respond to negotiation in the same manner. However, if Clients have different observable behavior, e.g., if certain extension use is determined by user choice, Origins

can observe this differential behavior and therefore partition Clients in a redemption context.

6. Security Considerations

Privacy considerations for tokens that include additional information are discussed in [ARCHITECTURE]. Additional considerations for use of extensions, including those that arise when deciding which extensions to use, are described in Section 5. The TLS Protocol [RFC8446], Section 1.3 provides the transport layer security guarantees that protect the transmission of the PrivateToken and its extensions.

The primary security concern introduced by the extensions parameter is the risk of client tagging or tracking through the metadata itself. The size and specific content of the extension metadata can potentially add extra, observable bits of information that a service could use to tag clients. Mitigations include mechanisms to monitor that clients are getting evenly distributed bits, mechanisms to ensure clients fairly pick their own metadata, and mechanisms to ensure that clients understand the total traffic on the system. These are out of scope.

7. IANA Considerations

IANA is requested to create a new "Privacy Pass PrivateToken Extensions" registry in the "Privacy Pass Parameters" page to list possible extension values and their meaning. Each extension has a two-byte type, so the maximum possible value is 0xFFFF = 65535.

Template:

- * Type: The two-byte extension type
- * Name: Name of the extension
- * Value: Syntax and semantics of the extension
- * Reference: Where this extension and its value are defined
- * Notes: Any notes associated with the entry

New entries in this registry are subject to the Specification Required registration policy ([RFC8126], Section 4.6). Designated experts need to ensure that the extension is sufficiently clearly defined and, importantly, has a clear description of the privacy implications of using the extension, framed in the context of partitioning the client anonymity set as described in [ARCHITECTURE].

8. References

8.1. Normative References

[AUTHSCHEME]

Pauly, T., Valdez, S., and C. A. Wood, "The Privacy Pass HTTP Authentication Scheme", Work in Progress, Internet-Draft, draft-ietf-privacypass-auth-scheme-15, 23 October 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-privacypass-auth-scheme-15>>.

[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>>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/rfc/rfc8126>>.

[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>>.

8.2. Informative References

[ARCHITECTURE]

Celi, S., Davidson, A., Valdez, S., and C. A. Wood, "Privacy Pass Issuance Protocols", RFC 9578, DOI 10.17487/RFC9578, June 2024, <<https://www.rfc-editor.org/rfc/rfc9578>>.

[PUBLIC-ISSUANCE]

Hendrickson, S. and C. A. Wood, "Public Metadata Issuance", Work in Progress, Internet-Draft, draft-hendrickson-privacypass-public-metadata-03, 25 November 2023, <<https://datatracker.ietf.org/doc/html/draft-hendrickson-privacypass-public-metadata-03>>.

Authors' Addresses

Scott Hendrickson
Google
Email: scott@shendrickson.com

Christopher A. Wood
Email: caw@heapingbits.net