Workgroup: Network Working Group
Internet-Draft:
draft-ietf-privacypass-architecture-11
Published: 6 March 2023
Intended Status: Informational
Expires: 7 September 2023

Authors: A. Davidson J. Iyengar C. A. Wood
LIP Fastly Cloudflare

The Privacy Pass Architecture

Abstract

This document specifies the Privacy Pass architecture and requirements for its constituent protocols used for constructing privacy-preserving authentication mechanisms. It provides recommendations on how the architecture should be deployed to ensure the privacy of clients and the security of all participating entities.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Privacy Pass is an architecture for authorization based on privacy-preserving authentication mechanisms. Typical approaches for authorizing clients, such as through the use of long-term state (cookies), are not privacy-friendly since they allow servers to track clients across sessions and interactions. Privacy Pass takes a different approach: instead of presenting linkable state-carrying information to servers, e.g., a cookie indicating whether or not the client is an authorized user or has completed some prior challenge, clients present unlinkable proofs that attest to this information. These proofs, or tokens, are private in the sense that a given token cannot be linked to the protocol interaction where that token was initially issued.

At a high level, the Privacy Pass architecture consists of two protocols: redemption and issuance. The redemption protocol, described in [AUTHSCHEME], runs between Clients and Origins (servers). It allows Origins to challenge Clients to present tokens for authorization. Depending on the type of token, e.g., whether or not it can be cached, the Client either presents a previously obtained token or invokes an issuance protocol, such as [ISSUANCE], to acquire a token to present as authorization.

This document describes requirements for both redemption and issuance protocols and how they interact. It also provides recommendations on how the architecture should be deployed to ensure the privacy of clients and the security of all participating entities.

2. 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.

The following terms are used throughout this document:

Client: An entity that seeks authorization to an Origin.

Origin: An entity that redeems tokens presented by Clients.

Issuer: An entity that issues tokens to Clients for properties attested to by the Attester.

Attester: An entity that attests to properties of Clients for the purposes of token issuance.

Attestation procedure: The procedure by which an Attester determines whether or not a Client is trusted with a specific set of properties for token issuance.

3. Architecture

The Privacy Pass architecture consists of four logical entities -- Client, Origin, Issuer, and Attester -- that work in concert for token redemption and issuance. This section presents an overview of Privacy Pass, a high-level description of the threat model and privacy goals of the architecture, and the goals and requirements of the redemption and issuance protocols.

3.1. Overview

The typical interaction flow for Privacy Pass uses the following steps:

1. A Client interacts with an Origin by sending a request or otherwise interacting with the Origin in a way that triggers a response containing a token challenge. The token challenge indicates a specific Issuer to use.

2. If the Client already has a token available that satisfies the token challenge, e.g., because the Client has a cache of previously issued tokens, it can skip to step 6 and redeem its token; see Section 7.1 for security considerations of cached tokens.

3. If the Client does not have a token available and decides it wants to obtain one (or more) bound to the token challenge, it then invokes the issuance protocol. As a prerequisite to the issuance protocol, the Client runs the deployment specific attestation process that is required for the designated Issuer. Client attestation can be done via proof of solving a CAPTCHA, checking device or hardware attestation validity, etc; see Section 3.4.1 for more details.

4. If the attestation process completes successfully, the client creates a Token Request to send to the designated Issuer (generally via the Attester, though it is not required to be sent through the Attester). The Attester and Issuer might be functions on the same server, depending on the deployment model (see Section 4). Depending on the attestation process, it is possible for attestation to run alongside the issuance protocol, e.g., where Clients send necessary attestation information to the Attester along with their Token Request. If the attestation process fails, the Client receives an error and issuance aborts without a token.

5. The Issuer generates a Token Response based on the Token Request, which is returned to the Client (generally via the Attester). Upon receiving the Token Response, the Client computes a token from the token challenge and Token Response. This token can be validated by anyone with the per-Issuer key, but cannot be linked to the content of the Token Request or Token Response.

6. If the Client has a token, it includes it in a subsequent request to the Origin, as authorization. This token is sent

only once. The Origin validates that the token was generated by the expected Issuer and has not already been redeemed for the corresponding token challenge. If the Client does not have a token, perhaps because issuance failed, the client does not reply to the Origin's challenge with a new request.

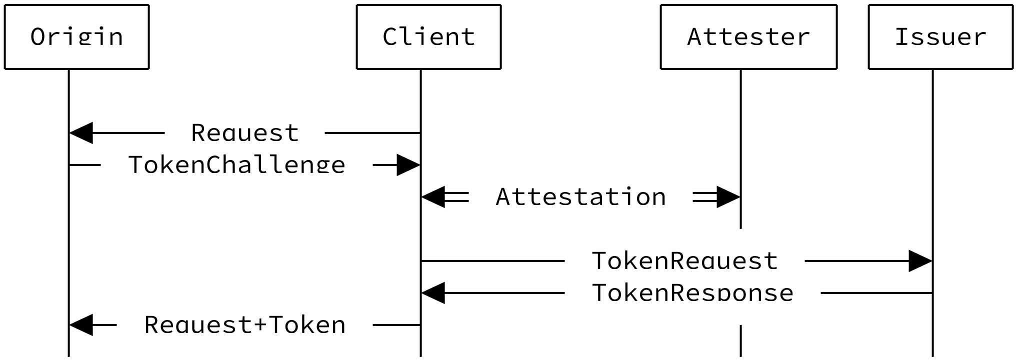

Figure 1: Privacy pass redemption and issuance protocol interaction

3.2. Privacy Goals and Threat Model

The end-to-end flow for Privacy Pass described in Section 3.1 involves three different types of contexts:

Redemption context: The interactions and set of information shared between the Client and Origin, i.e., the information that is provided or otherwise available to the Origin during redemption that might be used to identify a Client and construct a token challenge. This context includes all information associated with redemption, such as the timestamp of the event, Client visible information (including the IP address), and the Origin name.

Issuance context: The interactions and set of information shared between the Client, Attester, and Issuer, i.e., the information that is provided or otherwise available to Attester and Issuer during issuance that might be used to identify a Client. This context includes all information associated with issuance, such as the timestamp of the event, any Client visible information (including the IP address), and the Origin name (if revealed during issuance). This does not include the token challenge in its entirety, as that is kept secret from the Issuer during the issuance protocol.

Attestation context: The interactions and set of information shared between the Client and Attester only, for the purposes of attesting the vailidity of the Client, that is provided or otherwise available during attestation that might be used to identify the Client. This context includes all information associated with attestation, such as the timestamp of the event and any Client visible information, including information needed for the attestation procedure to complete.

The privacy goals of Privacy Pass assume a threat model in which Origins trust specific Issuers to produce tokens, and Issuers in turn trust one or more Attesters to correctly run the attestation procedure with Clients. This arrangement ensures that tokens which validate for a given Issuer were only issued to a Client that successfully completed attestation with an Attester that the Issuer trusts. Moreover, this arrangement means that if an Origin accepts tokens issued by an Issuer that trusts multiple Attesters, then a Client can use any one of these Attesters to issue and redeem tokens for the Origin.

The mechanisms for establishing trust between each entity in this arrangement are deployment specific. For example, in settings where Clients interact with Issuers through an Attester, Attesters and Issuers might use mutually authenticated TLS to authenticate one another. In settings where Clients do not communicate with Issuers through an Attester, the Attesters might convey this trust via a digital signature over that Issuers can verify.

Clients explicitly trust Attesters to perform attestation correctly and in a way that does not violate their privacy. However, Clients assume Issuers and Origins are malicious.

Given this threat model, the privacy goals of Privacy Pass are oriented around unlinkability based on redemption, issuance, and attestation contexts, as described below.

1. Origin-Client unlinkability. This means that given two redemption contexts, the Origin cannot determine if both redemption contexts correspond to the same Client or two different Clients. Informally, this means that a Client in a redemption context is indistinguishable from any other Client that might use the same redemption context. The set of Clients that share the same redemption context is referred to as a redemption anonymity set.

2. Issuer-Client unlinkability. This is similar to Origin-Client unlinkability in that a Client in an issuance context is indistinguishable from any other Client that might use the same issuance context. The set of Clients that share the same issuance context is referred to as an issuance anonymity set.

3. Attester-Origin unlinkability. This is similar to Origin-Client and Issuer-Client unlinkability. It means that given two attestation contexts, the Attester cannot determine if both contexts correspond to the same Origin or two different Origins. The set of Clients that share the same attestation context is referred to as an attestation anonymity set.

By ensuring that different contexts cannot be linked in this way, only the Client is able to correlate information that might be used to identify them with activity on the Origin. The Attester, Issuer, and Origin only receive the information necessary to perform their respective functions.

The manner in which Origin-Client, Issuer-Client, and Attester-Origin unlinkability are achieved depends on the deployment model, type of attestation, and issuance protocol details. For example, as discussed in Section 4, failure to use a privacy-enhancing proxy system such as Tor [DMS2004] when interacting with Attesters, Issuers, or Origins allows the set of possible Clients to be partitioned by the Client's IP address, and can therefore lead to unlinkability violations. Similarly, malicious Origins may attempt to link two redemption contexts together by using Client-specific Issuer public keys. See Section 4 and Section 6 for more information.

The remainder of this section describes the functional properties and security requirements of the redemption and issuance protocols in more detail. Section 3.5 describes how information flows between Issuer, Origin, Client, and Attester through these protocols.

3.3. Redemption Protocol

The Privacy Pass redemption protocol, described in [AUTHSCHEME], is an authorization protocol wherein Clients present tokens to Origins for authorization. Normally, redemption follows a challenge-response flow, wherein the Origin challenges Clients for a token with a TokenChallenge ([AUTHSCHEME], Section 2.1) and, if possible, Clients present a valid Token ([AUTHSCHEME], Section 2.2) in response. This interaction is shown below.

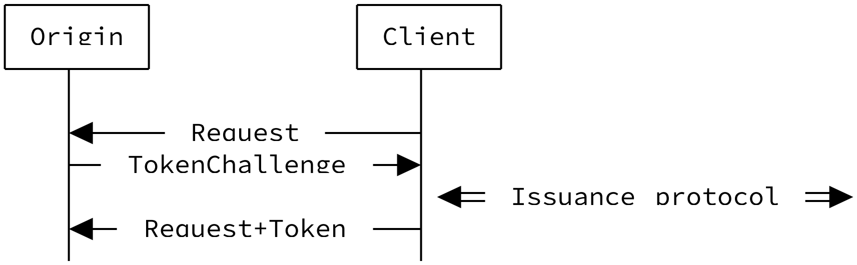

Figure 2: Challenge-response redemption protocol interaction

Alternatively, when configured to do so, Clients may opportunistically present Token values to Origins without a corresponding TokenChallenge.

The structure and semantics of the TokenChallenge and Token messages depend on the issuance protocol and token type being used; see [AUTHSCHEME] for more information.

The challenge provides the client with the information necessary to obtain tokens that the server might subsequently accept in the redemption context. There are a number of ways in which the token may vary based on this challenge, including:

- Issuance protocol. The challenge identifies the type of issuance protocol required for producing the token. Different issuance protocols have different security properties, e.g., some issuance protocols may produce tokens that are publicly verifiable, whereas others may not have this property.

- Issuer identity. Token challenges identify which Issuers are trusted for a given issuance protocol. As described in Section 3.2, the choice of Issuer determines the type of Attesters and attestation procedures possible for a token from the specified Issuer, but the Client does not learn exactly which Attester was used for a given token issuance event.

- Redemption context. Challenges can be bound to a given redemption context, which influences a client's ability to pre-fetch and cache tokens. For example, an empty redemption context always allows tokens to be issued and redeemed non-interactively, whereas a fresh and random redemption context means that the redeemed token must be issued only after the client receives the challenge. See Section 2.1.1 of [AUTHSCHEME] for more details.

- Per-Origin or cross-Origin. Challenges can be constrained to the Origin for which the challenge originated (referred to as per-Origin tokens), or can be used across multiple Origins (referred to as cross-Origin tokens). The set of Origins for which a cross-Origin token is applicable is referred to as the cross-Origin set.

Origins that admit cross-Origin tokens bear some risk of allowing tokens issued for one Origin to be spent in an interaction with another Origin. In particular, depending on the use case, Origins may need to maintain state to track redeemed tokens. For example, Origins that accept cross-Origin tokens across shared redemption contexts SHOULD track which tokens have been redeemed already in those redemption contexts, since these tokens can be issued and then spent multiple times in response to any such challenge. See Section 2.1.1 of [AUTHSCHEME] for discussion.

How Clients respond to token challenges can have privacy implications. For example, if an Origin allows the Client to choose an Issuer, then the choice of Issuer can reveal information about the Client used to partition anonymity sets; see Section 6.2 for more information about these privacy considerations.

3.4. Issuance Protocol

The Privacy Pass issuance protocol, described in [ISSUANCE], is a two-message protocol that takes as input a TokenChallenge from the redemption protocol ([AUTHSCHEME], Section 2.1) and produces a Token ([AUTHSCHEME], Section 2.2), as shown in Figure 1.

The structure and semantics of the TokenRequest and TokenResponse messages depend on the issuance protocol and token type being used; see [ISSUANCE] for more information.

Clients interact with the Attester and Issuer to produce a token in response to a challenge. The context in which an Attester vouches for a Client during issuance is referred to as the attestation context. This context includes all information associated with the issuance event, such as the timestamp of the event and Client visible information, including the IP address or other information specific to the type of attestation done.

Each issuance protocol may be different, e.g., in the number and types of participants, underlying cryptographic constructions used when issuing tokens, and even privacy properties.

Clients initiate the issuance protocol using the token challenge, a randomly generated nonce, and public key for the Issuer, all of which are the Client's private input to the protocol and ultimately bound to an output Token; see Section 2.2 of [AUTHSCHEME] for details. Future specifications may change or extend the Client's input to the issuance protocol to produce Tokens with a different structure.

The issuance protocol itself can be any interactive protocol between Client, Issuer, or other parties that produces a valid token bound to the Client's private input, subject to the following security requirements.

1. Unconditional input secrecy. The issuance protocol MUST NOT reveal anything about the Client's private input, including the challenge and nonce, to the Attester or Issuer, regardless of the hardness assumptions of the underlying cryptographic protocol(s). This property is sometimes also referred to as blindness.

2. One-more forgery security. The issuance protocol MUST NOT allow malicious Clients or Attesters (acting as Clients) to forge tokens offline or otherwise without interacting with the Issuer directly.

3. Concurrent security. The issuance protocol MUST be safe to run concurrently with arbitrarily many Clients, Attesters and Issuers.

See Section 3.4.4 for requirements on new issuance protocol variants and related extensions.

In the sections below, we describe the Attester and Issuer roles in more detail.

3.4.1. Attester Role

In Privacy Pass, attestation is the process by which an Attester bears witness to, confirms, or authenticates a Client so as to verify properties about the Client that are required for Issuance. Issuers trust Attesters to perform attestation correctly.

[RFC9334] describes an architecture for attestation procedures. Using that architecture as a conceptual basis, Clients are RATS attesters that produce attestation evidence, and Attesters are RATS verifiers that appraise the validity of attestation evidence.

The type of attestation procedure is a deployment-specific option and outside the scope of the issuance protocol. Example attestation procedures are below.

*Solving a CAPTCHA. Clients that solve CAPTCHA challenges can be attested to have this capability for the purpose of being ruled out as a bot or otherwise automated Client.

*Presenting evidence of Client device validity. Some Clients run on trusted hardware that are capable of producing device-level attestation evidence.

*Proving properties about Client state. Clients can be associated with state and the Attester can verify this state. Examples of state include the Client's geographic region and whether the Client has a valid application-layer account.

Attesters may support different types of attestation procedures.

In general, each attestation procedure has different security properties. For example, attesting to having a valid account is different from attesting to running on trusted hardware. In general, minimizing the set of supported attestation procedures helps minimize the amount of information leaked through a token.

The role of the Attester in the issuance protocol and its impact on privacy depends on the type of attestation procedure, issuance protocol, deployment model. For instance, requiring a conjunction of attestation procedures could decrease the overall anonymity set size. As an example, the number of Clients that have solved a CAPTCHA in the past day, that have a valid account, and that are running on a trusted device is less than the number of Clients that have solved a CAPTCHA in the past day. Attesters SHOULD NOT be based on attestation procedures that result in small anonymity sets.

Depending on the issuance protocol, the Issuer might learn information about the Origin. To ensure Issuer-Client unlinkability, the Issuer should be unable to link that information to a specific Client. For such issuance protocols where the Attester has access to Client-specific information, such as is the case for attestation procedures that involve Client-specific information (such as application-layer account information) or for deployment models where the Attester learns Client-specific information (such as Client IP addresses), Clients trust the Attester to not share any Client-specific information with the Issuer. In deployments where the Attester does not learn Client-specific information, the Client does not need to explicitly trust the Attester in this regard.

Issuers trust Attesters to correctly and reliably perform attestation. However, certain types of attestation can vary in value over time, e.g., if the attestation procedure is compromised. Broken attestation procedures are considered exceptional events and require configuration changes to address the underlying cause. For example, if attestation is compromised because of a zero-day exploit on compliant devices, then the corresponding attestation procedure should be untrusted until the exploit is patched. Addressing changes in attestation quality is therefore a deployment-specific task. In Split Attester and Issuer deployments (see Section 4.4), Issuers can choose to remove compromised Attesters from their trusted set until the compromise is patched.

From the perspective of an Origin, tokens produced by an Issuer with at least one compromised Attester cannot be trusted assuming the Origin does not know which attestation procedure was used for issuance. This is because the Origin cannot distinguish between tokens that were issued via compromised Attesters and tokens that were issued via uncompromised Attesters absent some distinguishing information in the tokens themselves or from the Issuer. As a result, until the attestation procedure is fixed, the Issuer cannot be trusted by Origins. Moreover, as a consequence, any tokens issued by an Issuer with a compromised attester may no longer be trusted by Origins, even if those tokens were issued to Clients interacting with an uncompromised Attester.

3.4.2. Issuer Role

In Privacy Pass, the Issuer is responsible for completing the issuance protocol for Clients that complete attestation through a trusted Attester. As described in Section 3.4.1, Issuers explicitly trust Attesters to correctly and reliably perform attestation. Origins explicitly trust Issuers to only issue tokens from trusted Attesters. Clients do not explicitly trust Issuers.

Depending on the deployment model case, issuance may require some form of Client anonymization service, similar to an IP-hiding proxy, so that Issuers cannot learn information about Clients. This can be provided by an explicit participant in the issuance protocol, or it can be provided via external means, such as through the use of an IP-hiding proxy service like Tor. In general, Clients SHOULD minimize or remove identifying information where possible when invoking the issuance protocol.

Issuers are uniquely identifiable by all Clients with a consistent identifier. In a web context, this identifier might be the Issuer host name. Issuers maintain one or more configurations, including issuance key pairs, for use in the issuance protocol. Issuers can rotate these configurations as needed to mitigate risk of compromise; see Section 6.2 for more considerations around configuration rotation. The Issuer public key for each active configuration is made available to Origins and Clients for use in the issuance and redemption protocols.

3.4.3. Issuance Metadata

Certain instantiations of the issuance protocol may permit public or private metadata to be cryptographically bound to a token. As an example, one trivial way to include public metadata is to assign a unique Issuer public key for each value of metadata, such that N keys yields $\log_2(N)$ bits of metadata. Metadata may be public or private.

Public metadata is that which clients can observe as part of the token issuance flow. Public metadata can either be transparent or opaque. For example, transparent public metadata is a value that the client either generates itself, or the Issuer provides during the issuance flow and the client can check for correctness. Opaque public metadata is metadata the client can see but cannot check for correctness. As an example, the opaque public metadata might be a "fraud detection signal", computed on behalf of the Issuer, during token issuance. In normal circumstances, Clients cannot determine if this value is correct or otherwise a tracking vector.

Private metadata is that which Clients cannot observe as part of the token issuance flow. Such instantiations can be built on the Private Metadata Bit construction from Kreuter et al. [KLOR20] or the attribute-based VOPRF from Huang et al. [HIJK21].

Metadata can be arbitrarily long or bounded in length. The amount of permitted metadata may be determined by application or by the underlying cryptographic protocol. The total amount of metadata bits included in a token is the sum of public and private metadata bits. Every bit of metadata can be used to partition the Client issuance or redemption anonymity sets; see Section 6.1 for more information.

3.4.4. Issuance Protocol Extensibility

The Privacy Pass architecture and ecosystem are both intended to be receptive to extensions that expand the current set of functionalities through new issuance protocols. Each issuance protocol MUST include a detailed analysis of the privacy impacts of the extension, why these impacts are justified, and guidelines on how to deploy the protocol to minimize any privacy impacts. Any extension to the Privacy Pass protocol MUST adhere to the guidelines specified in Section 3.4.2 for managing Issuer public key data.

3.5. Information Flow

The end-to-end process of redemption and issuance protocols involves information flowing between Issuer, Origin, Client, and Attester. That information can have implications on the privacy goals that Privacy Pass aims to provide as outlined in Section 3.2. In this section, we describe the flow of information between each party. How this information affects the privacy goals in particular deployment models is further discussed in Section 4.

3.5.1. Token Challenge Flow

To use Privacy Pass, Origins choose an Issuer from which they are willing to accept tokens. Origins then construct a token challenge using this specified Issuer and information from the redemption context it shares with the Client. This token challenge is then delivered to a Client. The token challenge conveys information about the Issuer and the redemption context, such as whether the Origin desires a per-Origin or cross-Origin token. Any entity that sees the token challenge might learn things about the Client as known to the Origin. This is why input secrecy is a requirement for issuance protocols, as it ensures that the challenge is not directly available to the Issuer.

3.5.2. Attestation Flow

Clients interact with the Attester to prove that they meet some required set of properties. In doing so, Clients contribute information to the attestation context, which might include sensitive information such as application-layer identities, IP addresses, and so on. Clients can choose whether or not to contribute this information based on local policy or preference.

3.5.3. Issuance Flow

Clients use the issuance protocol to produce a token bound to a token challenge. In doing so, there are several ways in which the issuance protocol contributes information to the attestation or issuance contexts. For example, a token request may contribute information to the attestation or issuance contexts as described below.

- Issuance protocol. The type of issuance protocol can contribute information about the Issuer's capabilities to the attestation or issuance contexts, as well as the capabilities of a given Client. For example, if a Client is presented with multiple issuance protocol options, then the choice of which issuance protocol to use can contribute information about the Client's capabilities.

- Issuer configuration. Information about the Issuer configuration, such as its identity or the public key used to validate tokens it creates, can be revealed during issuance and contribute to the attestation or issuance contexts.

- Attestation information. The issuance protocol can contribute information to the attestation or issuance contexts based on what attestation procedure the Issuer uses to trust a token request. In particular, a token request that is validated by a given Attester means that the Client which generated the token request must be capable of the completing the designated attestation procedure.

- Origin information. The issuance protocol can contribute information about the Origin that challenged the Client in Section 3.5.1. In particular, a token request designated for a specific Issuer might imply that the resulting token is for an

Origin that trusts the specified Issuer. However, this is not always true, as some token requests can correspond to cross-Origin tokens, i.e., they are tokens that would be accepted at any Origin that accepts the cross-Origin token.

Moreover, a token response may contribute information to the issuance attestation or contexts as described below.

- Origin information. The issuance protocol can contribute information about the Origin in how it responds to a token request. For example, if an Issuer learns the Origin during issuance and is also configured to respond in some way on the basis of that information, and the Client interacts with the Issuer transitively through the Attester, that response can reveal information to the Attester.

- Token. The token produced by the issuance protocol can contain information from the issuance context. In particular, depending on the issuance protocol, tokens can contain public or private metadata, and Issuers can choose that metadata on the basis of information in the issuance context.

Exceptional cases in the issuance protocol, such as when either the Attester or Issuer aborts the protocol, can contribute information to the attestation or issuance contexts. The extent to which information in this context harms the Issuer-Client or Attester-Origin unlinkability goals in Section 3.2 depends on deployment model; see Section 4. Clients can choose whether or not to contribute information to these contexts based on local policy or preference.

3.5.4. Token Redemption Flow

Clients send tokens to Origins during the redemption protocol. Any information that is added to the token during issuance can therefore be sent to the Origin. Information can either be explicitly passed in a token, or it can be implicit in the way the Client responds to a token challenge. For example, if a Client fails to complete issuance, and consequently fails to redeem a token in response to a token challenge, this can reveal information to the Origin that it might not otherwise have access to. However, an Origin cannot necessarily distinguish between a Client that fails to complete issuance and one that ignores the token challenge altogether.

4. Deployment Configurations

The Origin, Attester, and Issuer portrayed in Figure 1 can be instantiated and deployed in a number of ways. The deployment model directly influences the manner in which attestation, issuance, and redemption contexts are separated to achieve Origin-Client, Issuer-Client, and Attester-Origin unlinkability.

This section covers some expected deployment models and their corresponding security and privacy considerations. Each deployment model is described in terms of the trust relationships and communication patterns between Client, Attester, Issuer, and Origin.

The discussion below assumes non-collusion between entities that have access to the attestation, issuance, and redemption contexts, as collusion between such entities would enable linking of these contexts and may lead to unlinkability violations. Generally, this means that entities operated by separate parties do not collude. Mechanisms for enforcing non-collusion are out of scope for this architecture.

4.1. Shared Origin, Attester, Issuer

In this model, the Origin, Attester, and Issuer are all operated by the same entity, as shown in Figure 3.

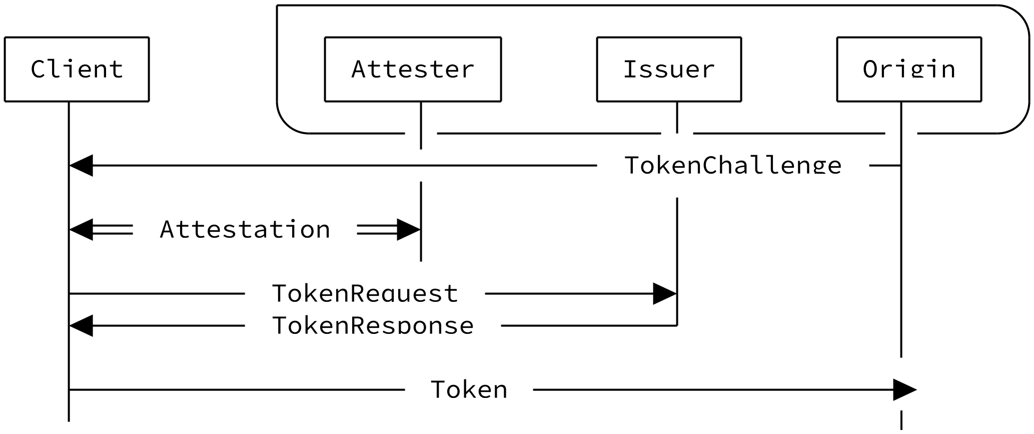

Figure 3: Shared Deployment Model

This model represents the initial deployment of Privacy Pass, as described in [PrivacyPassCloudflare]. In this model, the Attester, Issuer, and Origin share the attestation, issuance, and redemption contexts. As a result, attestation mechanisms that can uniquely identify a Client, e.g., requiring that Clients authenticate with some type of application-layer account, are not appropriate, as they could lead to unlinkability violations.

Origin-Client, Issuer-Client, and Attester-Origin unlinkability requires that issuance and redemption events be separated over time, such as through the use of tokens that correspond to token challenges with an empty redemption context (see Section 3.3), or be separated over space, such as through the use of an anonymizing proxy when connecting to the Origin.

4.2. Joint Attester and Issuer

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. Clients interact with the joint Attester and Issuer for attestation and issuance. This arrangement is shown in Figure 4.

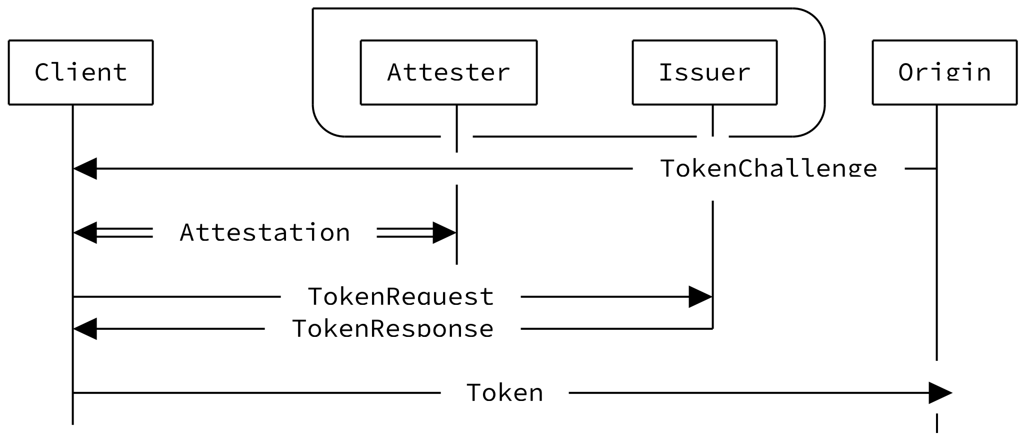

Figure 4: Joint Attester and Issuer Deployment Model

This model is useful if an Origin wants to offload attestation and issuance to a trusted entity. In this model, the Attester and Issuer share an attestation and issuance context for the Client, which is separate from the Origin's redemption context.

For certain types of issuance protocols, this model achieves Origin-Client, Issuer-Client, and Attester-Origin unlinkability. However, issuance protocols that require the Issuer to learn information about the Origin, such as that which is described in [RATE-LIMITED], are not appropriate since they could lead to Attester-Origin unlinkability violations through the Origin name.

4.3. Joint Origin and Issuer

In this model, the Origin and Issuer are operated by the same entity, separate from the Attester, as shown in the figure below. The Issuer accepts token requests that come from trusted Attesters. Since the Attester and Issuer are separate entities, this model requires some mechanism by which Issuers establish trust in the Attester (as described in Section 3.2). For example, in settings where the Attester is a Client-trusted service that directly communicates with the Issuer, one way to establish this trust is via mutually-authenticated TLS. However, alternative authentication mechanisms are possible. This arrangement is shown in Figure 5.

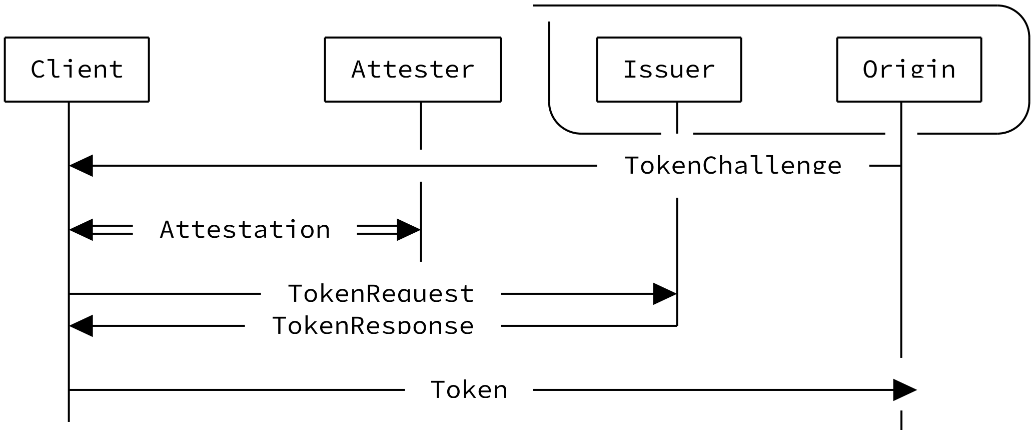

Figure 5: Joint Origin and Issuer Deployment Model

This model is useful for Origins that require Client-identifying attestation, e.g., through the use of application-layer account information, but do not otherwise want to learn information about individual Clients beyond what is observed during the token redemption, such as Client IP addresses.

In this model, attestation contexts are separate from issuer and redemption contexts. As a result, any type of attestation is suitable in this model.

Moreover, any type of token challenge is suitable assuming there is more than one Origin involved, since no single party will have access to the identifying Client information and unique Origin information. Issuers that produce tokens for a single Origin are not suitable in this model since an Attester can infer the Origin from a token request, as described in Section 3.5.3. However, since the issuance protocol provides input secrecy, the Attester does not learn details about the corresponding token challenge, such as whether the token challenge is per-Origin or cross-Origin.

4.4. Split Origin, Attester, Issuer

In this model, the Origin, Attester, and Issuer are all operated by different entities, as shown in the figure below. As with the joint Origin and Issuer model, the Issuer accepts token requests that come from trusted Attesters, and the details of that trust establishment depend on the issuance protocol and relationship between Attester and Issuer; see Section 3.2. This arrangement is shown in Figure 1.

This is the most general deployment model, and is necessary for some types of issuance protocols where the Attester plays a role in token issuance; see [RATE-LIMITED] for one such type of issuance protocol.

In this model, the Attester, Issuer, and Origin have a separate view of the Client: the Attester sees potentially sensitive Client identifying information, such as account identifiers or IP addresses, the Issuer sees only the information necessary for issuance, and the Origin sees token challenges, corresponding tokens, and Client source information, such as their IP address. As a result, attestation, issuance, and redemption contexts are separate, and therefore any type of token challenge is suitable in this model as long as there is more than a single Origin.

As in the Joint Origin and Issuer model in Section 4.3, and as described in Section 3.5.3, if the Issuer produces tokens for a single Origin, then per-Origin tokens are not appropriate since the Attester can infer the Origin from a token request.

5. Centralization Considerations

A consequence of limiting the number of participants (Attesters or Issuers) in Privacy Pass deployments for meaningful privacy is that it forces concentrated centralization amongst those participants. [CENTRALIZATION] discusses several ways in which this might be mitigated. For example, a multi-stakeholder governance model could be established to determine what candidate participants are fit to operate as participants in a Privacy Pass deployment. This is precisely the system used to control the Web's trust model.

Alternatively, Privacy Pass deployments might mitigate this problem through implementation. For example, rather than centralize the role of attestation in one or few entities, attestation could be a distributed function performed by a quorum of many parties, provided that neither Issuers nor Origins learn which Attester implementations were chosen. As a result, Clients could have more opportunities to switch between attestation participants.

6. Privacy Considerations

The previous section discusses the impact of deployment details on Origin-Client, Issuer-Client, and Attester-Origin unlinkability. The value these properties affords to end users depends on the size of anonymity sets in which Clients or Origins are unlinkable. For example, consider two different deployments, one wherein there exists a redemption anonymity set of size two and another wherein there redemption anonymity set of size 2^{32}. Although Origin-Client unlinkabiity guarantees that the Origin cannot link any two requests to the same Client based on these contexts, respectively, the probability of determining the "true" Client is higher the smaller these sets become.

In practice, there are a number of ways in which the size of anonymity sets may be reduced or partitioned, though they all center around the concept of consistency. In particular, by definition, all Clients in an anonymity set share a consistent view of information needed to run the issuance and redemption protocols. An example type of information needed to run these protocols is the Issuer public key. When two Clients have inconsistent information, these Clients effectively have different redemption contexts and therefore belong in different anonymity sets.

The following sections discuss issues that can influence anonymity set size. For each issue, we discuss mitigations or safeguards to protect against the underlying problem.

6.1. Partitioning by Issuance Metadata

Any metadata bits of information can be used to further segment the size of the Client's anonymity set. Any Issuer that wanted to track a single Client could add a single metadata bit to Client tokens. For the tracked Client it would set the bit to 1, and 0 otherwise. Adding additional bits provides an exponential increase in tracking granularity similarly to introducing more Issuers (though with more potential targeting).

For this reason, the amount of metadata used by an Issuer in creating redemption tokens must be taken into account -- together with the bits of information that Issuers may learn about Clients otherwise. Since this metadata may be useful for practical deployments of Privacy Pass, Issuers must balance this against the reduction in Client privacy.

In general, limiting the amount of metadata permitted helps limit the extent to which metadata can uniquely identify individual Clients. Clients SHOULD bound the number of possible metadata values in practice. Most token types do not admit any metadata, so this bound is implicitly enforced. Moreover, Privacy Pass deployments SHOULD NOT use metadata unless its value has been assessed and weighed against the corresponding reduction in Client privacy.

6.2. Partitioning by Issuance Consistency

Anonymity sets can be partitioned by information used for the issuance protocol, including: metadata, Issuer configuration (keys), and Issuer selection.

Any issuance metadata bits of information can be used to partition the Client anonymity set. For example, any Issuer that wanted to track a single Client could add a single metadata bit to Client tokens. For the tracked Client it would set the bit to 1, and 0 otherwise. Adding additional bits provides an exponential increase in tracking granularity similarly to introducing more Issuers (though with more potential targeting).

The number of active Issuer configurations also contributes to anonymity set partitioning. In particular, when an Issuer updates their configuration and the corresponding key pair, any Client that invokes the issuance protocol with this configuration becomes be part of a set of Clients which also ran the issuance protocol using the same configuration. Issuer configuration updates, e.g., due to key rotation, are an important part of hedging against long-term private key compromise. In general, key rotations represent a trade-off between Client privacy and Issuer security. Therefore, it is important that key rotations occur on a regular cycle to reduce the harm of an Issuer key compromise.

Lastly, if Clients are willing to issue and redeem tokens from a large number of Issuers for a specific Origin, and that Origin accepts tokens from all Issuers, segregation can occur. In particular, if a Client obtains tokens from many Issuers and an Origin later challenges that Client for a token from each Issuer, the Origin can learn information about the Client. Each per-Issuer token that a Client holds essentially corresponds to a bit of information about the Client that Origin learns. Therefore, there is an exponential loss in privacy relative to the number of Issuers.

The fundamental problem here is that the number of possible issuance configurations, including the keys in use and the Issuer identities themselves, can partition the Client anonymity set. To mitigate this problem, Clients SHOULD bound the number of active issuance configurations per Origin as well as across Origins. Moreover, Clients SHOULD employ some form of consistency mechanism to ensure that they receive the same configuration information and are not being actively partitioned into smaller anonymity sets. See [CONSISTENCY] for possible consistency mechanisms. Depending on the deployment, the Attester might assist the Client in applying these consistency checks across clients. Failure to apply a consistency check can allow Client-specific keys to violate Origin-Client unlinkability.

6.3. Partitioning by Side-Channels

Side-channel attacks, such as those based on timing correlation, could be used to reduce anonymity set size. In particular, for interactive tokens that are bound to a Client-specific redemption context, the anonymity set of Clients during the issuance protocol consists of those Clients that started issuance between the time of the Origin's challenge and the corresponding token redemption. Depending on the number of Clients using a particular Issuer during that time window, the set can be small. Appliations should take such side channels into consideration before choosing a particular deployment model and type of token challenge and redemption context.

7. Security Considerations

This document describes security and privacy requirements for the Privacy Pass redemption and issuance protocols. It also describes deployment models and privacy considerations for using Privacy Pass within those models. Ensuring Client privacy -- separation of attestation and redemption contexts -- requires active work on behalf of the Client, especially in the presence of malicious Issuers and Origins. Implementing mitigations discused in Section 4 and Section 6 is therefore necessary to ensure that Privacy Pass offers meaningful privacy improvements to end-users.

7.1. Token Caching

Depending on the Origin's token challenge, Clients can request and cache more than one token using an issuance protocol. Cached tokens help improve privacy by separating the time of token issuance from the time of token redemption, and also allow Clients to reduce the overhead of receiving new tokens via the issuance protocol.

As a consequence, Origins that send token challenges which are compatible with cached tokens need to take precautions to ensure that tokens are not replayed. This is typically done via keeping track of tokens that are redeemed for the period of time in which cached tokens would be accepted for particular challenges.

Moreover, since tokens are not intrinsically bound to Clients, it is possible for malicious Clients to collude and share tokens in a so-called "hoarding attack." As an example of this attack, many distributed Clients could obtain cacheable tokens and them share them with a single Client to redeem in a way that would violate an Origin's attempt to limit tokens to any one particular Client. Depending on the deployment model, it can be possible to detect these types of attacks by comparing issuance and redemption contexts; for example, this is possible in the Joint Origin and Issuer model.

8. References

8.1. 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>.

[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>.

8.2. Informative 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-08, 30 January 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-privacypass-auth-scheme-08>.

[CENTRALIZATION] Nottingham, M., "Internet Centralization: What Can Standards Do?", Work in Progress, Internet-Draft, draft-nottingham-avoiding-internet-centralization-09, 17 February 2023, <https://datatracker.ietf.org/doc/html/draft-nottingham-avoiding-internet-centralization-09>.

[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-00, 24 October 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-privacypass-key-consistency-00>.

[DMS2004] Dingledine, R., Mathewson, N., and P. Syverson, "Tor: The Second-Generation Onion Router", August 2004, <https://svn.torproject.org/svn/projects/design-paper/tor-design.html>.

[HIJK21]

Huang, S., Iyengar, S., Jeyaraman, S., Kushwah, S., Lee, C. K., Luo, Z., Mohassel, P., Raghunathan, A., Shaikh, S., Sung, Y. C., and A. Zhang, "PrivateStats: De-Identified Authenticated Logging at Scale", January 2021, <https://research.fb.com/privatestats>.

[ISSUANCE] Celi, S., Davidson, A., Faz-Hernandez, A., Valdez, S., and C. A. Wood, "Privacy Pass Issuance Protocol", Work in Progress, Internet-Draft, draft-ietf-privacypass-protocol-08, 30 January 2023, <https://

datatracker.ietf.org/doc/html/draft-ietf-privacypass-protocol-08>.

[KLOR20] Kreuter, B., Lepoint, T., Orrù, M., Raykova, M., and Springer International Publishing, "Anonymous Tokens with Private Metadata Bit", Advances in Cryptology - CRYPTO 2020, pp. 308-336, DOI 10.1007/978-3-030-56784-2_11, 2020, <http://dx.doi.org/10.1007/978-3-030-56784-2_11>.

[PrivacyPassCloudflare] Sullivan, N., "Cloudflare Supports Privacy Pass", n.d., <https://blog.cloudflare.com/cloudflare-supports-privacy-pass/>.

[PrivacyPassExtension] "Privacy Pass Browser Extension", n.d., <https://github.com/privacypass/challenge-bypass-extension>.

[RATE-LIMITED] Hendrickson, S., Iyengar, J., Pauly, T., Valdez, S., and C. A. Wood, "Rate-Limited Token Issuance Protocol", Work in Progress, Internet-Draft, draft-privacypass-rate-limit-tokens-03, 6 July 2022, <https://datatracker.ietf.org/doc/html/draft-privacypass-rate-limit-tokens-03>.

[RFC9334] Birkholz, H., Thaler, D., Richardson, M., Smith, N., and W. Pan, "Remote ATtestation procedureS (RATS) Architecture", RFC 9334, DOI 10.17487/RFC9334, January 2023, <https://www.rfc-editor.org/rfc/rfc9334>.

Appendix A. Acknowledgements

The authors would like to thank Eric Kinnear, Scott Hendrickson, Tommy Pauly, Christopher Patton, Benjamin Schwartz, Martin Thomson, Steven Valdez and other contributors of the Privacy Pass Working Group for many helpful contributions to this document.

Authors' Addresses

Alex Davidson
LIP
Lisbon
Portugal

Email: alex.davidson92@gmail.com

Jana Iyengar
Fastly

Email: jri@fastly.com

Christopher A. Wood
Cloudflare
101 Townsend St
San Francisco,
United States of America

Email: caw@heapingbits.net