

Privacy Pass
Internet-Draft
Intended status: Informational
Expires: 20 August 2026

T. Meunier
Cloudflare Inc.
16 February 2026

Privacy Pass Reverse Flow
draft-meunier-privacypass-reverse-flow-03

Abstract

This document specifies an instantiation of Privacy Pass Architecture [RFC9576] that allows for a "reverse" flow from the Origin to the Client. It describes a method for an Origin to issue a state update to the Client in response to a request in which a token is redeemed.

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://thibmeu.github.io/draft-meunier-privacypass-reverse-flow-informational/draft-meunier-privacypass-reverse-flow.html>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-meunier-privacypass-reverse-flow/>.

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/thibmeu/draft-meunier-privacypass-reverse-flow-informational>.

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 20 August 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	3
2. Motivation	3
2.1. Refunding tokens	3
2.2. Bootstrapping Issuer	4
2.3. Attester feedback loop	4
2.4. Anonymous credential composition	4
3. Terminology	5
4. Architecture overview	5
5. CredentialRequest, CredentialResponse, and CredentialFinalization	6
6. Reverse flow with an HTTP header	7
6.1. Client behaviour	8
6.2. Origin/Issuer/Attester deployment	8
6.3. Split Origin-Attester deployment	9
7. Privacy Considerations	10
7.1. Issuer face values	11
7.2. Token for specific Clients	11
7.3. Swap endpoint and its privacy implication	12
8. IANA Considerations	12
9. References	12
9.1. Normative References	12
9.2. Informative References	13
Acknowledgments	13
Changelog	13
Author's Address	14

1. Introduction

This document specifies an instantiation of Privacy Pass Architecture [RFC9576] that allows for a reverse flow from the Origin to the Client.

In other words, it specifies a way for an Origin to act as an Attester + Issuer.

2. Motivation

With Privacy Pass issuance as described in [RFC9576], once a token is presented by a Client, it is considered spent and cannot be reused in order to guarantee unlinkability. If a token was to be presented twice, the two requests would be linkable by the Origin.

However, requiring that all tokens are spent only once means that Clients need to request more tokens to perform more requests. This is true even if the initial request didn't need a token presentation, for instance due to a cost being insignificant to the Origin.

This draft provides a mechanism for an Origin to provide a requesting Client with an updated state, allowing them to present new tokens on future requests. Origin act as a new Attester/Issuer entity.

Below, we present different use cases.

2.1. Refunding tokens

Certain Origins use Privacy Pass tokens to rate-limit requests they receive over a certain time window because of resource constraints. If a Client sends a request that can be served without utilising that resource, the Origin would like to authorise them to do a second request. This is the case for request requiring compute and the compute is low, or when the request leads to a redirection instead of content generation for instance.

With a reverse flow, a Client that has already been authorised by an Origin can maintain that authorization, without losing the unlinkability property provided by Privacy Pass.

2.2. Bootstrapping Issuer

An Origin wants to grant 30 access for Clients that solved a CAPTCHA. To do so, it consumes a type 0x0002 public verifiable token from an initial Issuer that guarantees a CAPTCHA has been solved, and use it to issue 30 type 0x0001 private tokens. Without a reverse flow, the Origin would have to require 30 0x0002 Issuer tokens, which have lower performance and a higher number of requests going to the Issuer.

2.3. Attester feedback loop

In [RFC9576], a Client gets a token from an Issuer and redeems it at an Origin. However, if the Client's request is deemed unwanted by the Origin at redemption time, there is no mechanism that prevents the Client from going back to the initial Issuer to get a new token and be authorized again.

With a reverse flow, the initial Issuer may require Clients to present an Origin-issued token before providing them with a second token. This allows for a feedback loop between the Origin and the initial Issuer, without breaking Client unlinkability.

2.4. Anonymous credential composition

Privacy Pass Architecture as defined by [RFC9576] centers around tokens, which issuance flows are defined in [RFC9578].

More recent explorations ([PRIVACYPASS-ARC], [PRIVACYPASS-BBS], [PRIVACYPASS-ACT]) are providing credentials to clients, which presentation result in a scoped token. These schemes are instantiation of a reverse flow, both because the Client holds a state it can use to perform multiple token presentation, as well as because the Origin can provides an updated state to requesting Client.

In additions, these schemes are more costly, and usage specific. With a reverse flow, the initial Issuer and the Origin issuer may use different credentials, which are suited to their use case. One use case is rate limiting and blocking. [PRIVACYPASS-ARC] provides rate-limit per origin with a unique credentials, while [PRIVACYPASS-ACT] allows to rate-limit a specific session once it's been established.

3. Terminology

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.

We reuse terminology from [RFC9576].

The following terms are used throughout this document:

Flow: Direction from PrivateToken issuance to its redemption. The entity starting the flow acts as an Issuer, while the end of the flow acts as an Origin. The Client is always included, as it finalises the CredentialResponse, and coordinate interactions.

Initial Flow: Issuer -> Attester -> Client -> Origin. This flow produces a PrivateToken that is used by the Origin to kickstart a Reverse Flow.

Reverse Flow: Issuer <- Attester <- Client <- Origin. This flow allows the Origin to issue a PrivateToken. In the reverse flow, the Origin operates one or more Issuer, and the Client MAY provide these tokens either to the Initial Attester/Issuer, or use them against the Origin

Initial Attester/Issuer: Attester/Issuer part of the Initial Flow

Origin Issuer: Issuer operated by the Origin

Origin PrivateToken: PrivateToken issued by the Origin

Reverse Origin: An entity that consumes the Origin PrivateToken. It can be the Origin, or the Initial Attester/Issuer

4. Architecture overview

Along with sending their PrivateToken for authentication (as specified in [RFC9576]), Client sends CredentialRequest

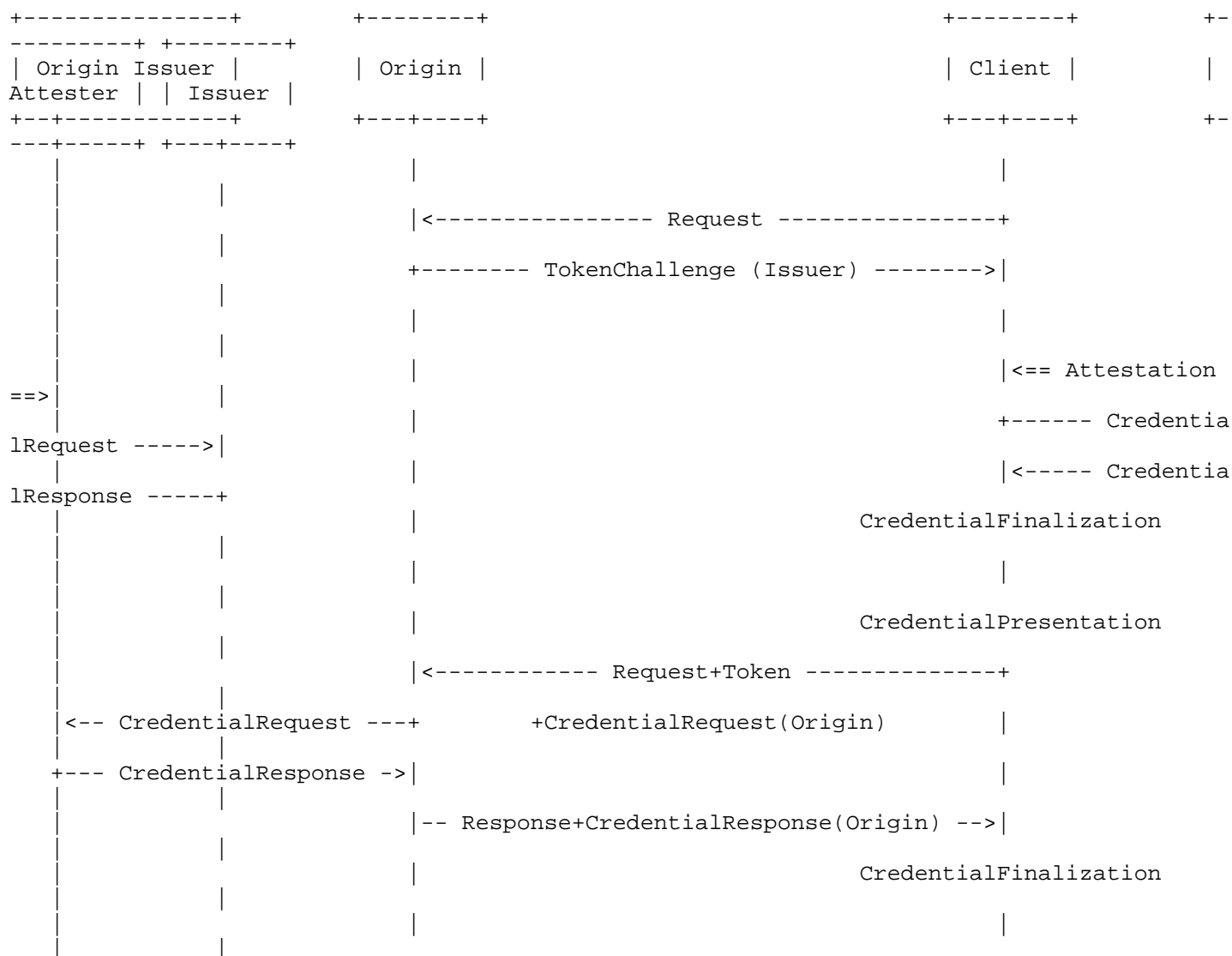


Figure 1: Architecture of Privacy Pass with a Reverse Flow

The initial flow matches the one defined by [RFC9576]. A Client gets challenged when accessing a resource on an Origin. The Client goes to the Attester to get issued a Token.

Through configuration mechanism not defined in this document, the Client is aware the Origin acts as a Reverse Flow Issuer.

This is an extension of [RFC9576]. The redemption flow of a Privacy Pass token is defined in Section 3.6.4 of [RFC9576]. Reverse flow extends this so that redemption flow is interleaved with the issuance flow described in Section 3.6.3 of [RFC9576]. This is denoted in the diagram above by the Client sending Request+Token+CredentialRequest(Origin). The Origin runs the issuance protocol, and returns Response+CredentialResponse(Origin).

Such flow can be performed through various means. This document introduces one to serve as example and first basis.

5. CredentialRequest, CredentialResponse, and CredentialFinalization

In Figure 1, the Client sends an CredentialRequest and receives an

CredentialResponse. These are meant to abstract request from different protocol to the Issuer.

As specified in Section 3.5 of [RFC9576],

The structure and semantics of the TokenRequest and TokenResponse messages depend on the issuance protocol and token type being used.

The introduction of Privacy Pass issuance protocol based on Anonymous Credentials, such as [PRIVACYPASS-ARC] or [PRIVACYPASS-ACT], modifies TokenRequest (resp. TokenResponse) to use CredentialRequest instead (resp. CredentialResponse).

Upon receiving an CredentialResponse, the Client has to finalise the Token so it can be presented to an Origin. This may be a Finalization for type 0x0002 as defined in Section 7 of [RFC9578], a presentation for Section 7.3 of [PRIVACYPASS-ARC], or even a TokenRefund for [PRIVACYPASS-ACT].

All three examples ensure that an Issuer provides the Client with a state update that it needs to finalize, and present.

6. Reverse flow with an HTTP header

This section defines a Reverse Flow, as presented in Section 4, leveraging PrivacyPass-Reverse HTTP header.

CredentialRequest(Origin) and CredentialResponse(Origin) are transmitted through HTTP Header PrivacyPass-Reverse. PrivacyPass-Reverse is a base64url ([RFC4648]) encoded GenericBatchTokenRequest (resp. GenericBatchTokenResponse) as defined in Section 6.1 of [BATCHED-TOKENS] (resp. Section 6.1 of [BATCHED-TOKENS]).

Below is an example request that uses [RFC9577] to pass the request Token, as well as PrivacyPass-Request for its reverse flow.

```
GET /foo HTTP/1.1
Host: example.com
Authorization: PrivateToken token="abc..."
PrivacyPass-Reverse: "def..."
```

```
HTTP/1.1 200 OK
PrivacyPass-Reverse: "001..."
```

[BODY]

6.1. Client behaviour

Along with sending a finalised token from the Initial Issuer to the Origin that it sends through an authorization response as defined in [RFC9577], the Client may send a TokenRequest as defined in [RFC9578], [BATCHED-TOKENS], or [PRIVACYPASS-ARC]. In all these definitions, CredentialRequest MUST be prepended by a uint16_t representing the token type.

The same security and privacy guarantees applies as to the initial issuance flow. The Client is responsible to coordinate between the different entities. Specifically, if the Reverse Origin is the Initial Attester/Issuer, the Client SHOULD account for possible privacy leakage.

6.2. Origin/Issuer/Attester deployment

In this model, the Origin, Attester, and Issuer are all operated by the same entity, as shown in Figure 2. The Reverse Flow is the same as the Initial Flow, except for the request/response encapsulation. The Origin is the Reverse Origin.

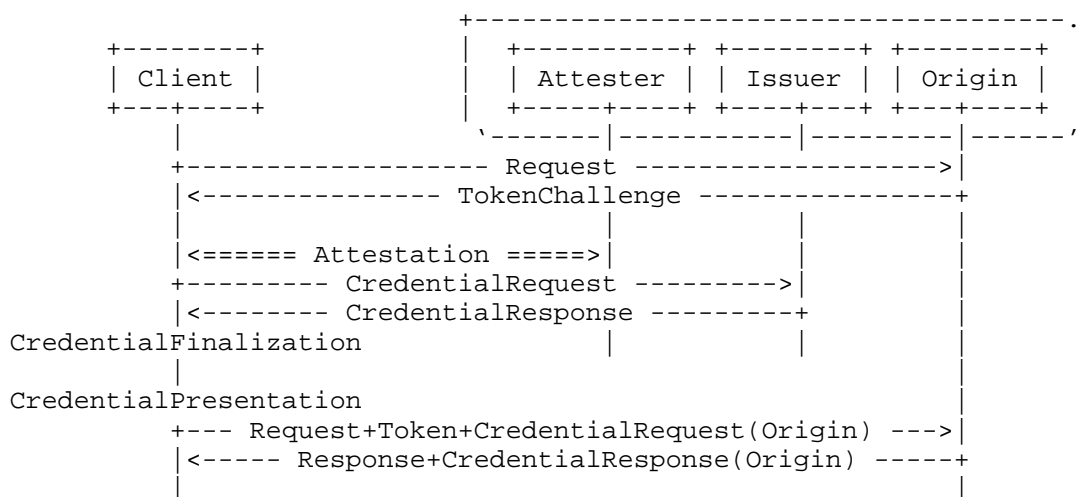


Figure 2: Shared Deployment Model

Similar to the original Shared Deployment Model (Section 4.1 of [RFC9576]), the Attester, Issuer, and Origin share the attestation, issuance, and redemption contexts. Even if this context changes between the Initial and Reverse Flow, attestation mechanism that can uniquely identify a Client are not appropriate as they could lead to unlinkability violations.

6.3. Split Origin-Attester deployment

In this model, the Attester and Issuer are operated by the same entity that is separate from the Origin. The Origin trusts the joint Attester and Issuer to perform attestation and issue Tokens. Origin Tokens can then be sent by Client on new requests, as long as the Reverse Origin trusts the Origin to perform attestation and issue Tokens.

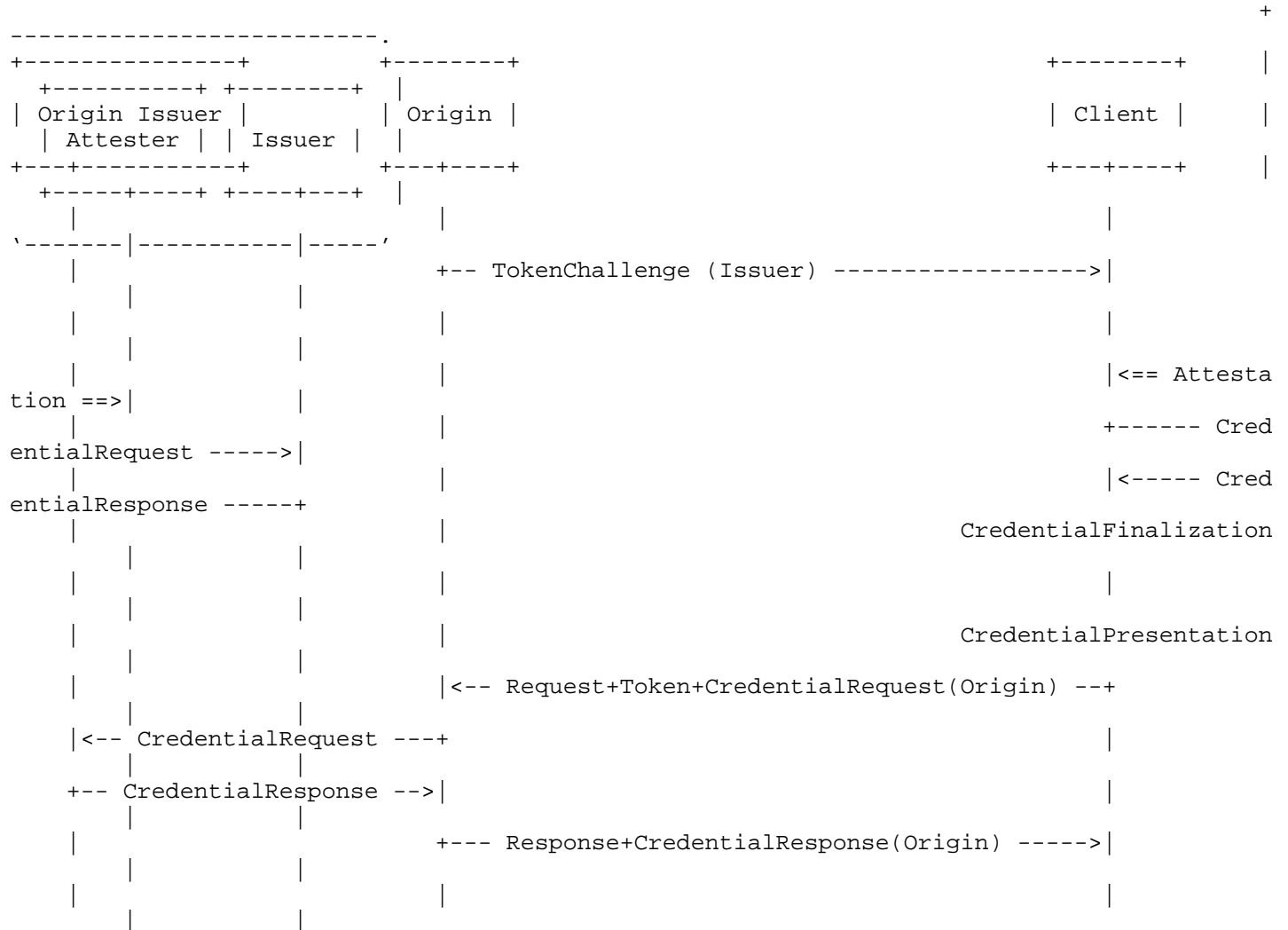


Figure 3: Joint Attester and Issuer Deployment Model

The Origin Issuer MUST NOT issue privately verifiable tokens, as this would lead to secret material being shared between the Origin and the Reverse Origin.

A particular deployment model is when the Reverse Origin is the Attester/Issuer. This model is described in Figure 4

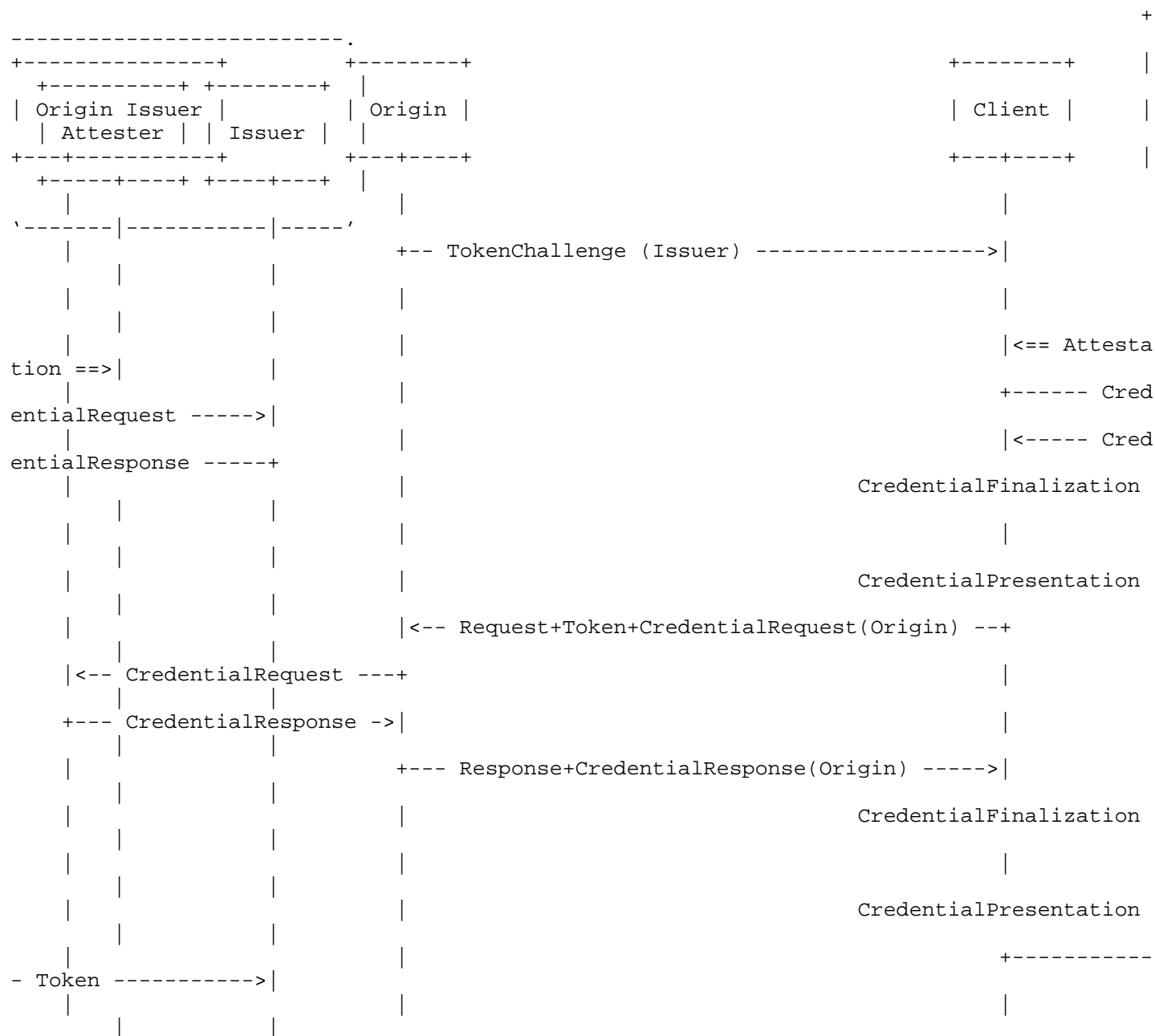


Figure 4: Joint Attester and Issuer Deployment Model with reverse

This deployment SHOULD not allow the Reverse Origin such as an Initial Issuer to infer the request made to the Origin, as it would break unlinkability.

7. Privacy Considerations

Privacy Pass [RFC9576] states

In general, limiting the amount of metadata permitted helps limit the extent to which metadata can uniquely identify individual Clients. Failure to bound the number of possible metadata values can therefore lead to a reduction in Client privacy. Most token types do not admit any metadata, so this bound is implicitly enforced.

In Privacy Pass with a reverse flow, Clients are provided with new PrivateTokens depending on their request. They can present these tokens to continue making further requests.

While the tokens are still unlinkable, the `token_key_id` associated to them represent metadata. It leaks some information about the Client. The following subsections discuss the issues that influence the anonymity set, and possible mitigations/safeguards to protect against this underlying problem.

7.1. Issuer face values

When setting up a reverse flow deployment, an Origin MAY operate multiple Issuers, and assign them some metadata to them. The amount of possible metadata grows as $2^{(\text{origin_issuers})}$.

We RECOMMEND that:

1. Origins define their anonymity sets, and deploy no more than $\log_2(\# \text{anonymity_sets})$. This bounds the possible anonymity sets by design.
2. Clients to only send 1 PrivateToken per request. This is consistent with Section 3.2 of [RFC9577] and Section 11.6.2 of [RFC9110] which only allows one challenge response to be provided as part of Authorization HTTP header.
3. Issuers metadata to be publicly disclosed via an Origin endpoint, and externally monitored.

7.2. Token for specific Clients

In Privacy Pass with a reverse flow, an Origin MAY operate multiple Issuers, with arbitrary metadata associated to them. A malicious Origin MAY uses this opportunity to associate certain token values to a specific set of Clients.

Let's consider the following deployment: the Origin operates two Issuers A and B. The Client sends `Token_A`, and (`CredentialRequest_A`, `CredentialRequest_B`). Issuer B is associated to people that like croissant. Issuer A is for the rest of the clients.

If a Client requests croissant, or sends `Token_B`, the Origin provides `CredentialResponse_B`. If not, it provides `CredentialResponse_A`.

Over time, this means the Origin is able to track people that like croissants.

To mitigate this, we RECOMMEND:

1. The initial PrivateToken to be provided by an Issuer not in control of the Origin.

2. Clients to reset their state regularly with the initial Issuer.

7.3. Swap endpoint and its privacy implication

With multiple Issuers, a Client MAY end up with a bunch of tokens, for various Issuers. Origins MAY propose a swap endpoint at which a Client can exchange one or more Origin tokens against one or more new Origin tokens.

The Origin SHOULD ensure this endpoint receives enough traffic to not reduce the anonymity sets.

8. IANA Considerations

This document has no IANA actions.

9. References

9.1. Normative References

[BATCHED-TOKENS]

Robert, R., Wood, C. A., and T. Meunier, "Batched Token Issuance Protocol", Work in Progress, Internet-Draft, draft-ietf-privacypass-batched-tokens-06, 20 October 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-privacypass-batched-tokens-06>>.

[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>>.

[RFC9576] Davidson, A., Iyengar, J., and C. A. Wood, "The Privacy Pass Architecture", RFC 9576, DOI 10.17487/RFC9576, June 2024, <<https://www.rfc-editor.org/rfc/rfc9576>>.

[RFC9578] 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>>.

9.2. Informative References

[PRIVACYPASS-ACT]

Schlesinger, S. and T. Meunier, "Privacy Pass Issuance Protocol for Anonymous Credit Tokens", Work in Progress, Internet-Draft, draft-schlesinger-privacypass-act-01, 13 February 2026, <<https://datatracker.ietf.org/doc/html/draft-schlesinger-privacypass-act-01>>.

[PRIVACYPASS-ARC]

Yun, C., Wood, C. A., and A. F. Faz-Hernandez, "Privacy Pass Issuance Protocol for Anonymous Rate-Limited Credentials", Work in Progress, Internet-Draft, draft-ietf-privacypass-arc-protocol-00, 4 February 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-privacypass-arc-protocol-00>>.

[PRIVACYPASS-BBS]

Ladd, W., "BBS for PrivacyPass", Work in Progress, Internet-Draft, draft-ladd-privacypass-bbs-01, 26 February 2024, <<https://datatracker.ietf.org/doc/html/draft-ladd-privacypass-bbs-01>>.

[RFC9110] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <<https://www.rfc-editor.org/rfc/rfc9110>>.

[RFC9577] Pauly, T., Valdez, S., and C. A. Wood, "The Privacy Pass HTTP Authentication Scheme", RFC 9577, DOI 10.17487/RFC9577, June 2024, <<https://www.rfc-editor.org/rfc/rfc9577>>.

Acknowledgments

The author would like to thank Tommy Pauly, Chris Wood, Raphael Robert, and Armando Faz Hernandez for helpful discussion on Privacy Pass architecture and its considerations.

Changelog

v02

- * Diagrams now use Credential instead of Token, and use both Finalization and Presentation as keyword
- * Rework the intro to make it consistent with Anonymous credentials evolutions

- * Have Anonymous credentials use case, given it needs a new architecture
- * Editorial pass on PrivacyPass-Reverse header

v01

- * Editorial pass on the introduction
- * Add a motivation section: refunding tokens, bootstrapping issuer, attester feedback loop
- * Split protocol overview via HTTP headers in its own section
- * Add consideration about anonymous credentials in joint origin/issuer deployment

v00

- * Initial draft
- * Possibility of a new HTTP request for inlining request
- * Privacy considerations about additional metadata

Author's Address

Thibault Meunier
Cloudflare Inc.
Email: ot-ietf@thibault.uk