Web Bot Auth
Internet-Draft
Intended status: Informational
Expires: 21 December 2025

T. Meunier
Cloudflare

19 June 2025

Web bot auth Glossary
draft-meunier-web-bot-auth-glossary-01

Abstract

Automated traffic authentication presents unique security challenges, constraints, and opportunities that impact all Internet users. This document seeks to collect terminology and examples within the space, with a specific focus on AI related technologies.

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-glossary/draft-meunier-web-bot-auth-glossary.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-meunier-web-bot-auth-glossary/.

Discussion of this document takes place on the Web Bot Auth Working Group mailing list (mailto:web-bot-auth@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/web-bot-auth/. Subscribe at https://www.ietf.org/mailman/listinfo/web-bot-auth/.

Source for this draft and an issue tracker can be found at https://github.com/thibmeu/draft-meunier-glossary-somehow.

Status of This Memo

This Internet-Draft will expire on 21 December 2025.

Table of Contents

1. Introduction

Agents are increasingly used in business and user workflows, including AI assistants, search indexing, content aggregation, and automated testing. These agents need to reliably identify themselves to origins for several reasons:

1. Regulatory compliance requiring transparency of automated systems

2. Origin resource management and access control

3. Protection against impersonation and reputation management

4. Service level differentiation between human and automated traffic

Current identification methods such as IP allow-listing, User-Agent strings, or shared API keys have significant limitations in security, scalability, manageability, and fairness. This document presents these examples, as well as possible paths to address them.

2. Motivation

There is an increase in agent traffic on the Internet. Many agents choose to identify their traffic today via lists of IP Addresses and/or unique User-Agents. This is often done to demonstrate trust and safety claims, support allow-listing/deny-listing the traffic in a granular manner, and enable sites to monitor and rate limit per agent operator. However, these mechanisms have drawbacks:

1. User-Agent, when used alone, can be spoofed meaning anyone may attempt to act as that agent. It is also overloaded - an agent may be using Chromium and wish to present itself as such to ensure rendering works, yet it still wants to differentiate its traffic to the site.

2. IP blocks alone can present a confusing story. IPs on cloud plaforms have layers of ownership - the platform owns the IP and registers it in their published IP blocks, only to be re-published by the agent with little to bind the publication to the actual service provider that may be renting infra. Purchasing dedicated IP blocks is expensive, time consuming, and requires significant specialist knowledge to set up. These IP blocks may have prior reputation history that needs to be carefully inspected and managed before purchase and use.

3. An agent may go to every website on the Internet and share a secret with them like a Bearer from [OAUTH-BEARER-RFC]. This is impractical to scale for any agent beyond select partnerships, and insecure, as key rotation is challenging and becomes less secure as the consumers scale.

Using well-established cryptography, we can instead define a simple and secure mechanism that empowers small and large agents to share their identity.

3. 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.

Agent An autonomous entity that perceives the environment and can take actions on behalf of users.

Bot A type of agent that operates automatically, often performing repetitive tasks. Bots may identify themselves or attempt to mimic human behavior.

Origin (Server) The primary server hosting the web content or service that an agent intends to access.

Application Firewall Controls incoming traffic to an origin based on a set of rules. This may include but is not limited to IP filtering, User-Agent matching, or cryptographic signature verification.

Reverse proxy An intermediary server that forwards client requests to the origin server, often performing functions like load balancing, authentication, or caching.

Browser A client application used to access web content. Browsers may also be orchestrated.

Human A physical person, like you and me.

Rate limit A control mechanism that restricts the access of an Agent to a resource provided by an Origin Server. An Origin can decide to rate limit all connections from an individual Client, from a specific Provider, or to a specific resource. This may be a fixed number of requests, a budget, a time, a location, or legal requirements.

Unlinkability A property ensuring that multiple interactions or credentials from the same agent cannot be correlated by the verifier.

Account Persistent identifier of an entity to an origin. This requires a registration.

Registration The creation of an identity. It can involve one time payment, a subscription, an account with user name/password, an age, a legal jurisdiction, others.

Issuer An entity that generates and provides credentials to agents after the Attester has verified certain attributes.

Attester An entity that evaluates an agent’s characteristics or behavior and provides evidence to an Issuer to support credential issuance.

Verifier An entity that validates the authenticity and integrity of a credential presented by an agent.

4. Web bot authentication categories

We divide web bot authentication in three categories.

4.1. Identifying providers

Organizations operating bots may need to authenticate their agents to access certain web resources. Authentication mechanisms can help distinguish legitimate bots from malicious ones.

Examples:

* Web crawlers wanting to authenticate against origins such as search engines,

* Security companies that want to perform scans to identify malicious URLs,

* AI augmented queries that are looking to identify themselves to a set of newspapers.

4.2. User Account Identification

Bots acting on behalf of registered users may require authentication to access user-specific data or services.

Examples:

* Authenticating and authorizing a known user against particular resources, such as newspapers they have a subscription for,

* Most authorization use cases for [MCP-AUTH] and [A2A-AUTH].

4.3. Attribute-Based Access

In scenarios where full identification is unnecessary or undesirable, agents may present credentials that attest to specific attributes without revealing their identity.

Examples:

* Add a signal to limit visual CAPTCHA challenge such as [PRIVATE-ACCESS-TOKEN],

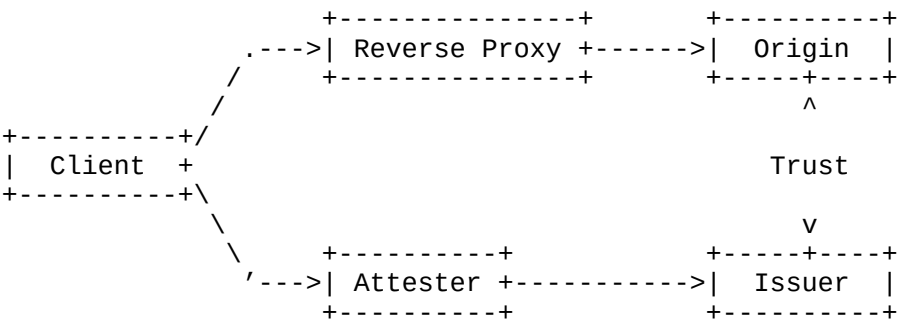
* Gating access to a resource for longstanding users such as [LOX],

* Using a search engine with a fixed number of requests such as [PRIVACYPASS-KAGI],

* Selective disclosure of a credential attribute (location, age) such as [PRIVATE-PROOF-API].

* Redeeming previously issued credits as in [ANONYMOUS-CREDIT-TOKENS].

5. Ecosystem overview

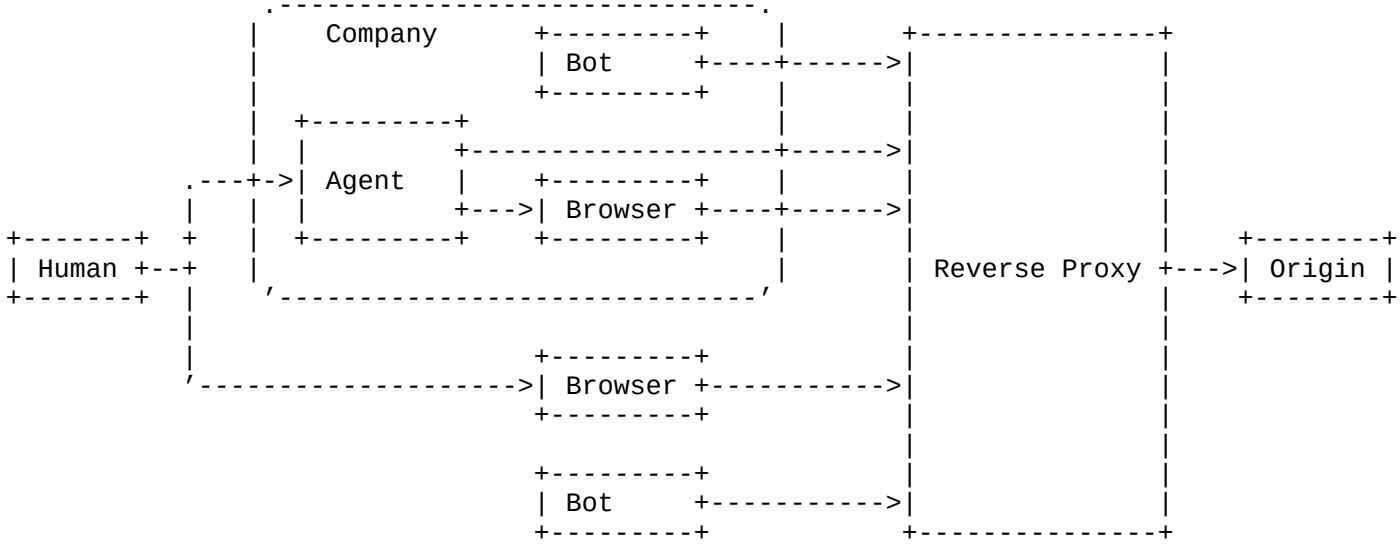


The ecosystem involves multiple actors: a credential issuer that requires an certain criteria to be passed via an attester, the client which can be a bot or human-mediated agent whose IP is unknown, and the web origin placed behind a reverse proxy that may be fronting its infrastructure. The issuer provides cryptographic credentials to the client, which are then linked to requests and optionally verified by proxies before reaching the origin. This chain allows for authentication without necessarily revealing identifying details to each intermediate.

5.1. AI agent use example

Humans and bots often interact with origins indirectly via clients such as browsers, agents, or CLI tools. These clients handle requests, potentially traversing reverse proxies that manage TLS termination, DDoS protection, and caching.

The rise of advanced browser orchestration blurs the line between human-driven and automated requests, making identifying traffic as automated or not increasingly ambiguous.



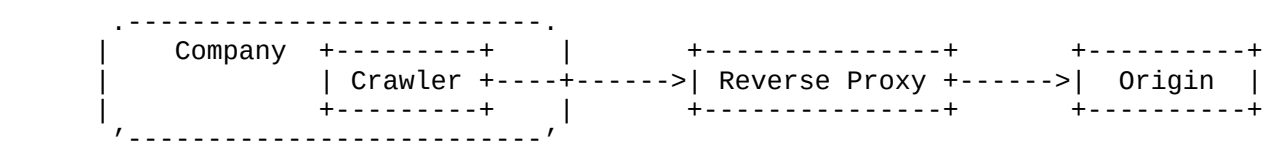
The attester/issuer roles could be filled by the AI company, reverse proxy, origin, or a third party. Origins need mechanisms to identify organizations, rate-limit individuals, and authenticate users without relying solely on client IP or heuristics presented in Section 2.

5.2. Web crawler example

Search engines, web archivers, and security analysis tools operate bots as part of their operations to scan and retrieve content from a variety of sources. These automated agents are commonly referred to as crawlers.

While their traffic is automated, it is often seen as desirable by origins. Legitimate crawlers contribute to improved SEO, enhanced security posture, and broader content reach through search indexing and archiving. Reliable authentication allows origins to differentiate these beneficial crawlers from malicious actors.

In the framework provided in Section 5, the interaction for a web crawler can be described as follows:



Similar to the Section 5.1, the attester/issuer roles could be filled by the crawling company, a reverse proxy, the origin, or a third party. For instance, the crawling company itself is often the most authoritative entity to attest to the legitimacy of its crawlers, issuing cryptographic credentials that identify them. Origins require robust mechanisms to identify these organizations and differentiate their traffic without solely relying on the limitations of IP addresses or User-Agent heuristics presented in Section 2. This enables them to apply granular access controls, rate limits, or preferential treatment based on the verified identity of the crawler.

6. Deployment considerations

The security model includes several actors: credential issuers, attesters, clients (bots or agents), reverse proxies, and origin servers. The primary goals are to prevent impersonation, allow for credential revocation, support delegation and rotation, and maintain trust boundaries.

6.1. Public vs private presentation

If the Issuer is also the Origin or its reverse proxy, it is possible to use shared secrets for verification. In cases where the issuer and verifier are different entities, asymmetric cryptography becomes necessary, allowing the bot to prove its identity using a public key infrastructure.

Such work is being carried out in the CFRG and Privacy Pass working group with the following drafts

Presentation	Cryptography	Privacy Pass
Private	[ARC]	[PRIVACYPASS-ARC]
Private	[OPRF]	[PRIVACYPASS-PROTOCOL]
Public/ Private	[BBS]	[PRIVACYPASS-BBS]
Public	[BLINDRSA]	[PRIVACYPASS-PROTOCOL]
Public	[PARTIALLY-BLINDRSA]	[PRIVACYPASS-PUBLIC-METADATA]

Table 1

6.2. Single vs multi show

Some credentials may be designed for one-time use only (for anti replay or privacy reasons), while others can support multiple presentations through the use of cryptographic derivation techniques. This distinction affects privacy, scalability, and implementation complexity.

6.3. Transport

Authentication tokens may be exchanged at different protocol layers and through different transports. Each may have different deployment, performance, and security guarantees.

For TLS, we have seen [REQ-MTLS] and [PRIVACYPASS-IN-TLS] respectively addressing Section 4.1 and Section 4.3.

For HTTP, we see [HTTP-MESSAGE-SIGNATURE-FOR-BOTS] or [DPOP-AUTH-RFC], and [PRIVACYPASS-HTTP-AUTH-RFC] respectively addressing Section 4.1 and Section 4.3. [OAUTH-BEARER-RFC] fits as well for Section 4.2.

Other methods have been seen such as leveraging a dedicated format on top of a JavaScript API. This is the case for W3C [PRIVATE-STATE-TOKEN] or the more recent [PRIVATE-PROOF-API].

Focusing on AI specifically, it’s worth mentioning two proponent protocol definition efforts:

* [A2A-AUTH] which follows [OPENAPI3-AUTH]. This means it allows for Basic, Bearer, API Keys, and [OAUTH2-RFC]. OpenAPI mentions using the [HTTP-AUTHSCHEME] registry, but there does not seem to be a definition for recent schemes such as [PRIVACYPASS-HTTP-AUTH-RFC], [CONCEALED-AUTH-RFC], or [DPOP-AUTH-RFC].

* [MCP-AUTH] uses [OAUTH2-RFC] as a resource server.

6.4. Round trip

Protocols should strive to minimise the number of round trips between a client and the issuer, and between clients and the origin.

7. Key management and discovery

7.1. Catalog

Just as there are registries to resolve IP address metadata, there are going to be registries to identify the owner of public key material. These are mentioned by [A2A-DISCOVERY] and [MCP-DISCOVERY].

The primary goal of these catalogs is to associate metadata with a public key, and the discovery of the associated metadata. They SHOULD have some sort of tamper resistance, to prevent the provider of a catalog providing incorrect information.

As an analogy, one can think of [CERTIFICATE-TRANSPARENCY-RFC], or the more recent effort in [KEY-TRANSPARENCY-ARCHITECTURE].

7.2. Submission / out-of-band

Submission is also going to happen out-of-band. This is both for a practical reason, it is simpler than setting up a catalog, and for privacy reasons, given you don’t have to expose information through a catalog.

7.3. On-path

Discovery may happen on-path, that is when a request arrives from a client to an origin. This could be considered a form of trust-on-first-use. While the level of trust is low, it could be viable for certain deployments for which knowing all agents a-priori is not viable. This could be due to their multiplicity, a frequent change in involved actors, or an origin that is willing to review new changes manually for instance.

Such discovery could be via an HTTP header containing a domain name with a well-known, a URL, a certificate, etc.

7.4. Format

There are a multitude of Key and directory formats. These include but are not limited to JWKS, CWKS, Privacy Pass, Agent Card, and HTTP Message Signatures.

8. Security Considerations

This glossary provides terminology for web bot authentication. While this document does not define or recommend specific protocols, terminology choices have direct security implications:

Impersonation Resistance Clearly defined roles are essential for preventing entities from falsely claiming identities.

Credential Replay and Theft Definitions such as Section 6.2 help describe key mechanisms that mitigate the misuse of credentials if stolen.

Key Management Section 7 is key to protocol security, and has to be considered.

In addition, protocols should consider decentralization [RFC9518] and end-user impact [RFC8890].

9. Privacy Considerations

Authentication mechanisms should minimize the collection and exposure of personal data. Techniques like selective disclosure and unlinkability help protect user privacy. Protocols should refer to [RFC6973].

Multiple protocols are also likely to be used in coordination: to identify an orgnization, then to identify the User-Agent, and possibly rate limit. It is important to consider the privacy of these layers together as well.

10. IANA Considerations

This document has no IANA actions.

11. References

11.1. 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>.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <https://www.rfc-editor.org/rfc/rfc6973>.

[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>.

[RFC8890] Nottingham, M., "The Internet is for End Users", RFC 8890, DOI 10.17487/RFC8890, August 2020, <https://www.rfc-editor.org/rfc/rfc8890>.

[RFC9518] Nottingham, M., "Centralization, Decentralization, and Internet Standards", RFC 9518, DOI 10.17487/RFC9518, December 2023, <https://www.rfc-editor.org/rfc/rfc9518>.

11.2. Informative References

[A2A-AUTH] "A2A protocol Authentication", n.d., <https://google.github.io/A2A/#/documentation?id=authentication-and-authorization>.

[A2A-DISCOVERY] "A2A protocol Agent discovery", n.d., <https://google.github.io/A2A/#/topics/agent_discovery>.

[ANONYMOUS-CREDIT-TOKENS] "Anonymous Credit Tokens", n.d., <https://samuelschlesinger.github.io/ietf-anonymous-credit-tokens/draft-schlesinger-cfrg-act.html>.

[ARC] Yun, C. and C. A. Wood, "Anonymous Rate-Limited Credentials", Work in Progress, Internet-Draft, draft-yun-cfrg-arc-00, 5 February 2025, <https://datatracker.ietf.org/doc/html/draft-yun-cfrg-arc-00>.

[BBS] Looker, T., Kalos, V., Whitehead, A., and M. Lodder, "The BBS Signature Scheme", Work in Progress, Internet-Draft, draft-irtf-cfrg-bbs-signatures-08, 3 March 2025, <https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-bbs-signatures-08>.

[BLINDRSA] Denis, F., Jacobs, F., and C. A. Wood, "RSA Blind Signatures", RFC 9474, DOI 10.17487/RFC9474, October 2023, <https://www.rfc-editor.org/rfc/rfc9474>.

[CERTIFICATE-TRANSPARENCY-RFC] Laurie, B., Langley, A., and E. Kasper, "Certificate Transparency", RFC 6962, DOI 10.17487/RFC6962, June 2013, <https://www.rfc-editor.org/rfc/rfc6962>.

[CONCEALED-AUTH-RFC] Schinazi, D., Oliver, D., and J. Hoyland, "The Concealed HTTP Authentication Scheme", RFC 9729, DOI 10.17487/RFC9729, February 2025, <https://www.rfc-editor.org/rfc/rfc9729>.

[DPOP-AUTH-RFC] Fett, D., Campbell, B., Bradley, J., Lodderstedt, T., Jones, M., and D. Waite, "OAuth 2.0 Demonstrating Proof of Possession (DPoP)", RFC 9449, DOI 10.17487/RFC9449, September 2023, <https://www.rfc-editor.org/rfc/rfc9449>.

[HTTP-AUTHSCHEME] "IANA HTTP Authentication Scheme Registry", n.d., <https://www.iana.org/assignments/http-authschemes/http-authschemes.xhtml>.

[HTTP-MESSAGE-SIGNATURE-FOR-BOTS] Meunier, T., "HTTP Message Signatures for automated traffic Architecture", Work in Progress, Internet-Draft, draft-meunier-web-bot-auth-architecture-01, 7 May 2025, <https://datatracker.ietf.org/doc/html/draft-meunier-web-bot-auth-architecture-01>.

[KEY-TRANSPARENCY-ARCHITECTURE] McMillion, B., "Key Transparency Architecture", Work in Progress, Internet-Draft, draft-ietf-keytrans-architecture-03, 25 February 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-keytrans-architecture-03>.

[LOX] "Lox: Protecting the Social Graph in Bridge Distribution", n.d., <https://petsymposium.org/2023/files/papers/issue1/popets-2023-0029.pdf>.

[MCP-AUTH] "Model Context Protocol Authorization", n.d., <https://modelcontextprotocol.io/specification/2025-03-26/basic/authorization>.

[MCP-DISCOVERY] "Model Context Protocol Registry", n.d., <https://modelcontextprotocol.io/development/roadmap#registry>.

[OAUTH-BEARER-RFC] Jones, M. and D. Hardt, "The OAuth 2.0 Authorization Framework: Bearer Token Usage", RFC 6750, DOI 10.17487/RFC6750, October 2012, <https://www.rfc-editor.org/rfc/rfc6750>.

[OAUTH2-RFC] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, DOI 10.17487/RFC6749, October 2012, <https://www.rfc-editor.org/rfc/rfc6749>.

[OPENAPI3-AUTH] "OpenAPI 3.0 Authentication", n.d., <https://swagger.io/docs/specification/v3_0/authentication/>.

[OPRF] Davidson, A., Faz-Hernandez, A., Sullivan, N., and C. A. Wood, "Oblivious Pseudorandom Functions (OPRFs) Using Prime-Order Groups", RFC 9497, DOI 10.17487/RFC9497, December 2023, <https://www.rfc-editor.org/rfc/rfc9497>.

[PARTIALLY-BLINDRSA] Amjad, G. A., Hendrickson, S., Wood, C. A., and K. W. L. Yeo, "Partially Blind RSA Signatures", Work in Progress, Internet-Draft, draft-irtf-cfrg-partially-blind-rsa-01, 1 April 2025, <https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-partially-blind-rsa-01>.

[PRIVACYPASS-ARC] Yun, C. and C. A. Wood, "Privacy Pass Issuance Protocol for Anonymous Rate-Limited Credentials", Work in Progress, Internet-Draft, draft-yun-privacypass-arc-00, 5 February 2025, <https://datatracker.ietf.org/doc/html/draft-yun-privacypass-arc-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>.

[PRIVACYPASS-HTTP-AUTH-RFC]
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>.

[PRIVACYPASS-IN-TLS]
Pauly, T. and S. Hendrickson, "Including Privacy Pass Tokens in TLS Handshakes", Work in Progress, Internet-Draft, draft-pauly-privacypass-for-tls-00, 3 March 2025, <https://datatracker.ietf.org/doc/html/draft-pauly-privacypass-for-tls-00>.

[PRIVACYPASS-KAGI]
"Introducing Privacy Pass authentication for Kagi Search", n.d., <https://blog.kagi.com/kagi-privacy-pass>.

[PRIVACYPASS-PROTOCOL]
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>.

[PRIVACYPASS-PUBLIC-METADATA]
Hendrickson, S. and C. A. Wood, "Privacy Pass Issuance Protocols with Public Metadata", Work in Progress, Internet-Draft, draft-ietf-privacypass-public-metadata-issuance-02, 27 May 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-privacypass-public-metadata-issuance-02>.

[PRIVATE-ACCESS-TOKEN]
"Challenge: Private Access Tokens", n.d., <https://developer.apple.com/news/?id=huqjyh7k>.

[PRIVATE-PROOF-API]
"Explainer by Googlers Private Proof API", n.d., <https://explainers-by-googlers.github.io/private-proof/>.

[PRIVATE-STATE-TOKEN]
"W3C Private State Token API", n.d., <https://wicg.github.io/trust-token-api/>.

[REQ-MTLS] Hoyland, J., "TLS Flag - Request mTLS", Work in Progress, Internet-Draft, draft-jhoyla-req-mtls-flag-02, 28 February 2025, <https://datatracker.ietf.org/doc/html/draft-jhoyla-req-mtls-flag-02>.

[VERIFIABLE-CREDENTIALS] "Verifiable Credentials Data Model v1.1", n.d., <https://www.w3.org/TR/2022/REC-vc-data-model-20220303/>.

Acknowledgments

TODO acknowledge.

Changelog

v01

* Improve wording from on-path submission (thanks to Christopher Patton)

* Add CFRG and Privacy Pass drafts within public/private presentation (thanks to Christopher Patton)

* Add web crawler example along with AI agent

* Add anonymous credit tokens draft (thanks to Sam Schlesinger)

v00

* Initial draft

* Overall ecosystem architecture

* Terminology

* Rough deployment considerations

* Reference multiple agent protocols

Author’s Address

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