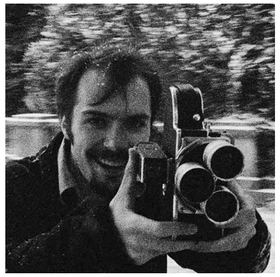
Threshold BBS+ Signatures for Distributed Anonymous Credential Issuance

Jack Doerner, Technion + Reichman



Yashvanth Kondi, Aarhus University

Eysa Lee, NEU → Brown

abhi shelat, NEU

and LaKyah Tyner, NEU

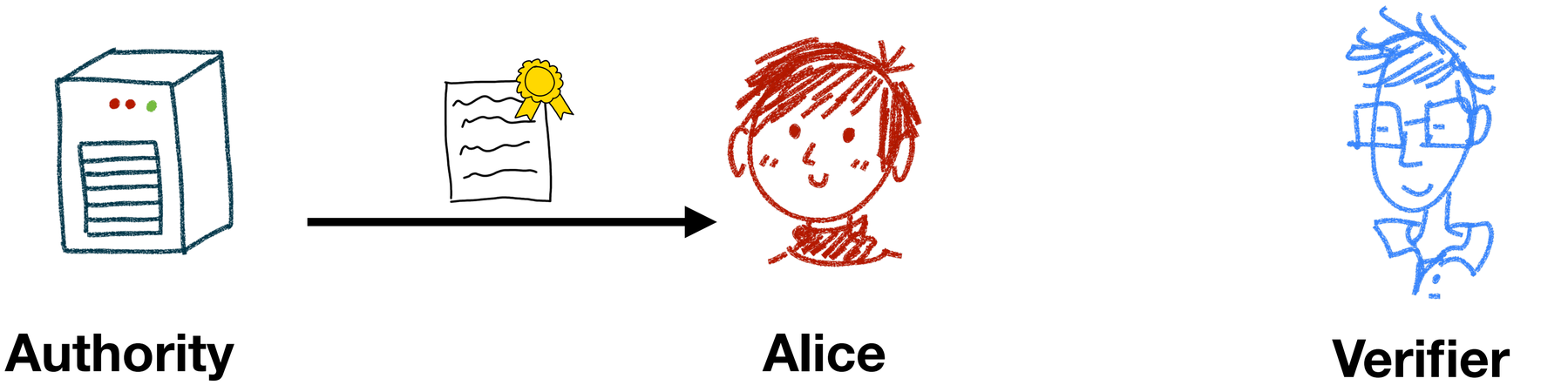
Digital Signatures as Credentials

Authority

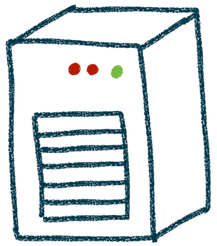
Alice

Verifier

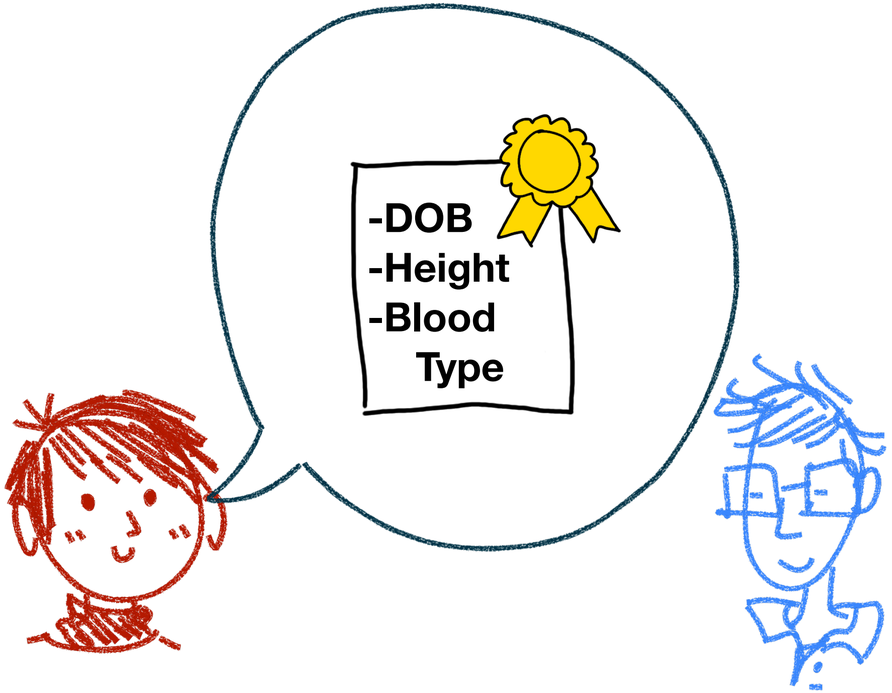
Digital Signatures as Credentials



Digital Signatures as Credentials



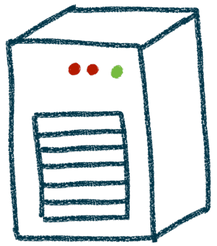
Authority



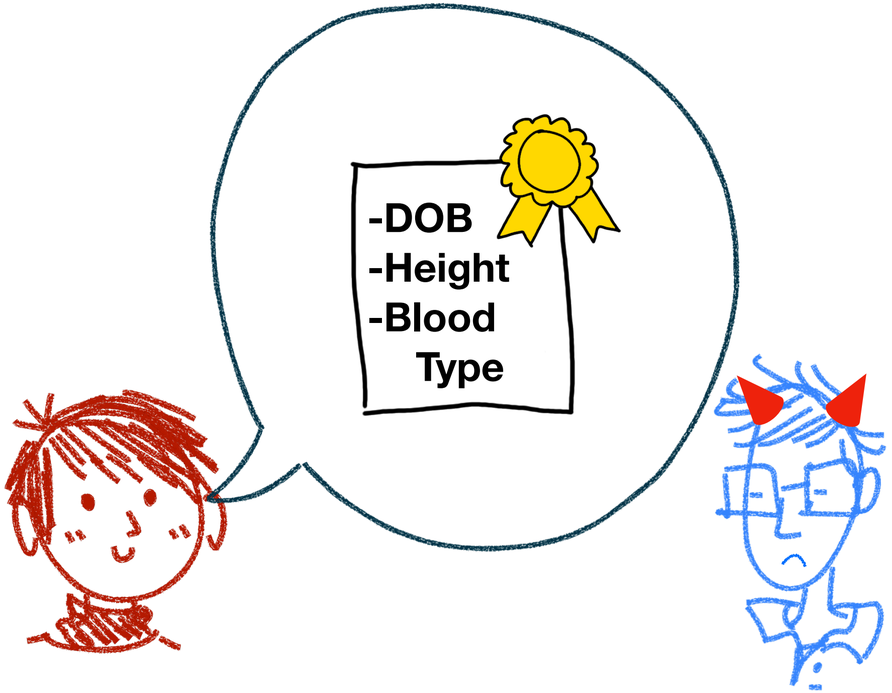
Alice

Verifier for Gammeldansk Subscription

Digital Signatures as Credentials



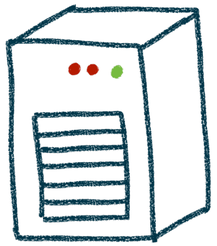
Authority



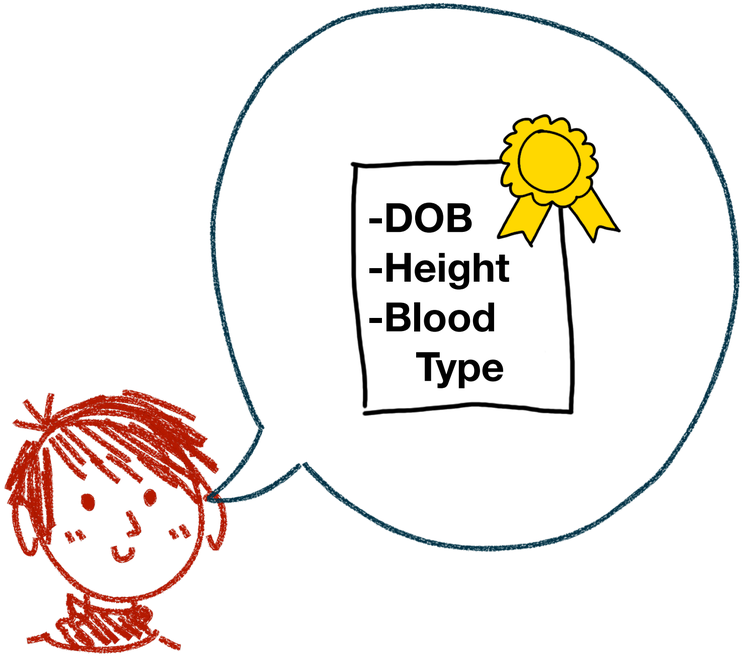
Alice

Verifier for Gammeldansk Subscription

Privacy Concerns

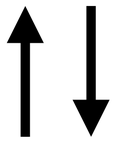


Authority



Alice

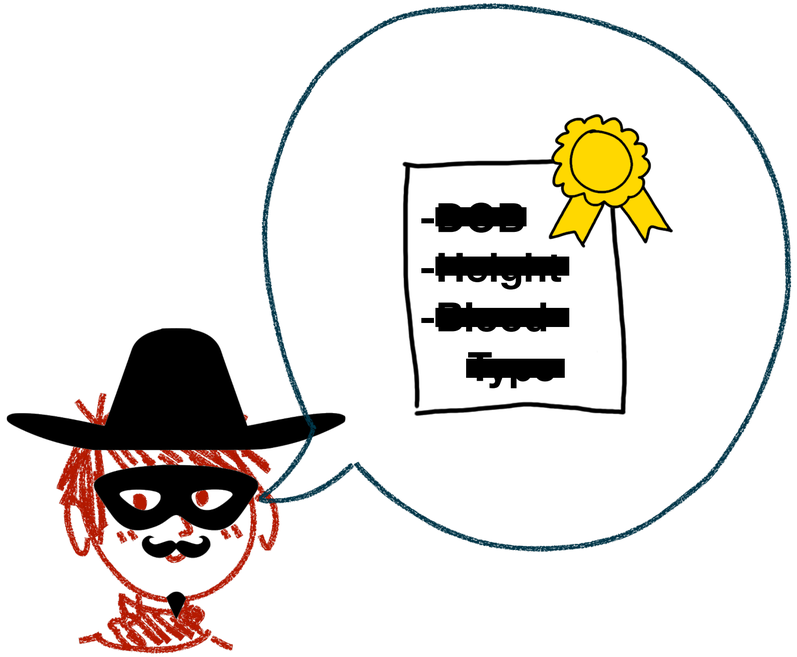
Verifier 1



Verifier 2

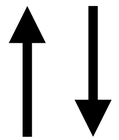
Privacy Concerns

Authority



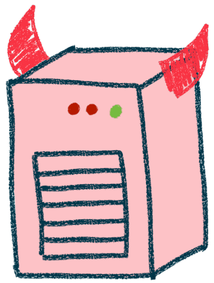
Alice

Verifier 1



Verifier 2

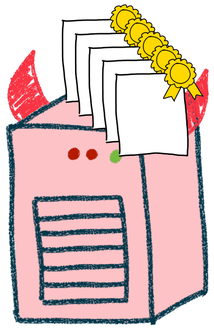
Corrupt Authority



Authority

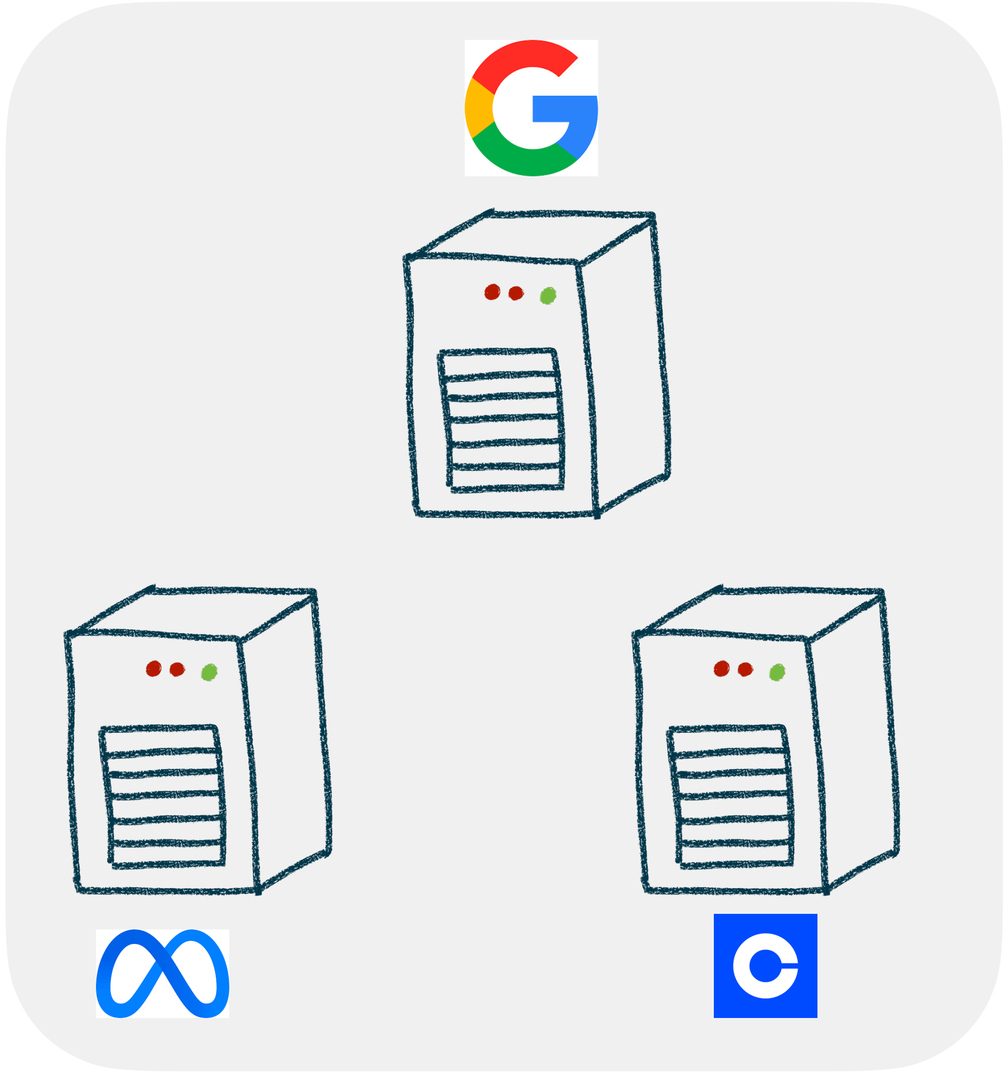
Alice

Corrupt Authority



Authority

Alice

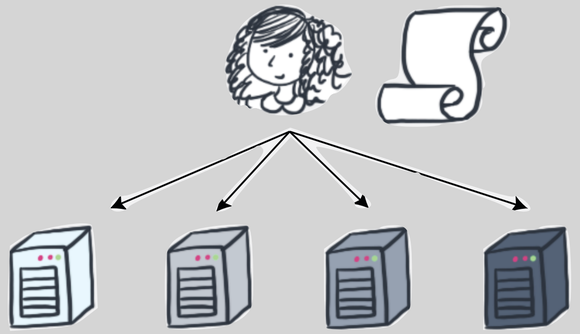


Authority

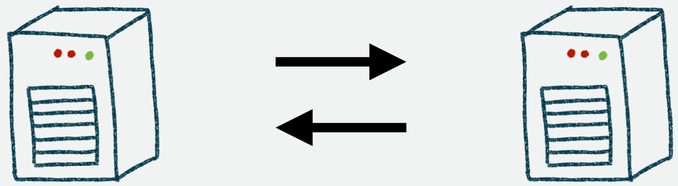
MPC!

Desired Issue Protocol flow

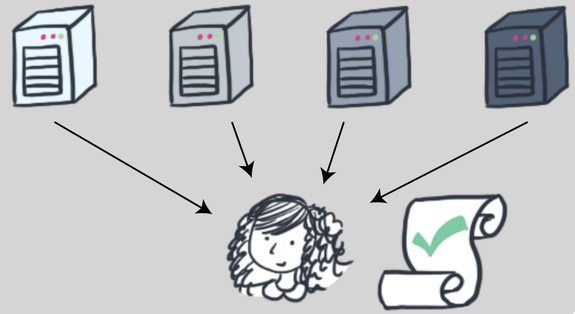
Request Signature



Pairwise Single Roundtrip



Reconstruct Signature



BBS+ [Au Susilo Mu 06]

- Efficient
- Secure under well known assumptions
- Used in privacy preserving applications
- Being standardized by the IETF

- Scheme specific properties
  - Allows for a vector of messages to be signed
  - Selectively reveal messages

Signing with BBS+

$$sk = x$$
$$pk = (g_2^x, h_1, \ldots, h_{L+1})$$

$$\text{BBS} + \text{Sign}(sk, m \in \mathbb{Z}_p^L) :$$

$$e, s \leftarrow \mathbb{Z}_p$$

$$A = \left(g_1 \; h_1^s \prod_{i=1}^{L} h_{i+1}^{m_i} \right)^{1/(x+e)}$$

$$\sigma := (A, e, s)$$

Signing with BBS+

$sk = x$
$pk = (g_2^x, h_1, \ldots, h_{L+1})$

$\mathsf{BBS} + \mathsf{Sign}(sk, m \in \mathbb{Z}_p^L)$:

$e, s \leftarrow \mathbb{Z}_p$

$$A = \left(g_1 \; h_1^s \prod_{i=1}^{L} h_{i+1}^{m_i} \right)^{1/(x+e)}$$

$\sigma := (A, e, s)$

Signing with BBS+

$sk = x$
$pk = (g_2^x, h_1, \ldots, h_{L+1})$

$\mathsf{BBS} + \mathsf{Sign}(sk, m \in \mathbb{Z}_p^L) :$

$e, s \leftarrow \mathbb{Z}_p$

$A = (B)^{1/(x+e)}$

$\sigma := (A, e, s)$

$B = B(pk, m, s)$

Signing with BBS+

$sk = x$
$pk = (g_2^x, h_1, \ldots, h_{L+1})$

$\mathsf{BBS} + \mathsf{Sign}(sk, m \in \mathbb{Z}_p^L)$:

$e, s \leftarrow \mathbb{Z}_p$

$A = (B)^{1/(x+e)}$

$\sigma := (A, e, s)$

Secure Inversion

$B = B(pk, m, s)$

Signing with BBS+

$$sk = x$$
$$pk = (g_2^x, h_1, \ldots, h_{L+1})$$

- Compute using techniques from [DOK+20]
  - EC operations in exponent

$$\text{BBS} + \text{Sign}(sk, m \in \mathbb{Z}_p^L) :$$
$$e, s \leftarrow \mathbb{Z}_p$$
$$A = (B)^{1/(x+e)}$$
$$\sigma := (A, e, s)$$

Secure Inversion

$$B = B(pk, m, s)$$

Signing with BBS+

$$sk = x$$
$$pk = (g_2^x, h_1, \ldots, h_{L+1})$$

- Compute using techniques from [DOK+20]
  - EC operations in exponent
  - MPC for output in EC group

$$\text{BBS} + \text{Sign}(sk, m \in \mathbb{Z}_p^L):$$
$$e, s \leftarrow \mathbb{Z}_p$$
$$A = (B)^{1/(x+e)}$$
$$\sigma := (A, e, s)$$

Secure Inversion

$$B = B(pk, m, s)$$

Signing with BBS+

$$sk = x$$
$$pk = (g_2^x, h_1, \ldots, h_{L+1})$$

- Compute using techniques from [DOK+20]
  - EC operations in exponent
  - MPC for output in EC group
    - MACing for active security

$$\text{BBS} + \text{Sign}(sk, m \in \mathbb{Z}_p^L) :$$
$$e, s \leftarrow \mathbb{Z}_p$$
$$A = (B)^{1/(x+e)}$$
$$\sigma := (A, e, s)$$

Secure Inversion

$$B = B(pk, m, s)$$

[BB89] Secure Inversion (in our context)

- Compute $\frac{1}{[x+e]}$ through $\frac{[r]}{([x+e])\cdot[r]}$

[BB89] Secure Inversion (in our context)

- Compute $\frac{1}{[x+e]}$ through $\frac{[r]}{([x+e])\cdot[r]}$
- Via 2-round multiplier [DOK+20]

[BB89] Secure Inversion (in our context)

- Compute $\frac{1}{[x+e]}$ through $\frac{[r]}{([x+e])\cdot[r]}$
- Via 2-round multiplier [DOK+20]
- Sample $[r]$

[BB89] Secure Inversion (in our context)

- Compute $\frac{1}{[x+e]}$ through $\frac{[r]}{([x+e])\cdot[r]}$
- Via 2-round multiplier [DOK+20]
- Sample $[r]$
- Send $[r],\ [x]$ to multiplier
- Parties get $[u] = [(x+e)\cdot r]$

Threshold BBS+ Signing Protocol

(m, J)

(m, J)

(m, J)

Our Issuing protocol

$(e_i, s_i, r_i) \leftarrow \mathbb{F}_p$

Local

To party P_j | From party P_j

$(e_i, s_i) \rightarrow$ Com

$sk_i \rightarrow$ Mul1

$(e_i, s_i) \rightarrow$ Open

$r_i \rightarrow$ Mul2

$R_i \leftarrow B^{r_i}$

$u_i \leftarrow (x_i + e) \cdot r_i$

(e, s, R_i, u_i)

To client

Our Issuing protocol

$(e_i, s_i, r_i) \leftarrow \mathbb{F}_p$

$(e_i, s_i) \rightarrow$ Com

$sk_i \rightarrow$ Mul1

$(e_i, s_i) \rightarrow$ Open

$r_i \rightarrow$ Mul2

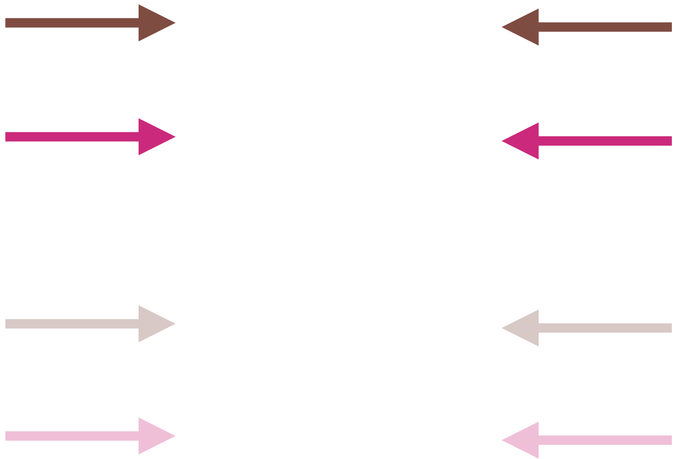
$R_i \leftarrow B^{r_i}$

$u_i \leftarrow (x_i + e) \cdot r_i$

Local

To party P_j

From party P_j



(e, s, R_i, u_i)

To client

Our Issuing protocol

$(e_i, s_i, r_i) \leftarrow \mathbb{F}_p$

$(e_i, s_i) \rightarrow$ Com

$sk_i \rightarrow$ Mul1

$(e_i, s_i) \rightarrow$ Open

$r_i \rightarrow$ Mul2

$R_i \leftarrow B^{r_i}$

$u_i \leftarrow (x_i + e) \cdot r_i$

Local

To party P_j | From party P_j

(e, s, R_i, u_i)

To client

Our Issuing protocol

$(e_i, s_i, r_i) \leftarrow \mathbb{F}_p$

$(e_i, s_i) \rightarrow$ Com

$sk_i \rightarrow$ Mul1

$(e_i, s_i) \rightarrow$ Open

$r_i \rightarrow$ Mul2

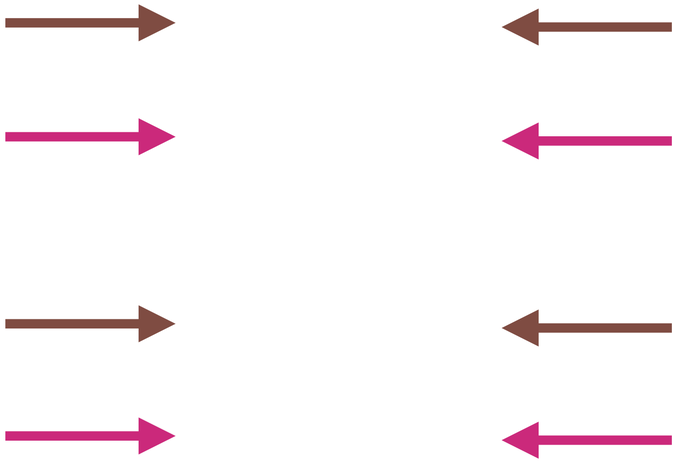
$R_i \leftarrow B^{r_i}$

$u_i \leftarrow (x_i + e) \cdot r_i$

Local

To party P_j

From party P_j



(e, s, R_i, u_i)

To client

Client Reconstruction

(e, s, R_i, u_i)

(e, s, R_i, u_i)

(e, s, R_i, u_i)

Computes

$$A = \left(\prod R_i\right)^{\frac{1}{\sum u_i}}$$

Client Reconstruction

(e, s, R_i, u_i)

(e, s, R_i, u_i)

(e, s, R_i, u_i)

Computes

$$A = \left(\prod R_i\right)^{\frac{1}{\sum u_i}}$$

$$= (B^r)^{\frac{1}{(x+e)\cdot r}}$$

Client Reconstruction

(e, s, R_i, u_i)

(e, s, R_i, u_i)

(e, s, R_i, u_i)

Computes

$$A = \left(\prod R_i\right)^{\frac{1}{\sum u_i}}$$

$$= (B^r)^{\frac{1}{(x+e)\cdot r}}$$

$$= B^{\frac{1}{x+e}}$$

Client Reconstruction

(e, s, R_i, u_i)

(e, s, R_i, u_i)

(e, s, R_i, u_i)

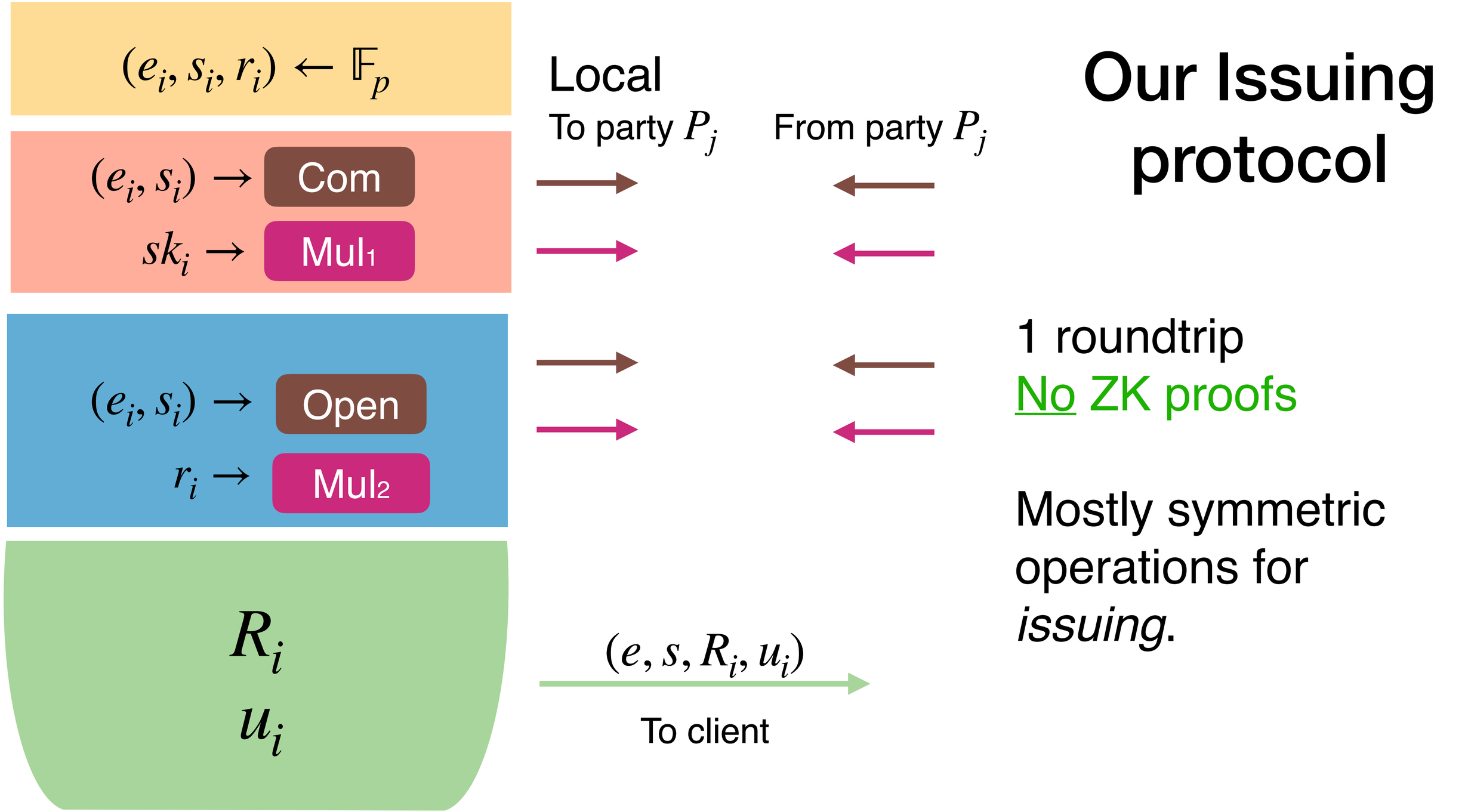
Computes

$$A = \left(\prod R_i\right)^{\frac{1}{\sum u_i}}$$

$$= (B^r)^{\frac{1}{(x+e)\cdot r}}$$

$$= B^{\frac{1}{x+e}}$$

Outputs (A, e, s) if it verifies



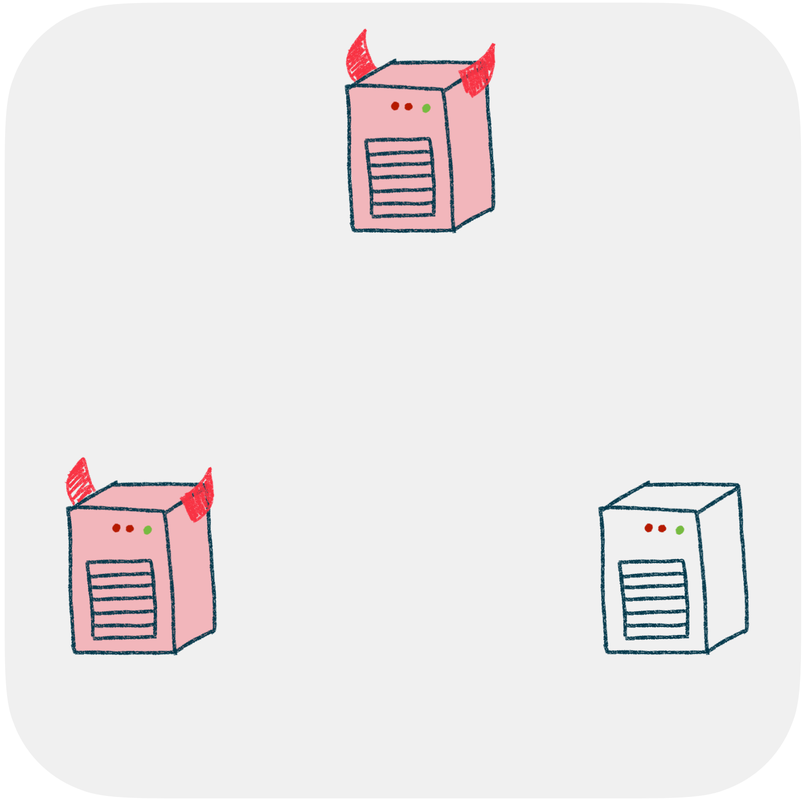
Our Issuing protocol

1 roundtrip

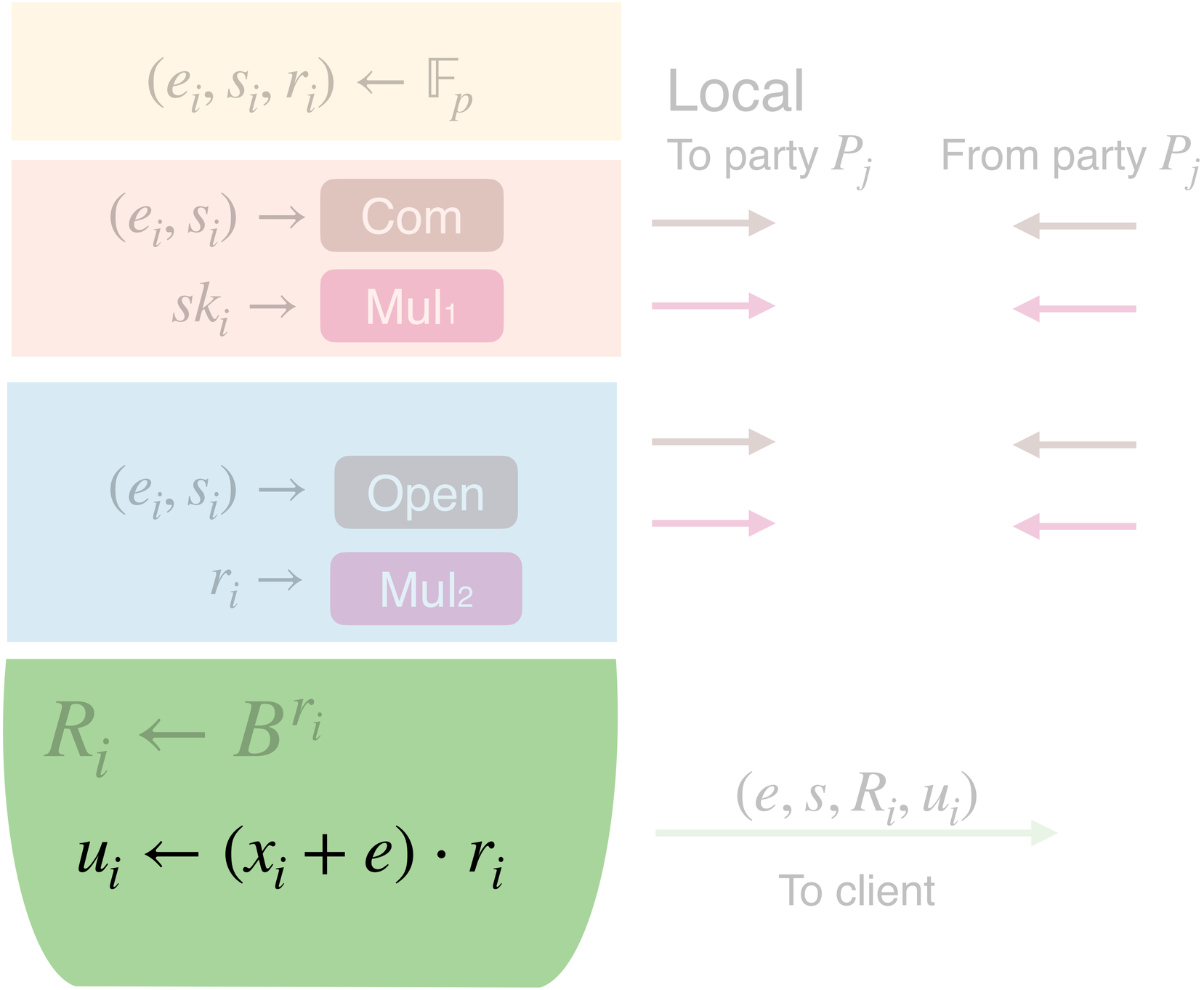
No ZK proofs

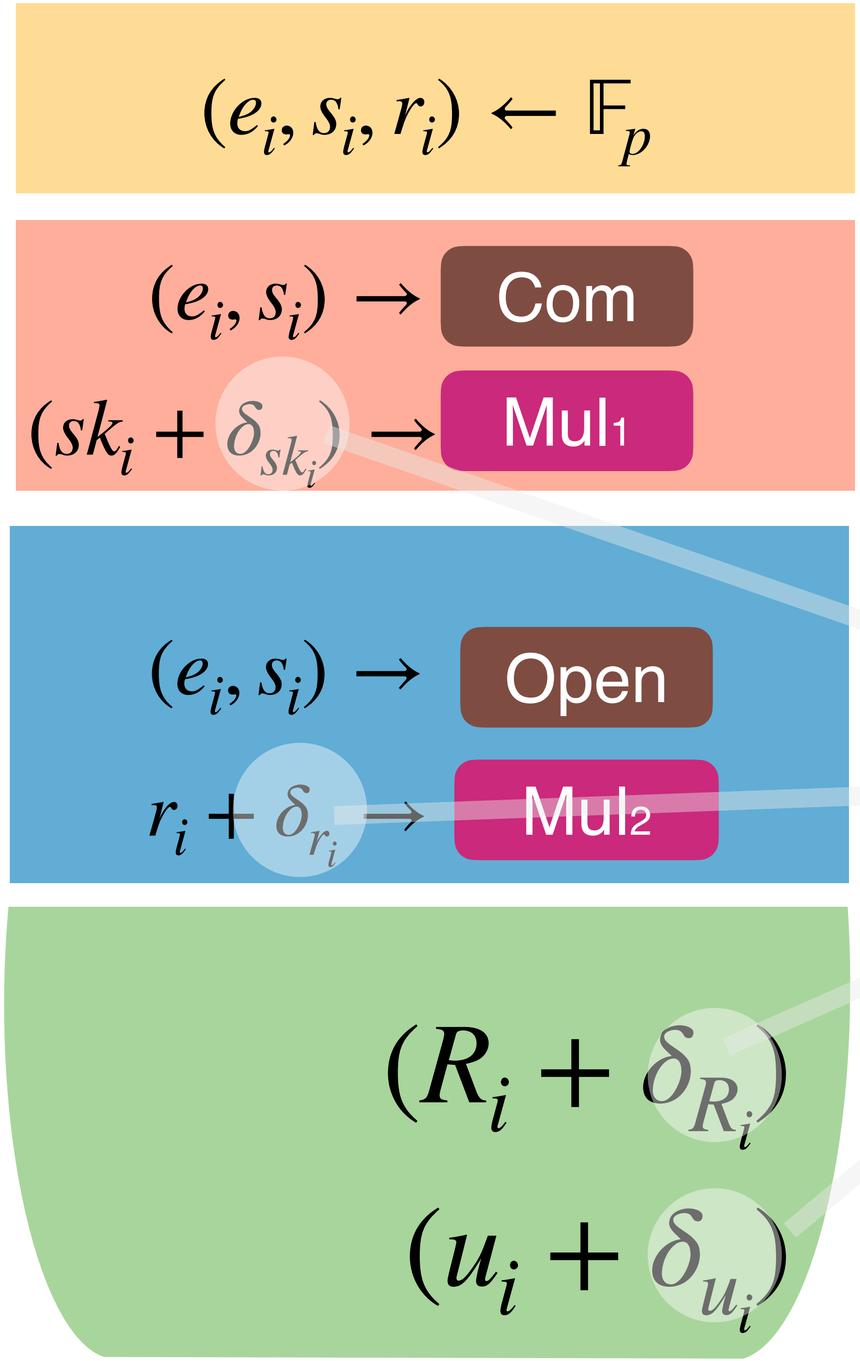
Mostly symmetric operations for *issuing*.

Malicious Security



Malicious Security





Malicious Security Idea

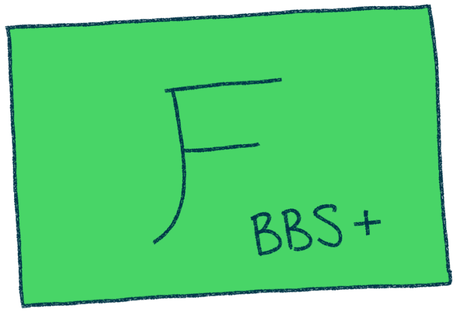
All cheating can be characterized by “additive offsets to messages.”

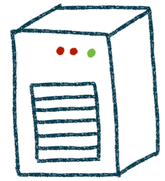
There are only 4 offsets to consider.

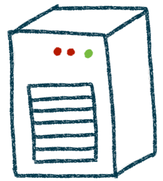
Cases for Cheating:
(1) offsets cancel out
(2) one offset serves as a valid signature

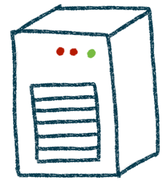
BBS+ Functionality for Weak Partially-Blind Signing

Signer can issue a signature without seeing contents of message





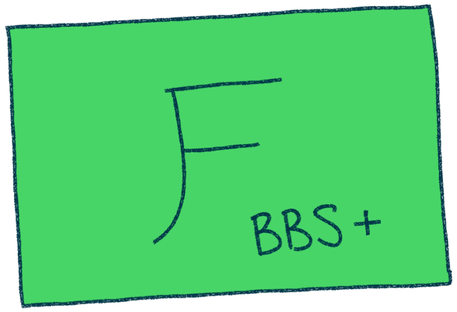


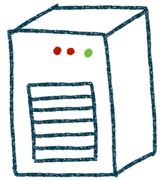


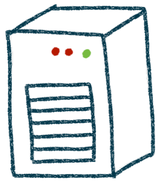
BBS+ Functionality for Weak Partially-Blind Signing

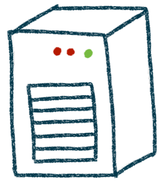
Weakly blind: message is hidden

Signer can issue a signature without seeing contents of message







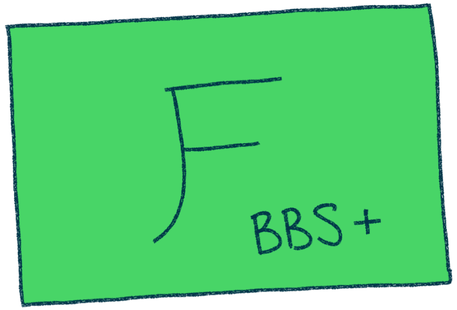


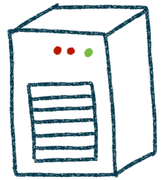
BBS+ Functionality for Weak Partially-Blind Signing

Weakly blind: message is hidden

Signer can see some predicate on the message

Signer can issue a signature without seeing contents of message

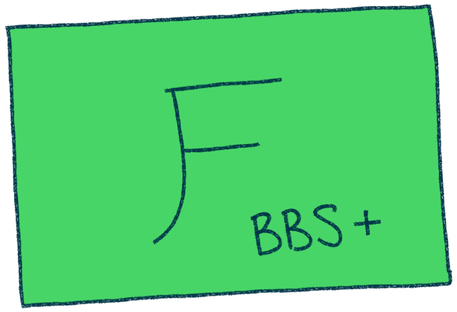


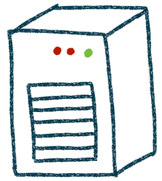


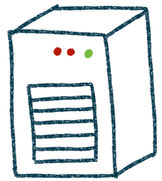
BBS+ Functionality for Strong Blind Signing

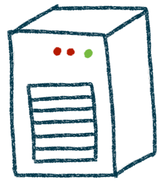
Strongly blind: message and signature are hidden

Signer can issue a signature without seeing contents of message

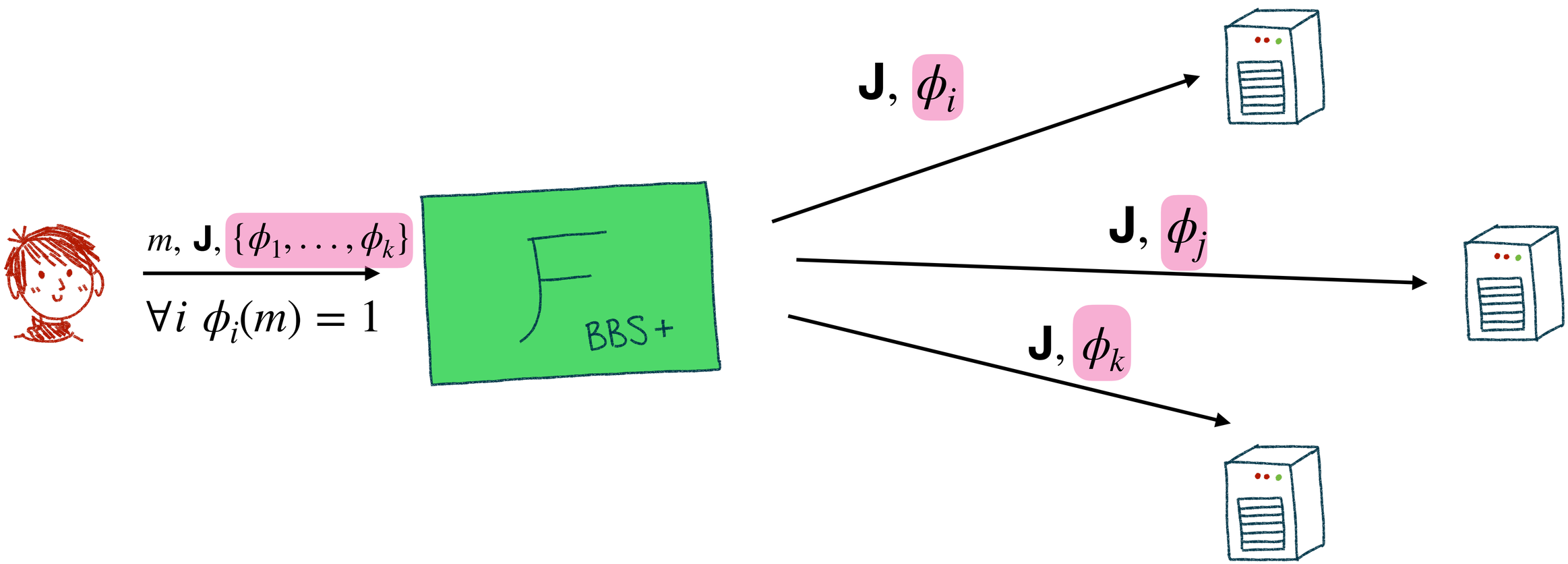




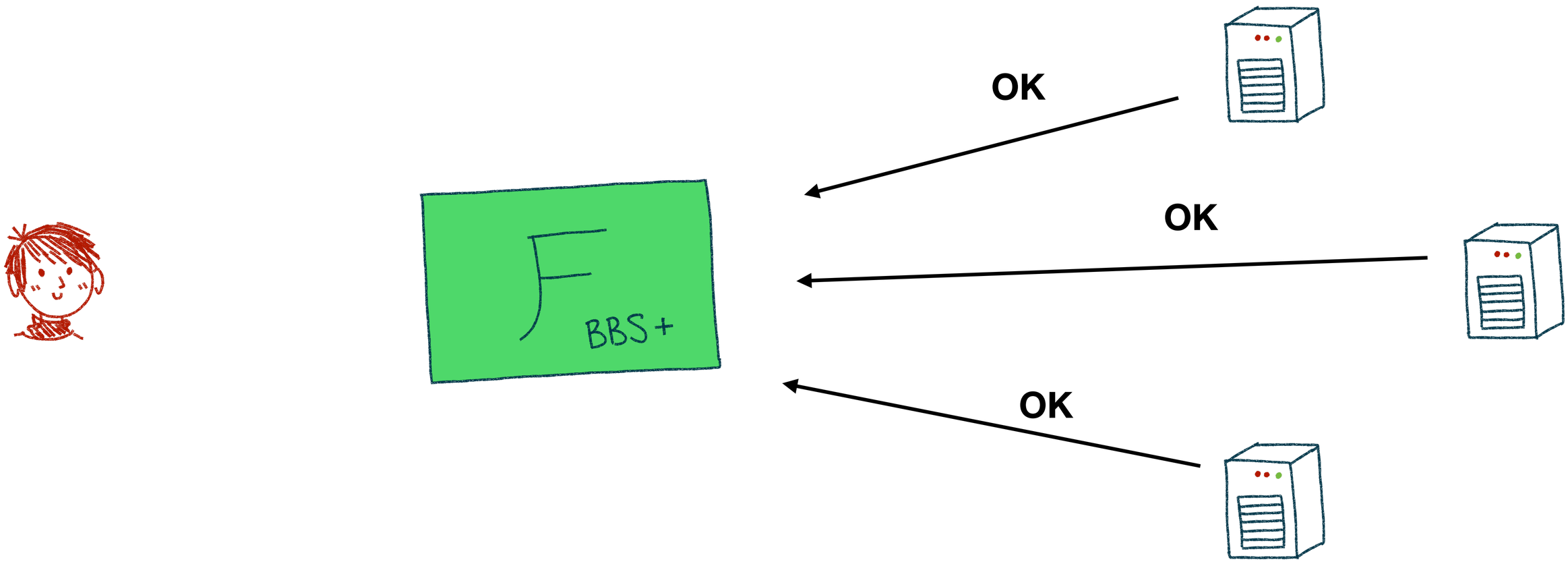




BBS+ Functionality for Weak Partially-Blind Signing



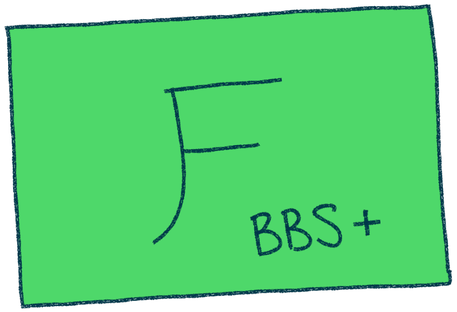
BBS+ Functionality for Weak Partially-Blind Signing

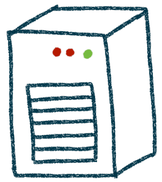


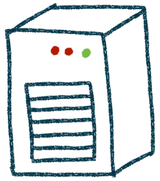
BBS+ Functionality for Weak Partially-Blind Signing

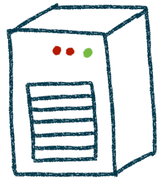
(A, e, s)

$(A, e, s) \leftarrow \mathsf{BBS} + \mathsf{Sign}(x, m)$









Weak Partially Blind Signing with BBS+

$$sk = x$$
$$pk = (g_2^x, h_1, \ldots, h_{L+1})$$

$$\text{BBS} + \text{Sign}(sk, m \in \mathbb{Z}_p^L) :$$

$$e, s \leftarrow \mathbb{Z}_p$$

$$A = (B)^{1/(x+e)}$$

$$\sigma := (A, e, s)$$

$$B = B(pk, m, s)$$

Weak Partially Blind Signing Protocol

$s_0 \leftarrow Z_p$

$B' = B'(m, s_0)$

(B', J)

(B', J)

(B', J)

Client Reconstruction

(e, s, R_i, u_i)

(e, s, R_i, u_i)

(e, s, R_i, u_i)

Computes A
Outputs $(A, e, s + s_0)$
if it verifies

Other extensions

- Strong Blindness
- Oblivious VRF (OVRF)
- Proactive security [KMOS21]
- Short Signatures via Tessero and Zhu [TZ2023]
- Okamoto signatures

Strong Blind Signing Protocol

(B', J)

(B', J)

(B', J)

$s_0, e_0 \leftarrow Z_p$

$B' = B'(m, s_0)$

Client Reconstruction

(e, s, R_i', u_i')

(e, s, R_i', u_i')

(e, s, R_i', u_i')

Computes $A' = \sum R_i' / \sum u_i'$

Outputs $(A', e + e_0, s + s_0)$ if it verifies

Oblivious Verifiable Random Function (OVRF)

- A VRF is a function that provides a proof for computation
- Dodis and Yampolskiy VRF [DY05]

$$F_x(e) \mapsto \mathsf{e}(G_1, G_2)/(x + e)$$

- Correctness proof

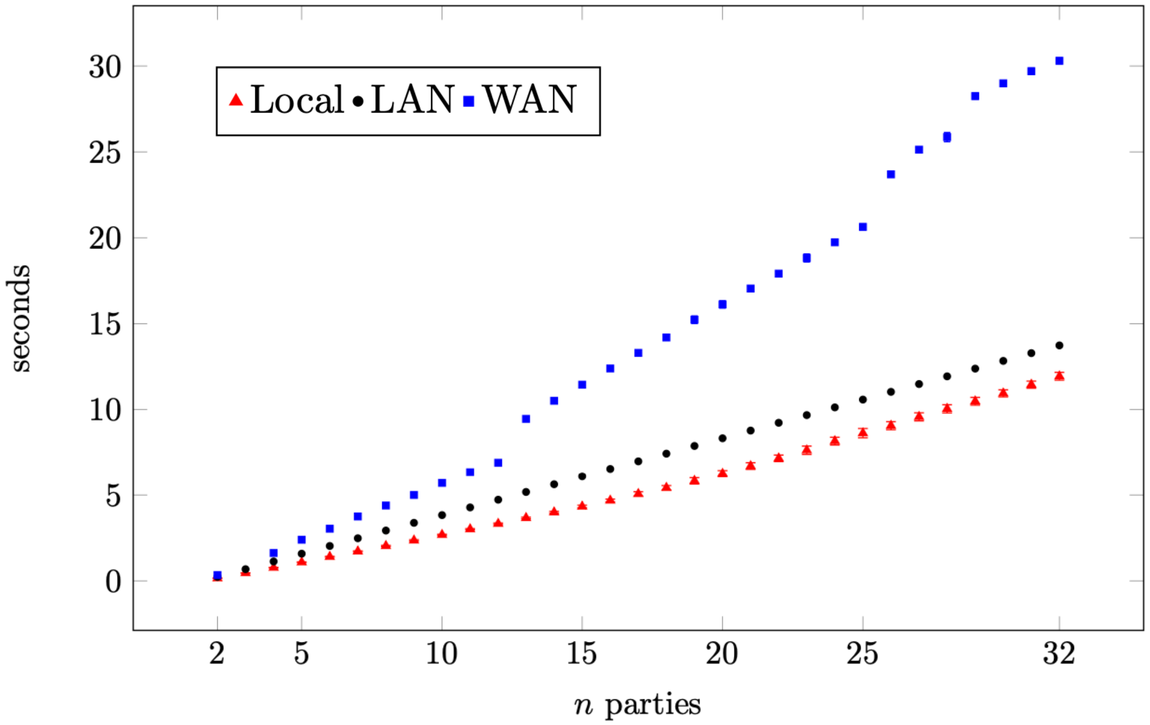
$$\pi = G_1/(x + e)$$

- Essentially computes the VRF on the clients input
- Composable security

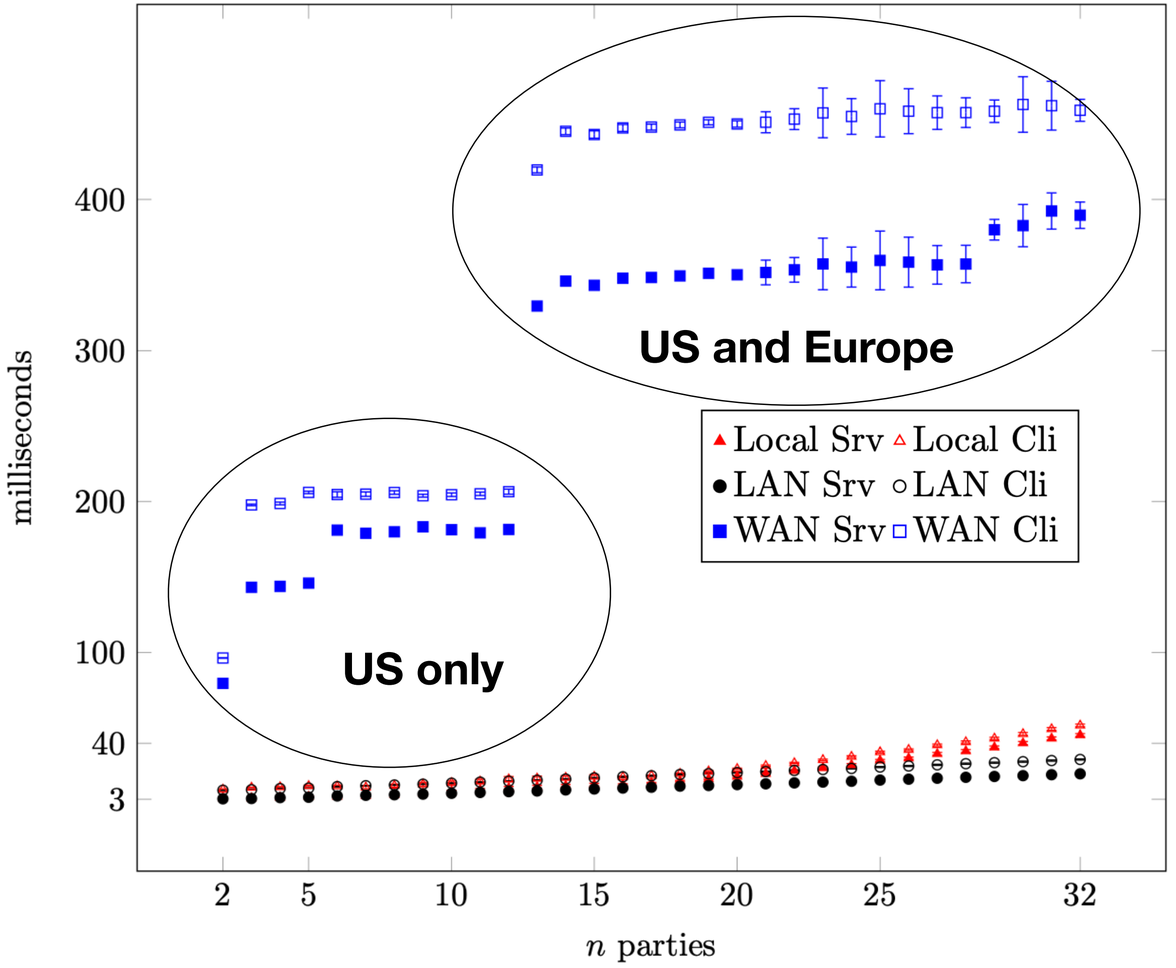
Implementation

- We benchmarked using the BLS12_381 curve
- Blackbox use of OT based multiplier [DKLs18, Gilboa88]
  - Cuts down on computation
  - Optimize wall clock time
- LAN and WAN benchmarked using Google Cloud
- Two setting for WAN
    - ≤ 12 parties servers in US
    - > 12 parties servers in US and Europe

Setup and Signing Times

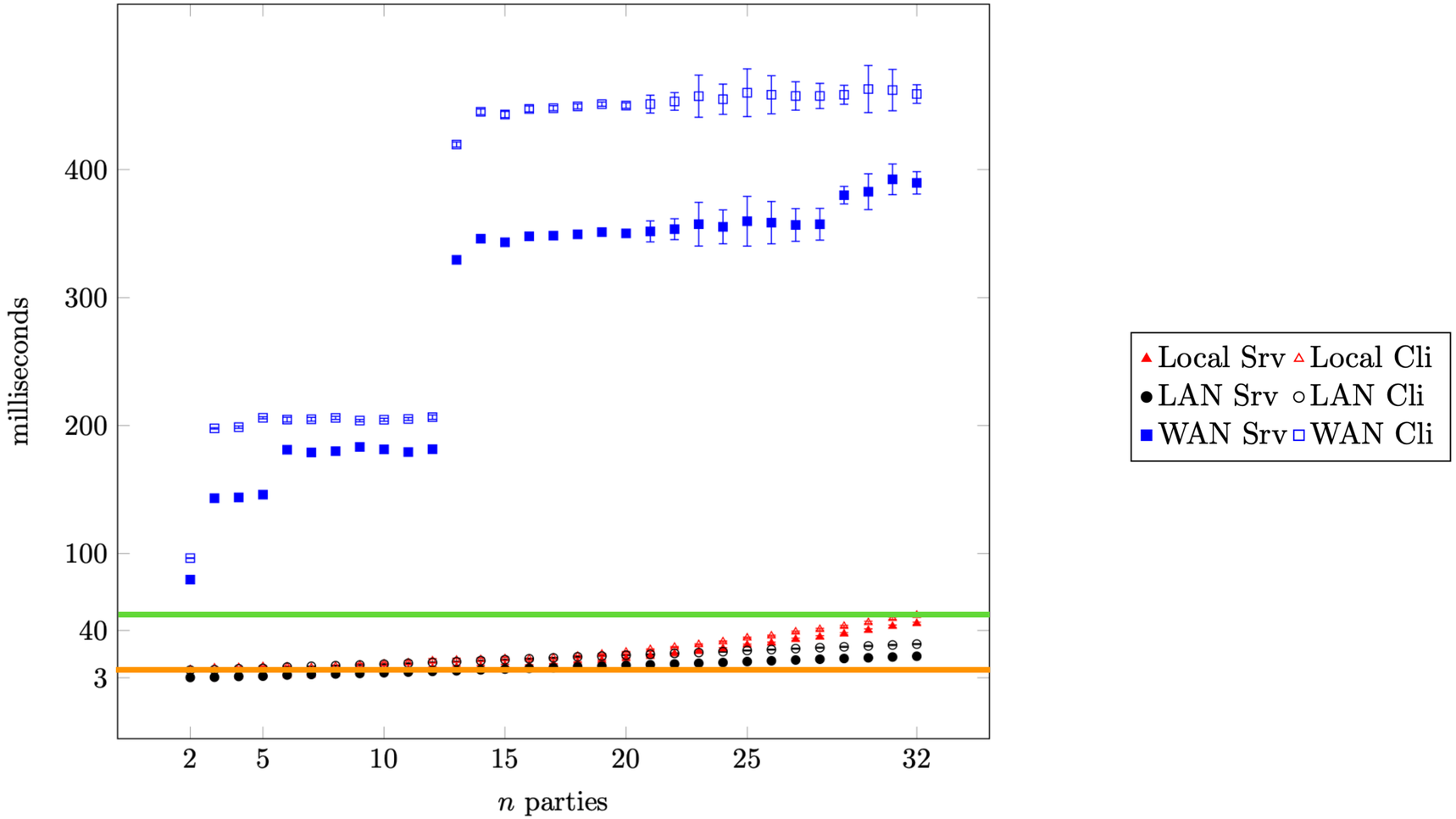


Setup times for n-of-n averaged over 150+ runs



Signing times for n-of-n averaged over 150+ runs

Setup and Signing Times



Signing times for n-of-n averaged over 150+ runs

Where does BBS+ fit among Threshold ?

Schnorr(sk, m) :

$k \leftarrow \mathbb{Z}_q$

$R = k \cdot G$

$e = H(R \| m)$

$s = k - \mathsf{sk} \cdot e$

$\sigma = (s, R)$

output σ

Linear

BBS+(sk, m) :

$e, s \leftarrow \mathbb{Z}_q$

$A = \dfrac{G + \cdots}{x + e}$

$\sigma = (A, e, s)$

output σ

1 mult

ECDSA(sk, m) :

$k \leftarrow \mathbb{Z}_q$

$R = k \cdot G$

$e = H(m)$

$s = \dfrac{e + \mathsf{sk} \cdot r_x}{k}$

output $\sigma = (s, R)$

2 mults +
many checks

EdDSA(sk, m) :

$k = F(\mathsf{sk}, m)$

$R = k \cdot G$

$e = H(R \| m)$

$s = k - \mathsf{sk} \cdot e$

$\sigma = (s, R)$

output σ

MPC or skip?

Where does BBS+ fit among Threshold ?

Schnorr(sk, m) :

$k \leftarrow \mathbb{Z}_q$

$R = k \cdot G$

$e = H(R \| m)$

$s = k - \mathsf{sk} \cdot e$

$\sigma = (s, R)$

output σ

Linear

BBS+(sk, m) :

$e, s \leftarrow \mathbb{Z}_q$

$A = \frac{G + \cdots}{x + e}$

$\sigma = (A, e, s)$

output σ

1 mult

ECDSA(sk, m) :

$k \leftarrow \mathbb{Z}_q$

$R = k \cdot G$

$e = H(m)$

$s = \frac{e + \mathsf{sk} \cdot r_x}{k}$

output $\sigma = (s, R)$

2 mults +
many checks

EdDSA(sk, m) :

$k = F(\mathsf{sk}, m)$

$R = k \cdot G$

$e = H(R \| m)$

$s = k - \mathsf{sk} \cdot e$

$\sigma = (s, R)$

output σ

MPC or skip?

Where does BBS+ fit among Threshold ?

Schnorr(sk, m) :

$k \leftarrow \mathbb{Z}_q$

$R = k \cdot G$

$e = H(R \| m)$

$s = k - \mathsf{sk} \cdot e$

$\sigma = (s, R)$

output σ

Linear

BBS+(sk, m) :

$e, s \leftarrow \mathbb{Z}_q$

$A = \dfrac{G + \cdots}{x + e}$

$\sigma = (A, e, s)$

output σ

1 mult

ECDSA(sk, m) :

$k \leftarrow \mathbb{Z}_q$

$R = k \cdot G$

$e = H(m)$

$s = \dfrac{e + \mathsf{sk} \cdot r_x}{k}$

output $\sigma = (s, R)$

2 mults +
many checks

EdDSA(sk, m) :

$k = F(\mathsf{sk}, m)$

$R = k \cdot G$

$e = H(R \| m)$

$s = k - \mathsf{sk} \cdot e$

$\sigma = (s, R)$

output σ

MPC or skip?

Where does BBS+ fit among Threshold ?

Schnorr(sk, m) :

$k \leftarrow \mathbb{Z}_q$

$R = k \cdot G$

$e = H(R \| m)$

$s = k - \mathsf{sk} \cdot e$

$\sigma = (s, R)$

output σ

Linear

BBS+(sk, m) :

$e, s \leftarrow \mathbb{Z}_q$

$A = \dfrac{G + \cdots}{x + e}$

$\sigma = (A, e, s)$

output σ

1 mult

ECDSA(sk, m) :

$k \leftarrow \mathbb{Z}_q$

$R = k \cdot G$

$e = H(m)$

$s = \dfrac{e + \mathsf{sk} \cdot r_x}{k}$

output $\sigma = (s, R)$

2 mults +
many checks

EdDSA(sk, m) :

$k = F(\mathsf{sk}, m)$

$R = k \cdot G$

$e = H(R \| m)$

$s = k - \mathsf{sk} \cdot e$

$\sigma = (s, R)$

output σ

MPC or skip?

Where does BBS+ fit among Threshold ?

Schnorr(sk, m) :

$k \leftarrow \mathbb{Z}_q$

$R = k \cdot G$

$e = H(R \| m)$

$s = k - \mathsf{sk} \cdot e$

$\sigma = (s, R)$

output σ

Linear

BBS+(sk, m) :

$e, s \leftarrow \mathbb{Z}_q$

$A = \frac{G + \cdots}{x + e}$

$\sigma = (A, e, s)$

output σ

1 mult

ECDSA(sk, m) :

$k \leftarrow \mathbb{Z}_q$

$R = k \cdot G$

$e = H(m)$

$s = \frac{e + \mathsf{sk} \cdot r_x}{k}$

output $\sigma = (s, R)$

2 mults +
many checks

EdDSA(sk, m) :

$k = F(\mathsf{sk}, m)$

$R = k \cdot G$

$e = H(R \| m)$

$s = k - \mathsf{sk} \cdot e$

$\sigma = (s, R)$

output σ

MPC or skip?

Tak!

ePrint 2023/602