Anonymous Tokens

Michele Orrù

ia.cr/2020/072

Anonymous Tokens

Michele Orrù

joint work with Ben Kreuter, Tancrède Lepoint, Mariana Raykova

ia.cr/2020/072

Definition

Anonymous tokens are lightweight, single-use anonymous credentials.

Definition

Anonymous tokens are lightweight, single-use anonymous credentials.

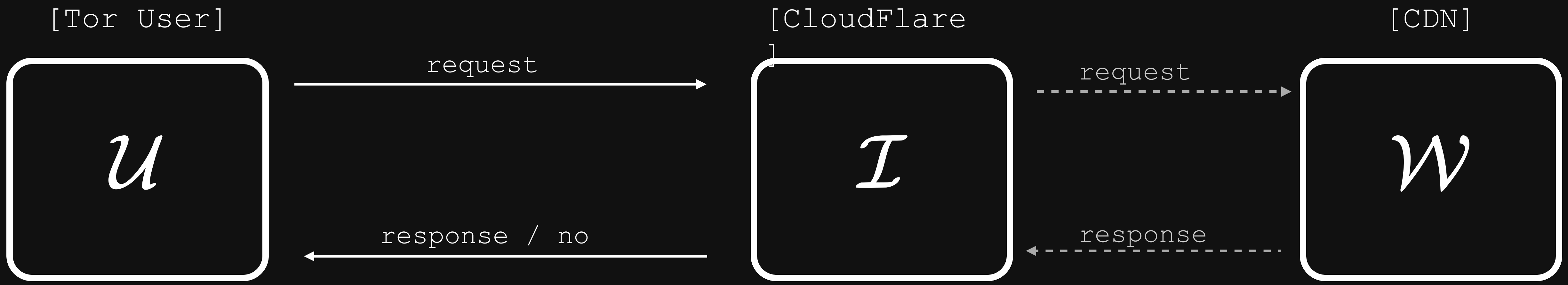
… we focus on secret-key tokens with a private metadata bit.

The Problem

"On the Internet, nobody knows you're a dog."

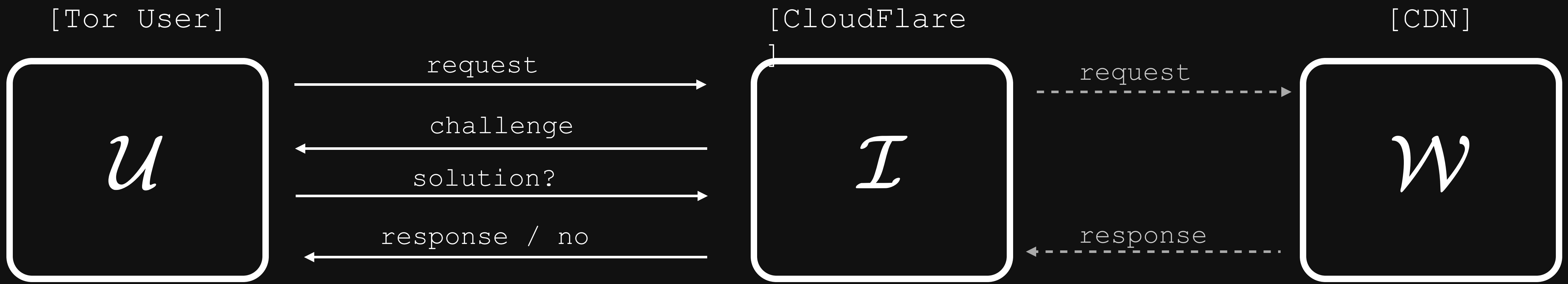
CloudFlare's story

Website protection.



CloudFlare's story

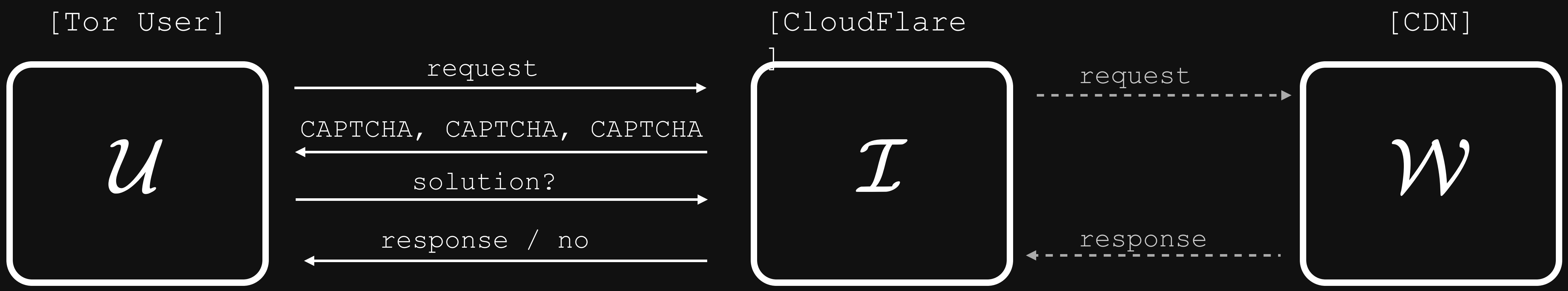
Website protection.



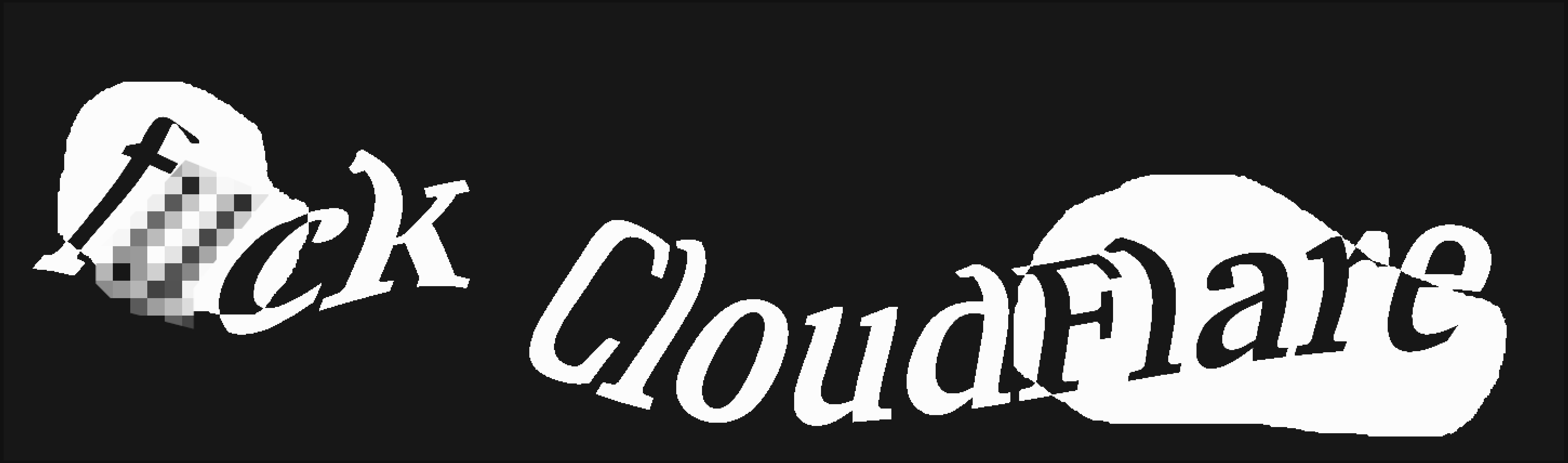
Privacy Pass: Bypassing Internet Challenges Anonymously. [PETS'18]

CloudFlare's story

Website protection.



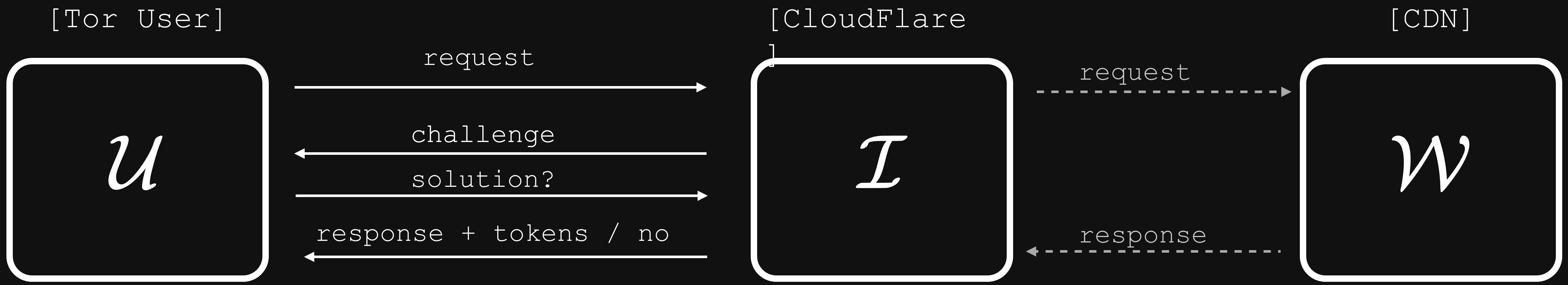
Privacy Pass: Bypassing Internet Challenges Anonymously. [PETS'18]



Art credits: Marie Gutbub. [source]

CloudFlare's story

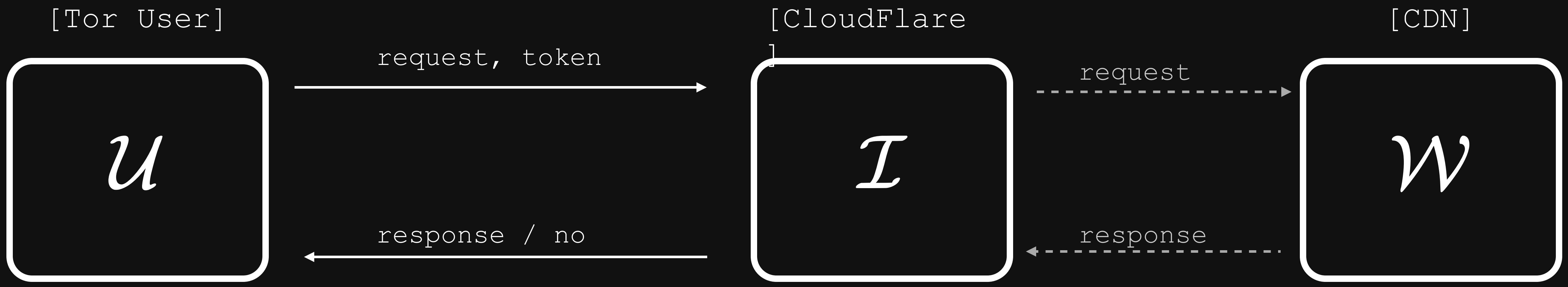
Website protection.



Privacy Pass: Bypassing Internet Challenges Anonymously. [PETS'18]

CloudFlare's story

Website protection.



Privacy Pass: Bypassing Internet Challenges Anonymously. [PETS'18]

Other stories

Micro payments.

Other stories

Fraud prevention.

Fighting fraud using partially blind signatures. [Facebook Engeneering Blog]

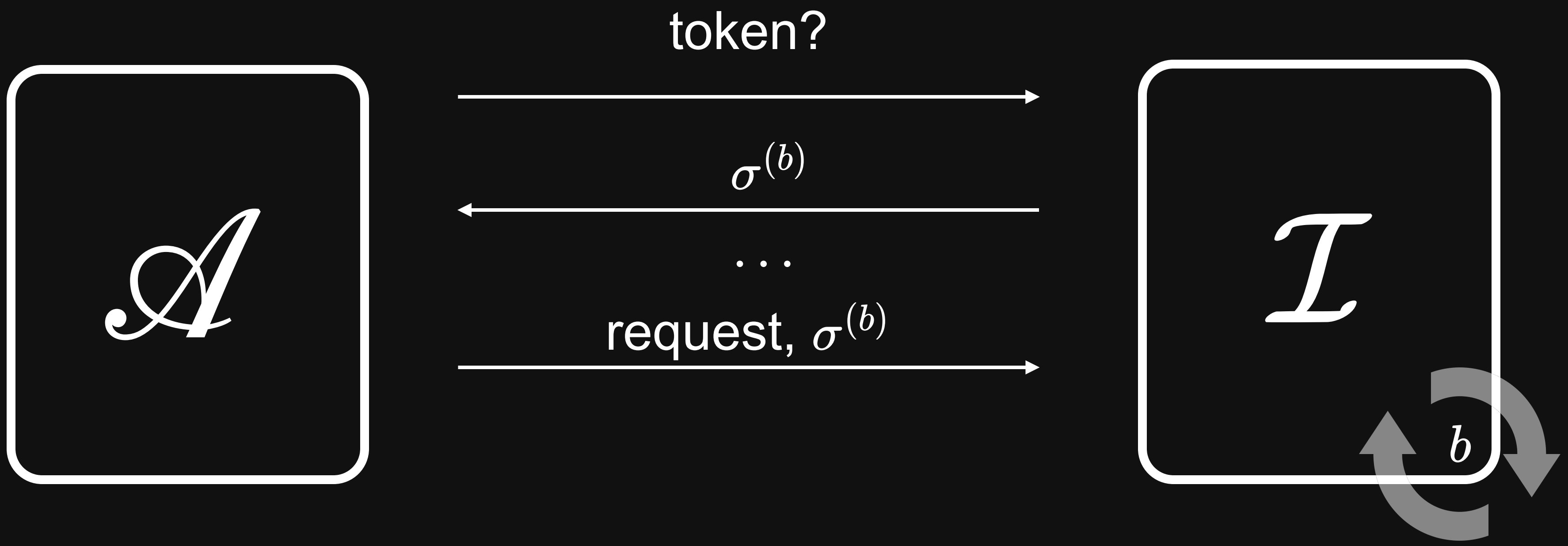
Other stories

Deprecating 3rd party cookies.

Building a more private web: A path towards making third party cookies obsolete. [*Chromium Blog*]

Private medatada

Private medatada



The (formal) problem

Issuance protocol:

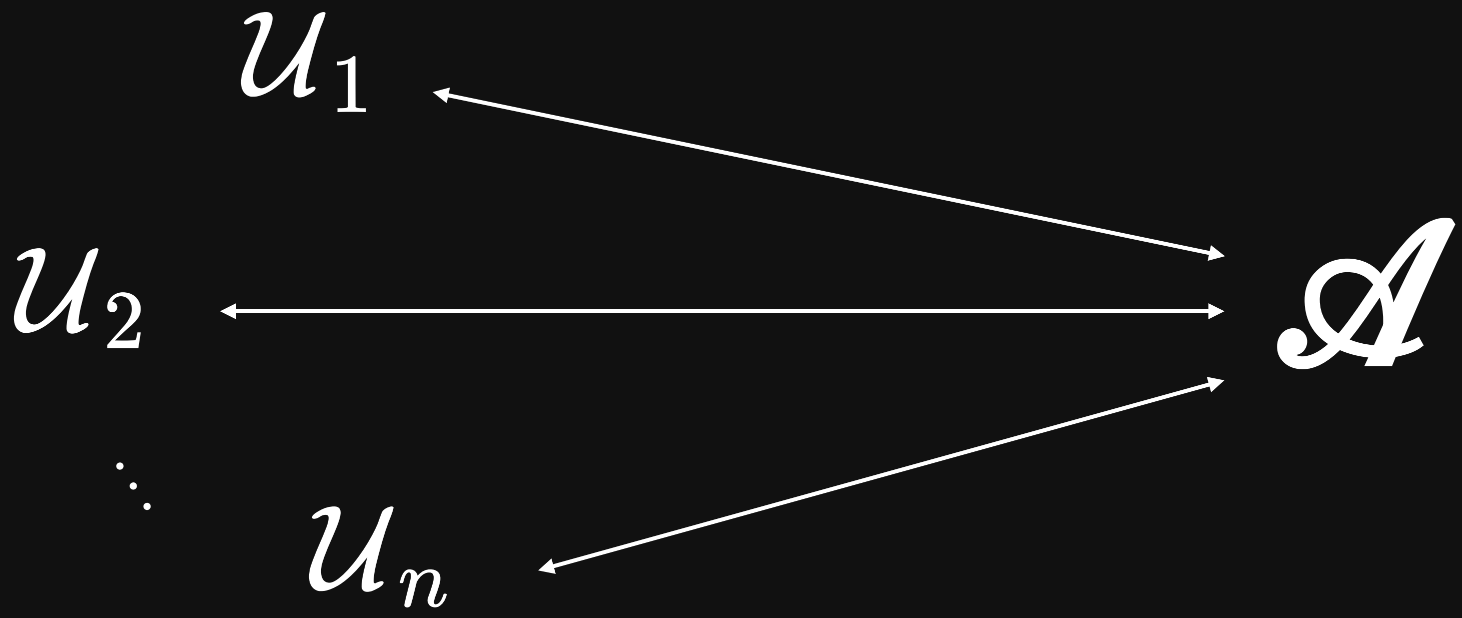
$$\sigma \leftarrow \langle \mathcal{U}(\mathsf{pp}, t), \mathcal{I}(\mathsf{sk}, b) \rangle$$

Redemption algorithm:

$$\{0, 1, \perp\} \leftarrow \mathcal{V}(\mathsf{sk}, t, \sigma)$$

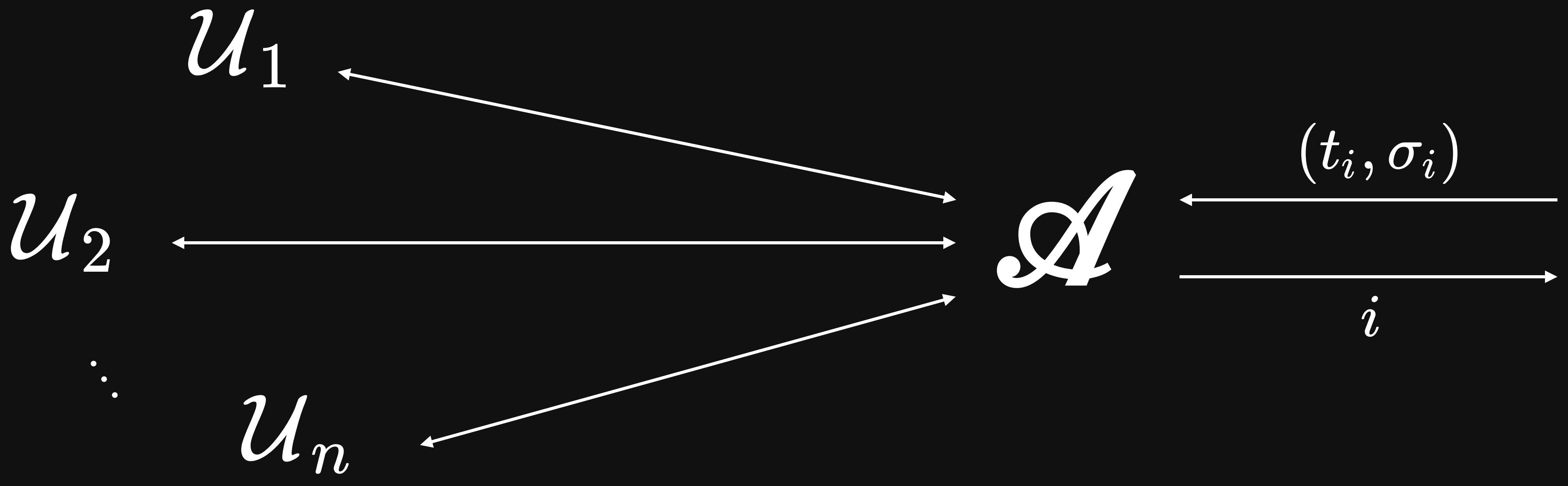
Security notions

- Unlinkability



Security notions

- Unlinkability



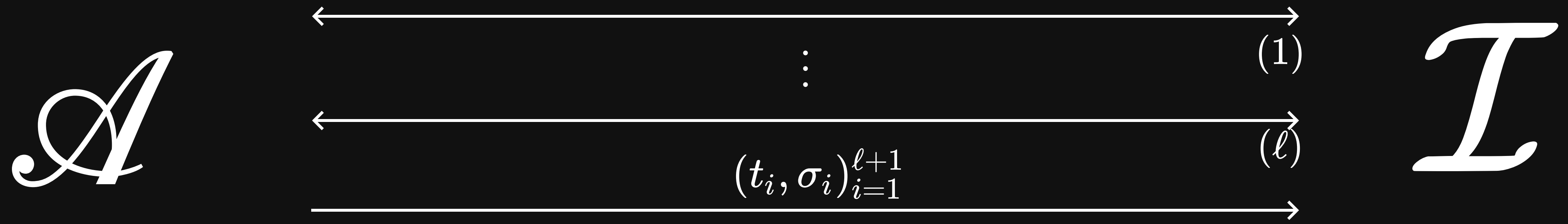
Security notions

- Unlinkability
- One-more unforgeability

$$\mathscr{A} \xrightarrow{(t_i, \sigma_i)_{i=1}^{\ell+1}} \mathcal{I}$$

Security notions

- Unlinkability
- One-more unforgeability



Security Notions

- Unnlinkability
- One-more unforgeability
- Privacy of the metadata bit

$$\mathcal{I}(\mathsf{sk}, b = 0) \overset{\text{ind.}}{\equiv} \mathcal{I}(\mathsf{sk}, b = 1)$$

Standardization

W3C: Trust Token API

```
fetch('https://iacr.org/.well-known/trust-token', {
  trustToken: {
    type: 'token-request',
    issuer: 'ens.fr'
  }
});
```

[Example derived from the original proposal.]

Standardization

W3C: Trust Token API

```
fetch('https://iacr.org/.well-known/trust-token', {
  trustToken: {
    type: 'token-request',
    issuer: 'ens.fr'
  }
});
```

```
fetch('https://eprint.iacr.org/2020/072.pdf', {
  trustToken: {
    type: 'raw-token-redemption',
    issuer: 'ens.fr'
  }
});
```

[Example derived from the original proposal.]

Standardization

W3C: Trust Token API

```
fetch('https://iacr.org/.well-known/trust-token', {
  trustToken: {
    type: 'token-request',
    issuer: 'ens.fr'
  }
});
```

```
fetch('https://eprint.iacr.org/2020/072.pdf', {
  trustToken: {
    type: 'raw-token-redemption',
    issuer: 'ens.fr'
  }
});
```

[Example derived from the original proposal.]

IETF: Privacy Pass draft

```
1.  Introduction

   In some situations, it may only be necessary to check that a clien
   has been previously authorized by a service; without learning any
   other information.  Such lightweight authorization mechanisms can
   useful in quickly assessing the reputation of a client in latency-
   sensitive communication.
```

[Draft version 00]

Our contribution

Our contribution

- Formalization of Anonymous Tokens;

Our contribution

- Formalization of Anonymous Tokens;

- Private Medatada extension;

Our contribution

- Formalization of Anonymous Tokens;
- Private Medatada extension;
- New techniques for removal of zk proofs.

Our contribution

- Formalization of Anonymous Tokens;
- Private Medatada extension;
- New techniques for removal of zk proofs.

Our contribution

- Formalization of Anonymous Tokens;
- Private Medatada extension;
- New techniques for removal of zk proofs.

Related works

Related works

- Anonymous Credentials

Related works

- Anonymous Credentials
- Algebraic MACs

Related works

- Anonymous Credentials
- Algebraic MACs
- Blind Singatures

Privacy Pass

User

Issuer

Privacy Pass

User

Issuer

$$\Gamma := (p, \mathbb{G}, G)$$
$$X = xG$$

Privacy Pass

User

$$\Gamma := (p, \mathbb{G}, G)$$
$$X = xG$$

Issuer

$r \leftarrow \mathbb{Z}_p^*$

$T' := r^{-1}\mathsf{H}(t)$

$$\xrightarrow{\quad T' \quad}$$

Privacy Pass

User

Issuer

$$\Gamma := (p, \mathbb{G}, G)$$
$$X = xG$$

$r \leftarrow \mathbb{Z}_p^*$

$T' := r^{-1}\mathsf{H}(t)$

$$\xrightarrow{\quad T' \quad}$$

$W' := xT'$

$$\xleftarrow{\quad W' \quad}$$

$W := rW'$

Privacy Pass

User | **Issuer**

$$\Gamma := (p, \mathbb{G}, G)$$
$$X = xG$$

User: $r \leftarrow \mathbb{Z}_p^*$

User: $T' := r^{-1}\mathsf{H}(t)$

User → Issuer: T'

Issuer: $W' := xT'$

Issuer → User: W'

User: $W := rW'$

$\cdots$ redemption $\cdots$

User → Issuer: t, W

Issuer:

1. check $x\mathsf{H}(t) = W$
2. add t to spent tokens.

Privacy Pass

User

Issuer

$$\Gamma := (p, \mathbb{G}, G)$$
$$X = xG$$

$r \leftarrow \mathbb{Z}_p^*$

$T' := r^{-1}\mathsf{H}(t)$

$\xrightarrow{\quad T' \quad}$

$W' := xT'$

$\pi :=$ zkp$\left\{ x \begin{bmatrix} G \\ T' \end{bmatrix} = \begin{bmatrix} X \\ W' \end{bmatrix} \right\}$

$\xleftarrow{\quad W', \pi \quad}$

check π

$W := rW'$

$\cdots$ redemption $\cdots$

$\xrightarrow{\quad t, W \quad}$

1. check $x\mathsf{H}(t) = W$
2. add t to spent tokens.

Private metadata?

User | **Issuer**

$$\Gamma := (p, \mathbb{G}, G)$$
$$X = x_b G, \;\; b \in \{0,1\}$$

$r \leftarrow \mathbb{Z}_p^*$

$T' := r^{-1}\mathsf{H}(t)$

$$\xrightarrow{\quad T' \quad}$$

$W' := x_b T'$

$\pi :=$ zkp$\left\{ x_b \begin{bmatrix} G \\ T' \end{bmatrix} = \begin{bmatrix} X_b \\ W' \end{bmatrix} \right\}$

$$\xleftarrow{\quad W', \pi \quad}$$

check π

$W := rW'$

$\cdots$ redemption $\cdots$

$$\xrightarrow{\quad t, W \quad}$$

1. check b s.t. $x_b\mathsf{H}(t) = W$
2. add t to spent tokens.

WRONG

Attack

Adversary

Issuer

$$\Gamma := (p, \mathbb{G}, G)$$
$$X_b = x_b G, \;\; b \in \{0,1\}$$

$r, s \leftarrow \mathbb{Z}_p^*$

$T' := r^{-1}\mathsf{H}(t)$

$S' := s^{-1}\mathsf{H}(t)$

$$\xrightarrow{\quad T' \quad}$$
$$\xrightarrow{\quad S' \quad}$$

$W' := x_0 T'$

$V' := x_1 S'$

Attack

Adversary | **Issuer**

$$\Gamma := (p, \mathbb{G}, G)$$
$$X_b = x_b G, \;\; b \in \{0,1\}$$

$r, s \leftarrow \mathbb{Z}_p^*$

$T' := r^{-1}\mathsf{H}(t)$

$S' := s^{-1}\mathsf{H}(t)$

$$\xrightarrow{\quad T' \quad}$$
$$\xrightarrow{\quad S' \quad}$$

$W' := x_0 T'$

$V' := x_1 S'$

$$\xleftarrow{\quad W' \quad}$$
$$\xleftarrow{\quad V' \quad}$$

$rW' \stackrel{?}{=} sV'$

Privacy Pass variant

User | **Issuer**

$$\Gamma := (p, \mathbb{G}, G, H)$$

$$X = xG + yH$$

$r \leftarrow \mathbb{Z}_p^*$

$T' := r^{-1}\mathsf{H}(t)$

$\xrightarrow{T'}$

$s \leftarrow \{0,1\}^\lambda;\ S' := \mathsf{H}(T', s)$

$W := xT' + yS'$

Privacy Pass variant

User

Issuer

$$\Gamma := (p, \mathbb{G}, G, H)$$

$$X = xG + yH$$

$r \leftarrow \mathbb{Z}_p^*$

$T' := r^{-1}\mathsf{H}(t)$

$\xrightarrow{T'}$

$s \leftarrow \{0,1\}^\lambda;\ S' := \mathsf{H}(T', s)$

$W := xT' + yS'$

$\xleftarrow{s, W', \pi}$

$W := rW'$

$S := r\mathsf{H}(T', s)$

$\cdots$ redemption $\cdots$

$\xrightarrow{t, S, W}$

1. check $x\mathsf{H}(t) + yS = W$
2. add t to spent tokens.

Privacy Pass variant

User | **Issuer**

$$\Gamma := (p, \mathbb{G}, G, H)$$

$$X = xG + yH$$

$r \leftarrow \mathbb{Z}_p^*$

$T' := r^{-1}\mathsf{H}(t)$

$\xrightarrow{\quad T' \quad}$

$s \leftarrow \{0,1\}^\lambda;\ S' := \mathsf{H}(T', s)$

$W := xT' + yS'$

$\pi :=$

zkp $\left\{ x \begin{bmatrix} G \\ T' \end{bmatrix} + y \begin{bmatrix} H \\ S' \end{bmatrix} = \begin{bmatrix} X \\ W' \end{bmatrix} \right\}$

$\xleftarrow{\quad s, W', \pi \quad}$

check π

$W := rW'$

$S := r\mathsf{H}(T', s)$

$\cdots$ redemption $\cdots$

$\xrightarrow{\quad t, S, W \quad}$

1. check $x\mathsf{H}(t) + yS = W$
2. add t to spent tokens.

Private metadata

User | **Issuer**

$$\Gamma := (p, \mathbb{G}, G, H)$$

$$X_b = x_b G + y_b H, \;\; b \in \{0,1\}$$

User: $r \leftarrow \mathbb{Z}_p^*$

User: $T' := r^{-1}\mathsf{H}(t)$

User → Issuer: T'

Issuer: $s \leftarrow \{0,1\}^\lambda;\; S' := \mathsf{H}(T', s)$

Issuer: $W := x_b T' + y_b S'$

Issuer: $\pi :=$ zkp $\left\{ x_b \begin{bmatrix} G \\ T' \end{bmatrix} + y_b \begin{bmatrix} H \\ S' \end{bmatrix} = \begin{bmatrix} X_b \\ W' \end{bmatrix} \right\}$

Issuer → User: s, W', π

User: check π

User: $W := rW'$

User: $S := r\mathsf{H}(T', s)$

$\cdots$ redemption $\cdots$

User → Issuer: t, S, W

Issuer:

1. check b s.t. $x_b\mathsf{H}(t) + y_b S = W$
2. add t to spent tokens.

Removing the zk proof

User | **Issuer**

$$\Gamma := (p, \mathbb{G}, G)$$
$$X = xG$$

$r, \rho \leftarrow \mathbb{Z}_p^*$

$T' := r(\mathsf{H}(t) - \rho G)$

$\xrightarrow{T'}$

$W' := xT'$

$\xleftarrow{W', \pi}$

$W := r^{-1}W' + \rho X$

$\cdots$ redemption $\cdots$

$\xrightarrow{t, W}$

1. check $x\mathsf{H}(t) = W$
2. add t to spent tokens.

Concrete security

Concrete security

- One-more Diffie-Hellman is not extensively studied;

Concrete security

- One-more Diffie-Hellman is not extensively studied;
- Token Hijacking;

Concrete security

- One-more Diffie-Hellman is not extensively studied;
- Token Hijacking;
- Engeneering issues.

Implementation

In Rust, using curve25519-dalek::Ristretto.

```
#[test]
fn it_works() {
        let mut csrng = rand::rngs::OsRng;
        // generate a keypair
        let keypair = KeyPair::generate(&mut csrng);

        // get the public parameters
        let pp = PublicParams::from(&keypair);
        // client's first message (the blinded token)
        let blinded_token = pp.generate_token(&mut csrng);
        // server's reponse (the signed token) with hidden metadata bit 0
        let signed_token = keypair.sign(&mut csrng, &blinded_token.to_bytes(), 0);
        // clien'ts unbliding (the final token)
        let token = blinded_token.unblind(signed_token);
        assert!(token.is_ok());

        // verification of the token
        assert!(keypair.verify(&token.unwrap()).is_ok());
}
```

Check out [benchmarks report] for fancy stats.

Future directions

Future directions

- public metadata

Future directions

- public metadata
- public verifiability
  - blind BLS
  - blind Okamoto-Schnorr? broken :(

Future directions

- public metadata
- public verifiability
  - blind BLS
  - blind Okamoto-Schnorr? broken :(
- batching proofs

Future directions

- public metadata
- public verifiability
  - blind BLS
  - blind Okamoto-Schnorr? broken :(
- batching proofs

Future directions

- public metadata
- public verifiability
  - blind BLS
  - blind Okamoto-Schnorr? broken :(
- batching proofs

Future directions

- public metadata
- public verifiability
  - blind BLS
  - blind Okamoto-Schnorr? broken :(
- batching proofs