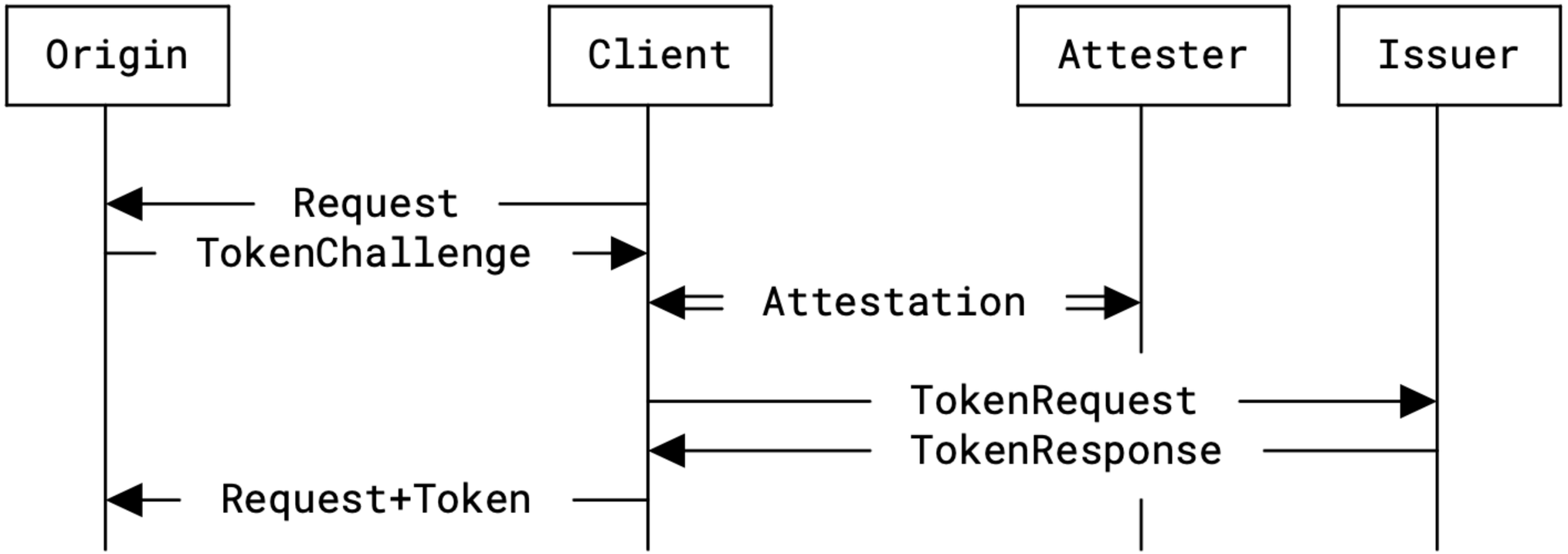
Privacy Pass Issuance Protocol for Anonymous Rate-Limited Credentials

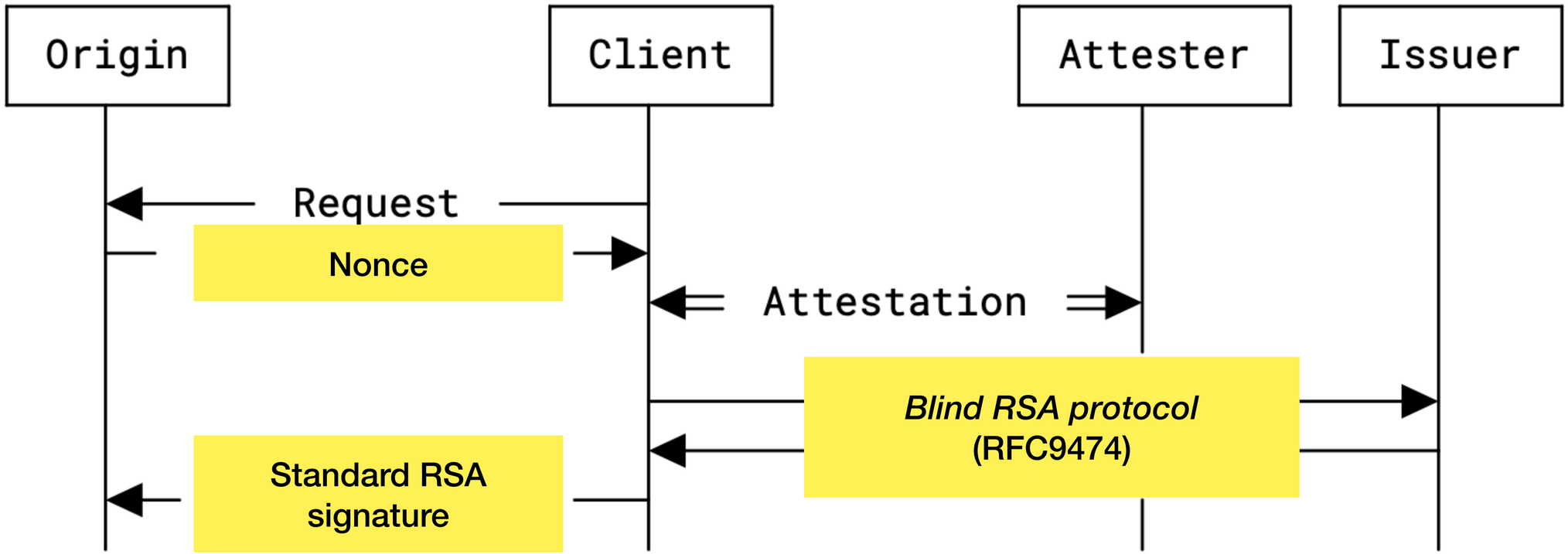
draft-yun-privacypass-arc

IETF 123 - Privacy Pass
Cathie Yun

Privacy Pass (RFC9576)



Privacy Pass (RFC9576)



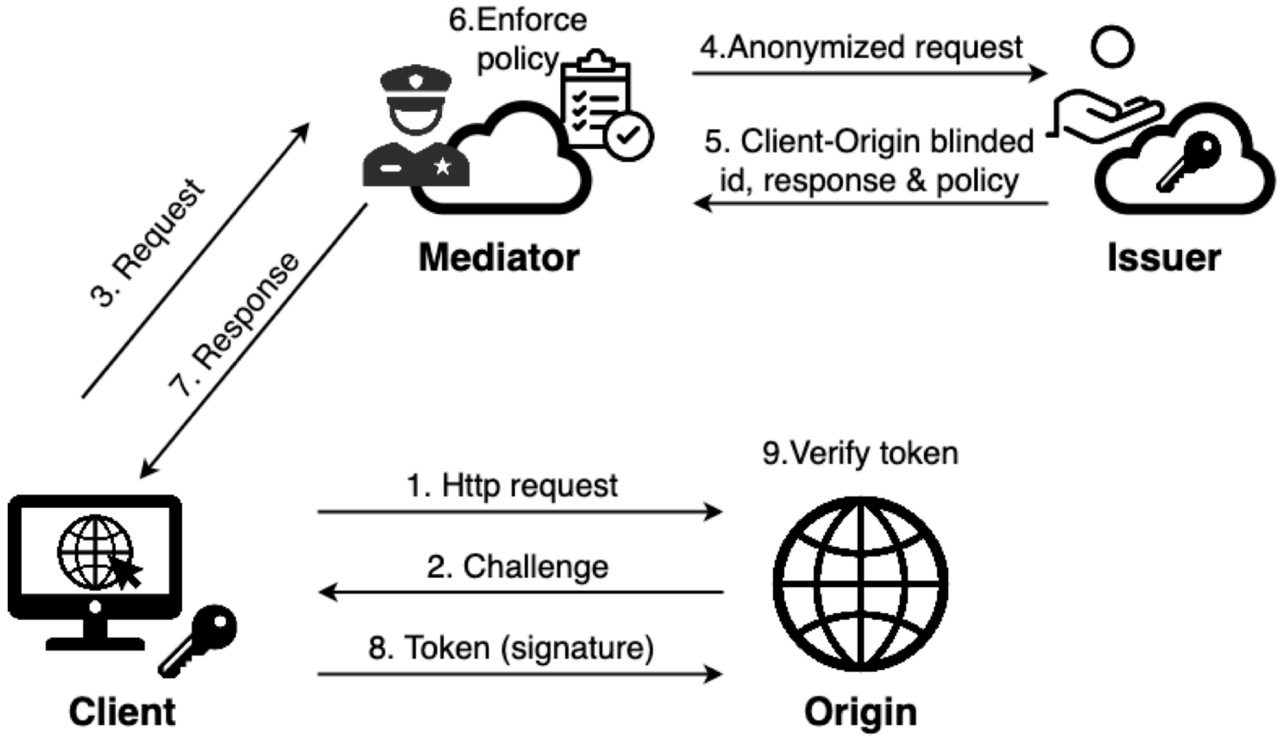
Limitations of RFC9474

Using RSA Blind Signatures

1. No per-client rate limiting
    - Doesn’t prevent individual clients from overloading the system
2. Each token requires its own issuance
    - Privacy concern: attester/issuer learn client’s usage rate
    - Issuance overhead: high issuer load, especially with bursty access

Rate-Limited Privacy Pass

Addresses per-client rate limiting



https://datatracker.ietf.org/doc/draft-ietf-privacypass-rate-limit-tokens/

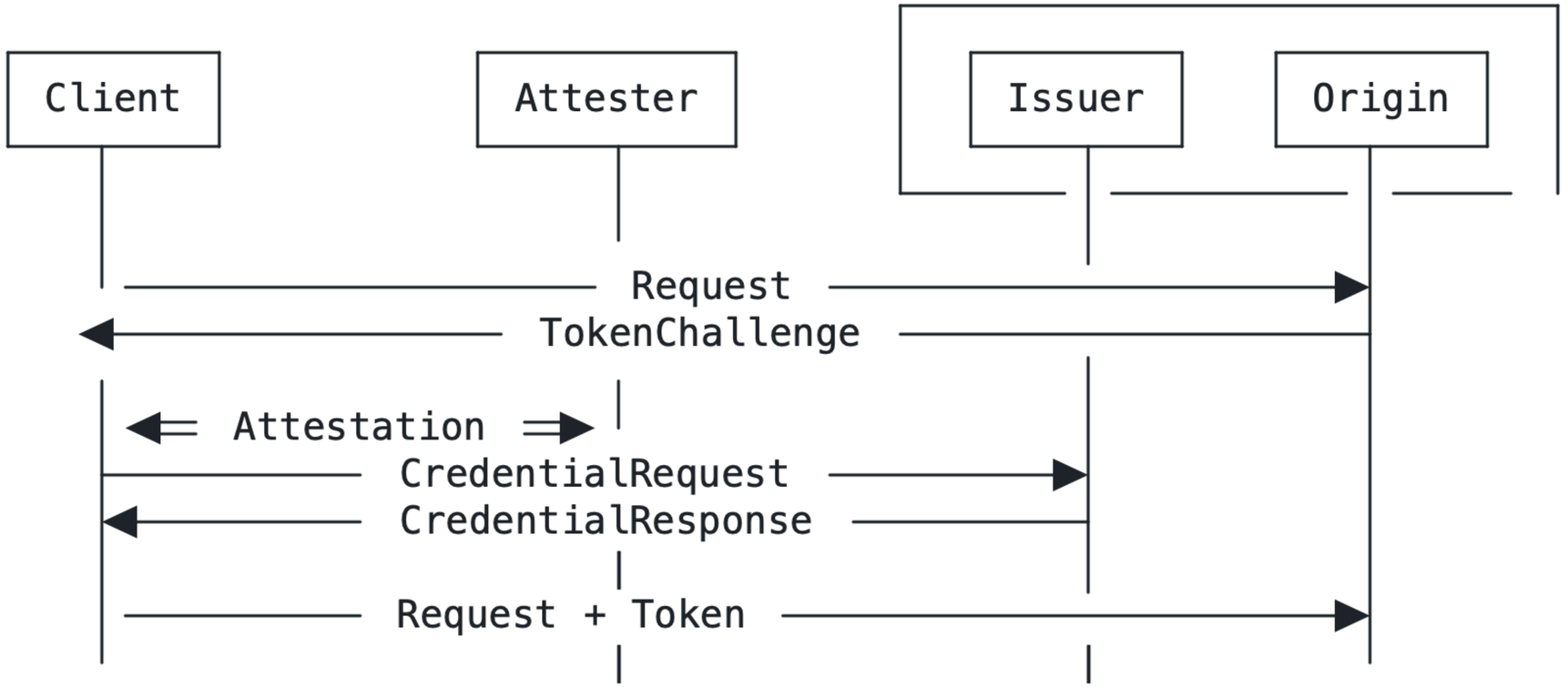
https://eprint.iacr.org/2023/1805.pdf

ARC: Anonymous Rate-limited Credentials

A better approach to rate-limited tokens

1. Supports per-client rate limiting
    - This is built-in, as each client's credential inherently rate limits that client
2. Many tokens can be created from one issuance
    - One issuance allows for creation of one credential
    - Tokens can be derived from that credential, up to the rate limit

ARC overview



Comparison

RSA-based rate-limited tokens	ARC
⚠️ 1:1 issuance:token - no usage pattern privacy - higher issuance overhead	✅ 1:many issuance:token - usage pattern privacy - lower issuance overhead
✅ Public-key verifiable	⚠️ Private-key verifiable

Comparison

RSA-based rate-limited tokens	ARC
⚠️ 1:1 issuance:token - no usage pattern privacy - higher issuance overhead	✅ 1:many issuance:token - usage pattern privacy - lower issuance overhead
✅ Public-key verifiable	⚠️ Private-key verifiable

← Could be solved with pairings.
- Is this of interest?
- Is this in scope for the ARC spec?

ARC token details

In the current spec, an ARC token contains:

- `nonce`: a value in `[0,rateLimit)`, sent in the clear
  - For a credential and presentation context, a nonce should only be used once
- `tag`: element used to enforce rate limit
  - Derived deterministically from nonce, credential secret, presentation context
  - Will be unique if rate limit has not been exceeded

Public nonce

Privacy concerns

`nonce`: a value in `[0,rateLimit)`, sent in the clear

`tag`: element used to enforce rate limit

What if Origin sees two tokens with the same nonce, but different tags?

- For one credential, one nonce will always give you the same tag
- So, this reveals that the tokens came from different credentials
  - With many users and high rate limits, may not have a significant privacy impact
  - However, violates Privacy Pass privacy requirements (RFC9576):
    Cannot distinguish between N tokens that come from the same Client,
    and N tokens that come from N different Clients

Nonce hiding

Rate limit limitations

We would like to hide the nonce, for more robust privacy properties

- Simplest: power-of-two `rateLimit`
  - Pros: simple to specify and implement, low size and cost overhead
  - Cons: power-of-two restriction on `rateLimit`. Is this acceptable?
- More flexible: arbitrary `rateLimit`
  - Pros: `rateLimit` can be an arbitrary value
  - Cons: more experimental and complex cryptography, harder to specify and implement, larger size and cost overhead

Nonce options

1. Public nonce with arbitrary rate limit (current design)
2. Hidden nonce with power-of-two rate limit
3. Hidden nonce with arbitrary rate limit

Next steps

- Working group discussion on the nonce options
- Interest in extensions: public verifiability?
- We’d like to have ARC adopted as a Privacy Pass type
  - Is the working group interested in adoption?