Network Working Group
Internet-Draft
Intended status: Informational
Expires: 10 October 2026

E. Rescorla
Independent
R. L. Barnes
Cisco

8 April 2026

Anonymous Bot Authentication: Authorization and Rate Limiting for Web Agents
draft-rescorla-anonymous-webbotauth-00

Abstract

Automated agents ("bots") represent a large fraction of the traffic to many Web sites. In some cases, this traffic is desired, in others undesired, and in yet others, desired as long as it remains within certain rate limits. This memo describes Anonymous Bot Authentication (ABA), a system that allows Web site operators to distinguish wanted from unwanted traffic, while not tying a given request to a specific sender.

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://ekr.github.io/draft-rescorla-anonymous-webbotauth/draft-rescorla-anonymous-webbotauth.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-rescorla-anonymous-webbotauth/.

Source for this draft and an issue tracker can be found at https://github.com/ekr/draft-rescorla-anonymous-webbotauth.

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 10 October 2026.

Table of Contents

1. Introduction

DISCLAIMER: This is a work-in-progress draft and has not yet seen significant security analysis. It is being published at an early stage for discussion purposes.

Automated agents ("bots") represent a large fraction of the traffic to many Web sites. For example, Wikimedia reports that about 35% of its traffic is from bots [PCMag-Wikipedia] and Cloudflare reports that over 50% of traffic is automated [Cloudflare-2025]. High traffic loads--even when from otherwise desired bots--can have negative impacts on sites in terms of performance, stability, and operational costs.

In addition, there may be certain types of bot traffic that sites wish to heavily restrict or block entirely. For example:

* Volumetric traffic intended to create a DoS attack on the site.

* Non-volumetric traffic intended to attack the site.

* Scraping for specific purposes such as training AI agents.

[I-D.nottingham-webbotauth-use-cases] provides a more complete list of potential use cases.

In response to this approach, there have been proposals to authenticate automated clients. Sites can use behavioral analysis to determine when traffic from a given endpoint appears bot-like (e.g., is high volume, has low latency between requests, appears to be retrieving the entire site, etc.) and restrict access by those endpoints unless they authenticate and the resulting identity is acceptable to the site. [I-D.meunier-webbotauth-registry] describes one such architecture, where identities are rooted in the DNS and bots use digital signatures to tie their activity to a given identity.

While directly identifying each bot allows the site to precisely monitor bots' behavior and restrict or block bots that it believes are misbehaving, it also has potential negative effects on the open Web, including:

* Allowing sites to precisely discriminate against specific bots, even when those bots are acting in the public interest. For example:
  - Government sites might block bots which are tracking law enforcement activity.
  - Employment or housing sites might block bots which are monitoring for discriminatory behavior.
  - Shopping sites might block bots used for price comparison.
* Preventing users and groups from being able to anonymously retrieve Web content using automated tools.

In many use cases, it is possible to resolve this tension using anonymous credentials. When a bot presents an anonymous credential, the Web site learns that the bot is part of some set of authorized bots -- say those whose operators have agreed to good behavior -- and not the specific identity of the bot or the bot operator.

This document describes an approach for using anonymous credentials to authenticate bots to websites. In addition to laying out how the system works, we describe the trade-offs between anonymity and fine-grained abuse mitigation.

2. Architectural Overview

The overall structure of ABA is shown in Figure 1.

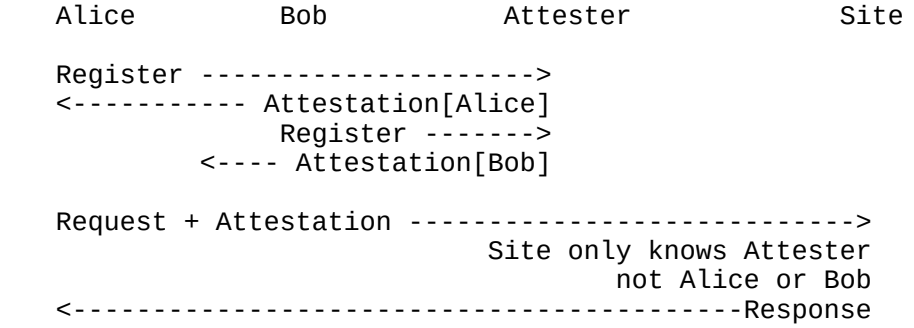

Figure 1: Architectural Overview

Prior to contacting the site, Alice and Bob both register with a credential Attester, which is responsible for evaluating whether they comply with the Attester's policies. They each are issued an anonymous credential, which they can use to authenticate to the server. What makes the credential anonymous is that the site only learns that someone with a credential from the Attester is authenticating, not whether it's Alice or Bob. This prevents the site from discriminating _between_ customers of the same attester, although it can discriminate between attesters.

As a result, while it is possible to distinguish authenticated from unauthenticated bots, it is not possible to use the authentication method to either (1) selectively block individual bots(2) determine which bots are accessing which resources or (3) link multiple visits by the same bot. It may still be possible to identify bots via other mechanisms such as IP address or fingerprinting.

In some cases it may be sufficient merely to identify the attester, for instance if the attester performs some vetting to ensure policy compliance. However, in some cases this may be insufficient, as discussed below.

2.1. Trade-offs

Assuring the anonymity of bot requests means imposes some limits on how this authentication mechanism can be used to counter abuse. These trade-offs are discussed in detail in Section 6, and the below bullets provide a summary:

* ABA supports:
  - Rate-limiting of requests by a a single bot
  - Providing separate content to bots vs. humans
  - Conditioning access based on a bot meeting certain criteria
  - Conditioning access based on participation in some scheme or protocol
  - Classifying human vs. bot traffic
  - IP address mobility and sharing of IP addresses
  - Unlinkability between authenticated requests by the same client
* ABA does not support:

- Allow lists / deny lists

- Auditing bot behavior

- Authenticating site services

Essentially, abuse controls that rely on knowing the identity of the abusive party are incompatible with anonymous authentication.

2.2. Rate Limiting

In some cases it may be desirable to limit any individual agent to a specific number of requests. For example, a given Attester may have a large number of subscribers but only a few may access a given site. In this case, overall rate limits for an attester will not be effective, but individual rate limits are.

3. Concrete Implementation With Privacy Pass and ARC

This section describes how to implement ABA using Privacy Pass [RFC9576], and Anonymous Rate-Limited Credentials [I-D.ietf-privacypass-arc-protocol] [I-D.ietf-privacypass-arc-crypto]. While this is not the only possible implementation approach, it leverages existing deployed and proposed IETF technologies and thus avoids duplicated functionality and fits well into existing deployments.

We make use of the "Joint Origin and Issuer Deployment Model" from Section 4.3 of [RFC9576], reproduced below in Figure 2:

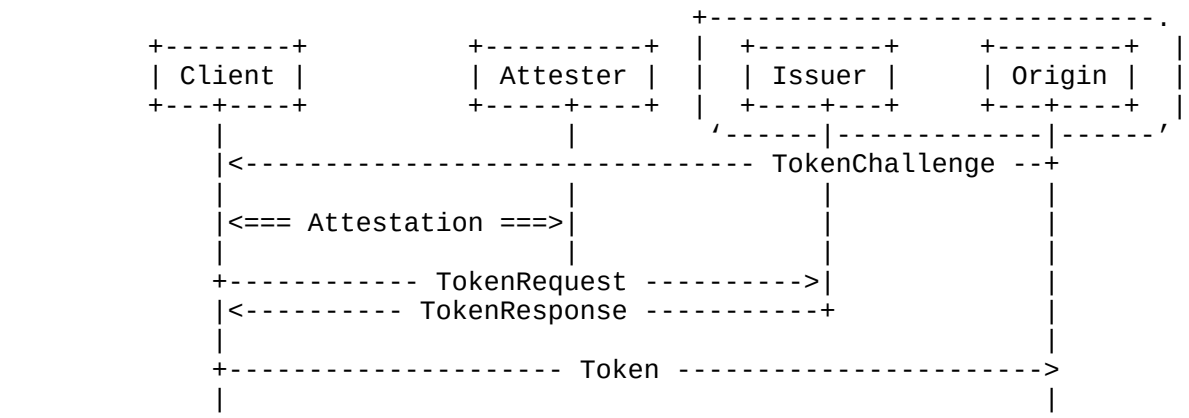

Figure 2: Privacy Pass Model (Joint Origin and Issuer)

In this model, the Client interacts with an Attester, which is responsible for determining whether the client conforms to the required policy. The Attester provides an Attestation which can then be presented to the site (the Issuer in the diagram above), which provides a Token. The bot can then use the Token to get services from the site (the Origin in the diagram above). The Issuer and the Origin are operated by the same entity (this is a technical constraint of ARC).

ABA extends this scheme with a zero-knowledge proof (ZKP) system (e.g., [I-D.google-cfrg-libzk]) in order to assure that the Attester's attestation does not leak information to the Issuer that could deanonymize the Client. This property is especially important in the "server as attester" case Section 4.2, and is not assured by ARC itself. In effect, ABA uses an expensive attestion and ZKP operation to authorize the issuance of ARC tokens that can be used cheaply on every request.

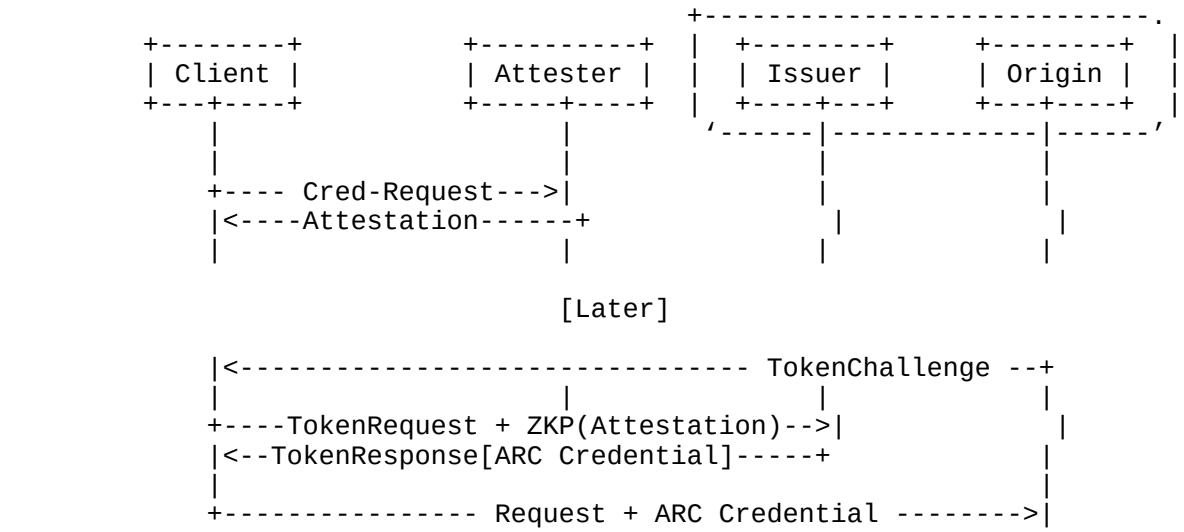

Figure 3: ABA with Privacy Pass

When a new Client is deployed, it first must register with some set of Attesters. These Attesters will require the bot to demonstrate that it complies with their policies, for instance that it is a registered corporation, holds a domain name or an IP address block, etc. Once the Attester is satisfied, it issues a Attestation to the Client in the form of a CWT [RFC8392] signed by the Attester. This Attestation can be used to authenticate to an arbitrary number of Issuers.

When a client contacts a new site for which it does not yet have an ARC Credential, client uses the Attestation to authenticate to the Issuer and request an ARC Credential. This authentication is performed anonymously using a zero-knowledge proof that it has a valid Attestation (shown as "ZKP(Attestation)" above), as described in Section 3.1, so that the Issuer only learns the following information:

1. This Client has been authenticated by the Attester.

2. This Client has not authenticated to the Issuer previously using this Attestation (potentially within a given time window).

Assuming that the Client’s proof verifies correctly and the Attester is acceptable, the Issuer issues an ARC credential.

Each credential is associated with a rate limit, which may be Attester-dependent. For instance, if the Attester has a policy designed for high traffic bots, the Issuer might use one rate limit, whereas if the policy is designed for low traffic bots, the Issuer might use a lower rate limit. Note that the choice of rate limit is entirely up to the Issuer, but because all Clients authenticated by a given Attester are within the same anonymity set, it cannot use different per-Client rate limits to multiple Clients attested to by the same Attester. Similarly, because all clients whose credentials are associated with the Issuer’s key pair, all authorizations by Clients using the

Once the Client has an ARC credential it can use it to authenticate to the Origin repeatedly up to the number of authentications in the rate limit associated with the Credential. These authentications are unlinkable provided that the Client does not exceed the rate limit; authentications beyond the rate limit are linkable. Because any Credential can only be used once for a given Issuer within a given time window, the total rate limit for a given Client/Issuer pair is bounded by the limit associated with the ARC credential.

3.1. Providing Attestation to the Issuer

The protocol for Attestation to the Issuer is designed to meet the following requirements:

* A Attestation can be used with an arbitrary number of Issuers.

* A Attestation can only be used once with a single Issuer within a given time window

* Attestation presentations are unlinkable, both between Issuer and Attester and between Issuers.

These requirements can be met by using a signed credential (in this case, a CWT [RFC8392]), along with a generic circuit-based zero-knowledge proof system (in this case Longfellow-ZK [I-D.google-cfrg-libzk]). The interaction between the Attester and the Client just yields an ordinary CWT and then the Client proves in zero-knowledge that they have a valid Attestation from a given Attester.

In order to prevent replay, the proof also includes an Issuer-specific nullifier tied to the Issuer's domain name and the current time window. While the proof itself varies between presentations, the nullifier remains the same and therefore can be used to detect replay.

Because we are using a generic ZK system it should also be possible to use it to conceal the Attester if desired: for example if there are multiple Attesters who follow equivalent vetting procedures, then the Issuer does not need to know which Attester the Client used; the ZKP can be designed to prove that the Client has a Attestation from one of a set of Attesters within the same equivalence class; of course, the Issuer will need to use the same rate limit for all such Attesters.

3.2. Alternate Cryptographic Approaches

This section considers a number of alternate cryptographic approaches and explains the choice of primitives in this section.

3.2.1. Only Generic Zero-Knowledge Proofs

In principle, it is possible to omit the use of ARC and instead have the Client authenticate each transaction directly to the Origin. However, even efficient generic ZKP systems like Longfellow-ZK have far higher computational and bandwidth costs than more limited systems such as the one used in ARC. Using a generic ZKP system to demonstrate ownership of an attested Attestation to the Issuer and then using that single interaction to obtain an ARC credential makes it possible to amortize the generic proof over a large number of subsequent interactions.

3.2.2. Other Attestation Structures

There are a number of other potential ways to convey attestations to the Issuer, but each fails to fulfill one of the requirements in Section 3.1.

* _Non-anonymous credentials_ allow for linkage of bots at redemption time.

* _Single-show credentials_ like those in Privacy Pass either allow or require the bot to interact with the Attester for each site it wants to interact with (otherwise linkage is possible).

* _Camenisch-Lysanskaya credentials_ do not prevent multiple shows to the same server, thus precluding rate limiting.

* _BBS+_ credentials require the use of pairings and have no obvious transition to post-quantum algorithms.

Note that in cases where the Attester and the Issuer are one and the same (see Section 4.2), it is not necessary to convey the attestation to the Issuer and so the generic ZKP system can be omitted.

4. Issuance Models

Anonymous credentials are compatible with a variety of issuance models, as discussed in this section.

4.1. Independent Attesters

Much like millions of websites rely on a much smaller number of independent WebPKI certificate authorities, it is possible to have a system of independent Attesters that are broadly relied upon by many websites. Each attester could publish the policy that it uses to issue credentials (e.g., verifying corporate existence, subject pays $100, etc.). Sites can then select which attesters have policies they are willing to accept. It is also possible to have multiple Attesters who conform to a common set of policies, as in the WebPKI, where each CA has to meet the same requirements, but site operators have the choice of which CA to use.

4.2. Server as Attester

It is also possible for a server to act as their own Attester. This is not likely to be practical for small sites, as bots will simply opt not to authenticate to those sites at all. However, a large CDN which hosts many sites might opt to operate its own Attester, and it could be practical for bots to register with such an Attester. Note that this approach makes it possible for the Attester to refuse service to individual Clients, but not to selectively do so for individual customers unless it uses separate Issuers for each of those customers.

The ZKP element of ABA ensures that clients remain anonymous even in this case, since the client's redemption of an Attestation is not linkable to its issuance.

4.3. Number of Attesters

In general, it is desirable for bots to be able to acquire a relatively small number of credentials and have high confidence that those credentials will be compatible with most if not all of the sites that the bot wishes to contact. This can be most straightforwardly accomplished if there is only a small number of attesters, but it can also work if there are a larger number of attesters but sites converge on a relatively small number of policies (expressed as which attesters they support) such that a bot can acquire a set of credentials that covers all of those policies. The least desirable outcome is if bots routinely are prompted to provide credentials for a new attester.

5. Relationship to Browser Authentication

There is significant overlap with existing work on anonymous authentication happening in the PrivacyPass WG and in the W3C AntiFraud Community Group. While that work is directed towards access control for users, the same cryptographic techniques can be used to authenticate bots. A good description of the vision and requirements can be found at [PACT-Issue]. The ideal scenario would be to be able to use compatible tokens for users and bots, differing only in the issuance policies, the attesters, and the rate limits.

6. Use Case Analysis

[I-D.nottingham-webbotauth-use-cases] describes a set of use cases for web bot authentication. The architecture described in this document address some but not all of these use cases. This is intentional rather than a deficiency; the objective is to address use cases which are compatible with limiting the negative impact of bots while avoiding making it trivial for sites to discriminate against individual bots. The remainder of this section addresses each use case individually.

6.1. Site Use Cases

6.1.1. Mitigating Volumetric Abuse by Bots

This document directly addresses the topic of volumetric abuse, because bots can be authenticated and authenticated bots can be restricted to specific bandwidth limits. Once a bot has exceeded its limit, it can be blocked.

6.1.2. Controlling Access by Bots

[I-D.nottingham-webbotauth-use-cases] provides the following example applications of controlling access:

- Only allow access by bots on an allow list;
- Disallow access to bots on an explicit deny list;
- Condition access upon meeting some criteria (e.g., non-profit, certification by a third party);
- Condition access upon participation in some scheme or protocol (e.g., payment for access);

Note that the first two imply some notion of bots being tied to a real-world identity, whereas the remaining do not necessarily require it.

ABA can be used for the second two use cases and can be used for some versions of the first two use cases. However, the more fine-grained controls are used (e.g., by having many Attesters with different policies) the worse the privacy and scalability properties of the system become.n

6.1.3. Providing Different Content to Bots

The architecture in this document may be usable to provide different content to bots generally than humans depending on the structure of attesters (e.g., does a given attester issue to both bots and to humans) and whether techniques are used to conceal which attester is in use. However, they are not generally useful to provide different content to specific bots.

6.1.4. Auditing Bot Behavior

This use case is not addressed by this document.

6.1.5. Classifying Traffic

Because this use case does not depend on determining which bot is which, but only which traffic is human versus bot, the architecture in this document may be able to address this use case, depending on the ultimate deployment model.

6.1.6. Authenticating Site Services

This use case is not addressed by this document.

6.2. Bot Use Cases

6.2.1. IP Address Mobility

ABA provides authentication for bots independent of IP address. Because ABA authentication is at the Attester rather than the Client level, it is not possible to use ABA to build bot-specific reputations based on observed bot activity; instead the Attester is responsible for assessing bots and then providing that information to sites.

6.2.2. Sharing IP Addresses

Because ABA provides authentication independent of IP address, it allows sites to discriminate between unauthenticated users of an IP address and those which are ABA-authenticated, thus reducing the negative reputational side effects of misuse by unauthenticated users sharing the same IP address.

6.2.3. Robots.txt Alignment

This use case is not addressed by this document.

6.2.4. Conveying Contextual Information

Because different Attesters can have different policies and Attesters can have multiple policies, ABA allows for limited conveyance of contextual information. While in principle this information can be arbitrarily fine-grained, coarse-grained information ensures a larger anonymity set (see Section 4.3).

7. 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.

8. Security Considerations

The precise security and privacy details of a system of this type depend on the detailed cryptographic mechanism being deployed. However, it is possible to make some general observations.

8.1. Anonymity Set

The anonymity set for a given transaction is the set of credentials associated with a given Attester, or, if Attester hiding is used, the set of credentials associated with the set of attesters. However, it is still possible to learn information about the client by manipulating the Attester set. For example, a site acting as an Attester could use different keys for each user or a site could use different Attester subsets to identify which of a set of Attesters was in use. Transparency/consistency mechanisms like [I-D.ietf-privacypass-key-consistency] may be useful in detecting this form of attack.

8.2. Credential Misuse

Because clients are anonymous, some forms of misuse are harder to manage. For example:

* Two clients can collude to exceed rate limits if they are interacting with disjoint sites.

* If registration standards are low and registration is cheap a bot can obtain multiple credentials.

* Patterns of misuse (e.g., credential stuffing) become harder to detect.

Note that existing mechanisms, such as IP address, will continue to be usable, but the techniques described in this document may not add to the server’s ability to address these issues.

9. IANA Considerations

This document has no IANA actions.

10. References

10.1. Normative References

[I-D.google-cfrg-libzk]
Frigo, M. and A. shelat, "The Longfellow Zero-knowledge Scheme", Work in Progress, Internet-Draft, draft-google-cfrg-libzk-01, 2 September 2025, <https://datatracker.ietf.org/doc/html/draft-google-cfrg-libzk-01>.

[I-D.ietf-privacypass-arc-crypto]
Yun, C., Wood, C. A., and A. F. Faz-Hernandez, "Anonymous Rate-Limited Credentials Cryptography", Work in Progress, Internet-Draft, draft-ietf-privacypass-arc-crypto-01, 2 March 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-privacypass-arc-crypto-01>.

[I-D.ietf-privacypass-arc-protocol]
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-01, 2 March 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-privacypass-arc-protocol-01>.

[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>.

[RFC8392] Jones, M., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "CBOR Web Token (CWT)", RFC 8392, DOI 10.17487/RFC8392, May 2018, <https://www.rfc-editor.org/rfc/rfc8392>.

[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>.

10.2. Informative References

[Cloudflare-2025]
Cloudflare, "Cloudflare 2025 Year in Review", 2025, <https://radar.cloudflare.com/year-in-review/2025>.

[I-D.ietf-privacypass-key-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>.

[I-D.meunier-webbotauth-registry]
Guerreiro, M., Kirazci, U., and T. Meunier, "Registry and Signature Agent card for Web bot auth", Work in Progress,

Internet-Draft, draft-meunier-webbotauth-registry-01, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-meunier-webbotauth-registry-01>.

[I-D.nottingham-webbotauth-use-cases]
Nottingham, M., "Use Cases for Authentication of Web Bots", Work in Progress, Internet-Draft, draft-nottingham-webbotauth-use-cases-02, 1 April 2026, <https://datatracker.ietf.org/doc/html/draft-nottingham-webbotauth-use-cases-02>.

[PACT-Issue]
Jackson, D., "Private Access Control Tokens", 2 December 2025, <https://github.com/antifraudcg/proposals/issues/22>.

[PCMag-Wikipedia]
Kan, L., "Wikipedia Faces Flood of AI Bots That Are 'Eating Bandwidth,' Raising Costs", 14 February 2024, <https://www.pcmag.com/news/wikipedia-faces-flood-of-ai-bots-that-are-eating-bandwidth-raising-costs>.

Acknowledgments

TODO acknowledge.

Authors' Addresses

Eric Rescorla
Independent
Email: ekr@rtfm.com

Richard L. Barnes
Cisco
Email: rlb@ipv.sx