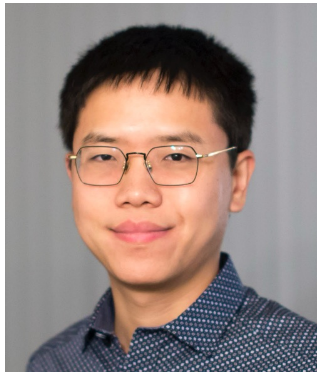
Powers of Tau in Asynchrony


Sourav Das


Zhuolun Xiang


Ling Ren

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

APTOS

souravd2@Illinois.edu

Problem Definitions

q-Strong Diffie-Hellman (q-SDH) Assumption

q-Strong Diffie-Hellman (q-SDH) Assumption

Challenger

Adversary $\mathcal{A}$

q-Strong Diffie-Hellman (q-SDH) Assumption

Challenger

Adversary $\mathcal{A}$

q-Strong Diffie-Hellman (q-SDH) Assumption

1. Elliptic curve group $\mathbb{G}$

Challenger

Adversary $\mathcal{A}$

q-Strong Diffie-Hellman (q-SDH) Assumption

1. Elliptic curve group $\mathbb{G}$
2. Scalar field $\mathbb{F}$

Challenger

Adversary $\mathcal{A}$

q-Strong Diffie-Hellman (q-SDH) Assumption

1. Elliptic curve group $\mathbb{G}$
2. Scalar field $\mathbb{F}$
3. Generator $\mathbb{G} \in \mathbb{G}$

Challenger

Adversary $\mathcal{A}$

q-Strong Diffie-Hellman (q-SDH) Assumption

1. Elliptic curve group $\mathbb{G}$
2. Scalar field $\mathbb{F}$
3. Generator $\mathrm{G} \in \mathbb{G}$

Challenger

$$\xleftarrow{\quad q \in \mathbb{N} \quad}$$

Adversary $\mathcal{A}$

q-Strong Diffie-Hellman (q-SDH) Assumption

1. Elliptic curve group $\mathbb{G}$
2. Scalar field $\mathbb{F}$
3. Generator $\mathrm{G} \in \mathbb{G}$

Challenger

$\tau \leftarrow \mathbb{F}$

$$\xleftarrow{\quad q \in \mathbb{N} \quad}$$

Adversary $\mathcal{A}$

q-Strong Diffie-Hellman (q-SDH) Assumption

1. Elliptic curve group $\mathbb{G}$
2. Scalar field $\mathbb{F}$
3. Generator $\mathrm{G} \in \mathbb{G}$

Challenger

$\tau \leftarrow \mathbb{F}$

$$\xleftarrow{\quad q \in \mathbb{N} \quad}$$

$$\xrightarrow{\quad \mathrm{G}, \tau\mathrm{G}, \tau^2\mathrm{G}, \ldots, \tau^q\mathrm{G} \quad}$$

Adversary $\mathcal{A}$

q-Strong Diffie-Hellman (q-SDH) Assumption

1. Elliptic curve group $\mathbb{G}$
2. Scalar field $\mathbb{F}$
3. Generator $\mathrm{G} \in \mathbb{G}$

Challenger

$\tau \leftarrow \mathbb{F}$

Adversary $\mathcal{A}$

$q \in \mathbb{N}$ ($\leftarrow$)

$\mathrm{G}, \tau\mathrm{G}, \tau^2\mathrm{G}, \ldots, \tau^q\mathrm{G}$ ($\rightarrow$)

$c, (\tau + c)^{-1} \cdot \mathrm{G}$ ($\leftarrow$)

q-Strong Diffie-Hellman (q-SDH) Assumption

1. Elliptic curve group $\mathbb{G}$
2. Scalar field $\mathbb{F}$
3. Generator $\mathrm{G} \in \mathbb{G}$

Challenger

$\tau \leftarrow \mathbb{F}$

Adversary $\mathcal{A}$

$$\xleftarrow{\quad q \in \mathbb{N} \quad}$$

$$\xrightarrow{\quad \mathrm{G}, \tau\mathrm{G}, \tau^2\mathrm{G}, \ldots, \tau^q\mathrm{G} \quad}$$

$$\xleftarrow{\quad c, (\tau + c)^{-1} \cdot \mathrm{G} \quad}$$

Adversary wins: if $c + \tau \neq 0$ and $(\tau + c)^{-1} \cdot \mathrm{G}$ is well formed

q-Strong Diffie-Hellman (q-SDH) Assumption

1. Elliptic curve group $\mathbb{G}$
2. Scalar field $\mathbb{F}$
3. Generator $\mathrm{G} \in \mathbb{G}$

Challenger

$\tau \leftarrow \mathbb{F}$

Adversary $\mathcal{A}$

$q \in \mathbb{N}$ (Adversary → Challenger)

$\mathrm{G}, \tau\mathrm{G}, \tau^2\mathrm{G}, \ldots, \tau^q\mathrm{G}$ (Challenger → Adversary)

$c, (\tau + c)^{-1} \cdot \mathrm{G}$ (Adversary → Challenger)

Adversary wins: if $c + \tau \neq 0$ and $(\tau + c)^{-1} \cdot \mathrm{G}$ is well formed

q-SDH assumptions says adversary wins with negligible probability

q-SDH parameters (aka Powers of Tau)

1. Elliptic curve group $\mathbb{G}$
2. Scalar field $\mathbb{F}$
3. Generator $\mathrm{G} \in \mathbb{G}$

Challenger

$\tau \leftarrow \mathbb{F}$

Adversary $\mathcal{A}$

$\xleftarrow{\quad q \in \mathbb{N} \quad}$

$\xrightarrow{\quad \boxed{\mathrm{G}, \tau\mathrm{G}, \tau^2\mathrm{G}, \ldots, \tau^q\mathrm{G}} \quad}$

$\xleftarrow{\quad c, (\tau + c)^{-1} \cdot \mathrm{G} \quad}$

Adversary wins: if $c + \tau \neq 0$ and $(\tau + c)^{-1} \cdot \mathrm{G}$ is well formed

q-SDH assumptions says adversary wins with negligible probability

Applications of q-SDH parameters

Applications of q-SDH parameters

- Short signatures

Applications of q-SDH parameters

- Short signatures
- Cryptographic Accumulators

Applications of q-SDH parameters

- Short signatures
- Cryptographic Accumulators
- Vector commitments

Applications of q-SDH parameters

- Short signatures
- Cryptographic Accumulators
- Vector commitments
- Constant size polynomial commitments

Applications of q-SDH parameters

- Short signatures
- Cryptographic Accumulators
- Vector commitments
- Constant size polynomial commitments
  - SNARKs

Applications of q-SDH parameters

- Short signatures
- Cryptographic Accumulators
- Vector commitments
- Constant size polynomial commitments
    - SNARKs
    - Verifiable Secret Sharing

Applications of q-SDH parameters

- Short signatures
- Cryptographic Accumulators
- Vector commitments
- Constant size polynomial commitments
  - SNARKs
  - Verifiable Secret Sharing
  - Randomness Beacon

Our Goal

Our Goal

MPC protocol to generate of Powers of Tau in an asynchronous network

Our Goal

MPC protocol to generate of Powers of Tau in an asynchronous network

System model:

Our Goal

MPC protocol to generate of Powers of Tau in an asynchronous network

System model:

- $n \geq 3t + 1$ nodes among which up to t are corrupt

Our Goal

MPC protocol to generate of Powers of Tau in an asynchronous network

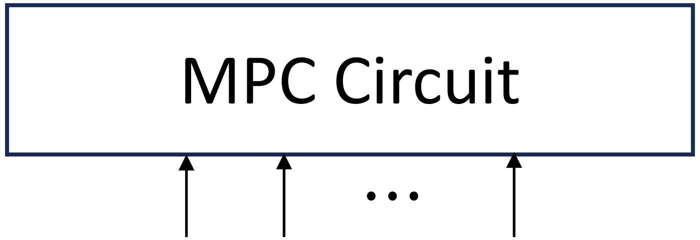
System model:

- $n \geq 3t + 1$ nodes among which up to t are corrupt
- Asynchronous network:
  - Message delays could be arbitrary

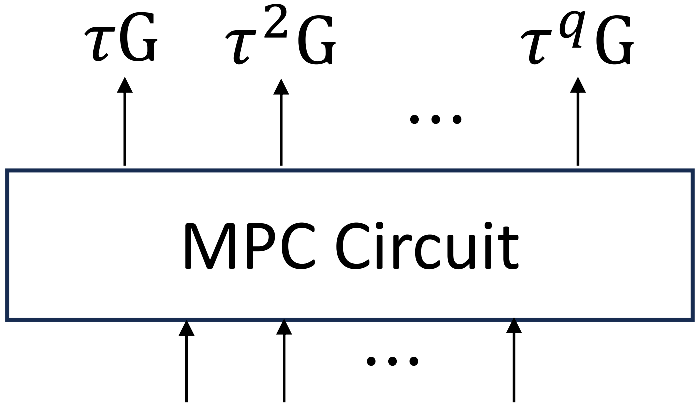
Related Works

Generic MPC based approach

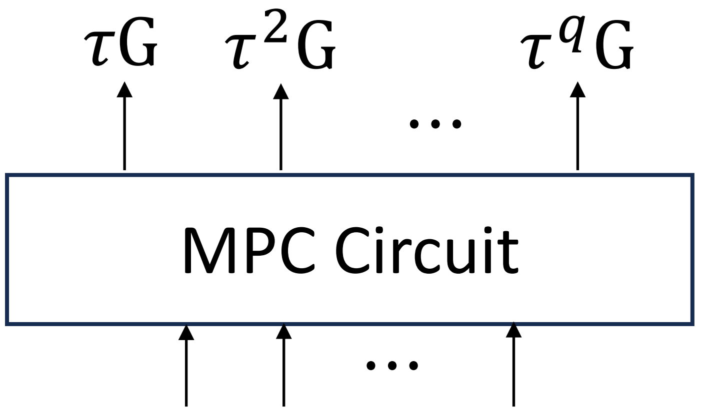
Generic MPC based approach



Generic MPC based approach

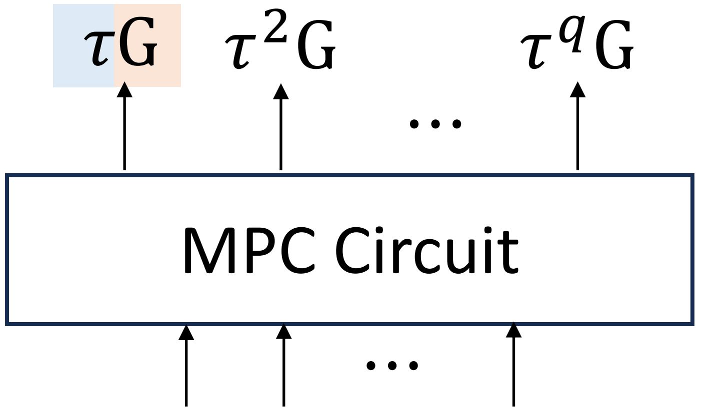


Generic MPC based approach



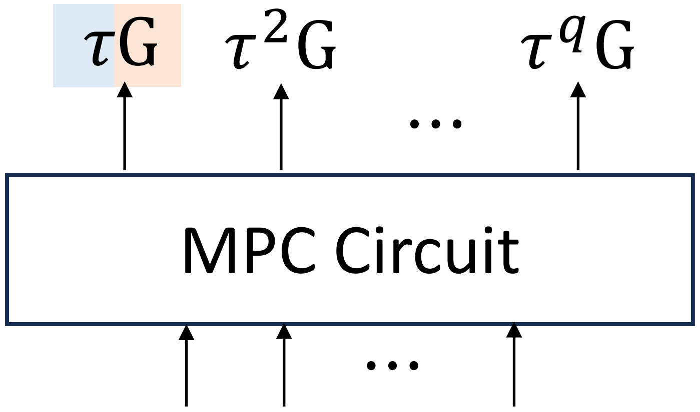
- MPC over both field $\mathbb{F}$ and group $\mathbb{G}$

Generic MPC based approach

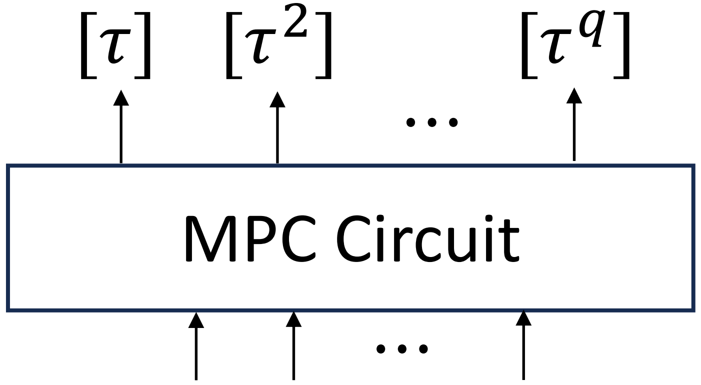


- MPC over both field $\mathbb{F}$ and group $\mathbb{G}$

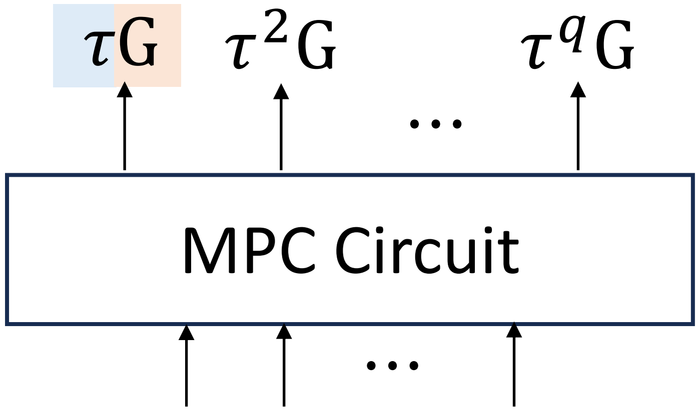
Generic MPC based approach



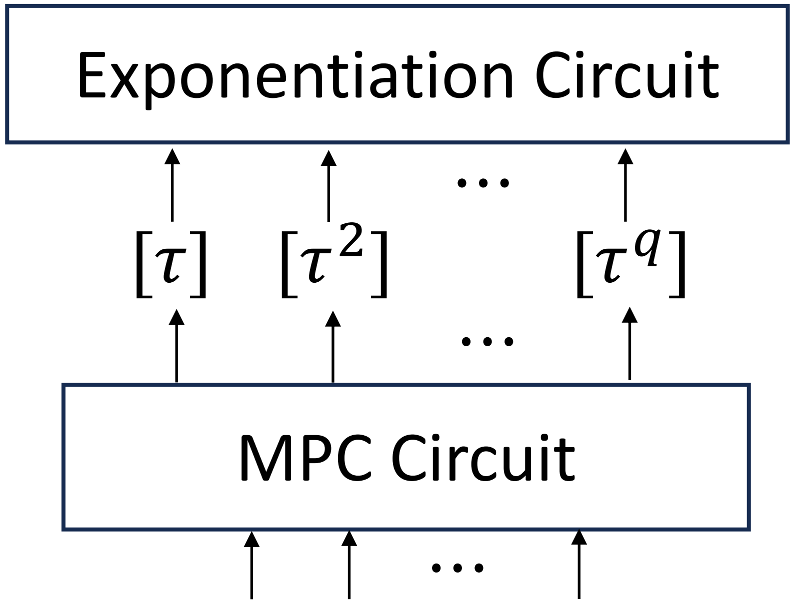
- MPC over both field $\mathbb{F}$ and group $\mathbb{G}$



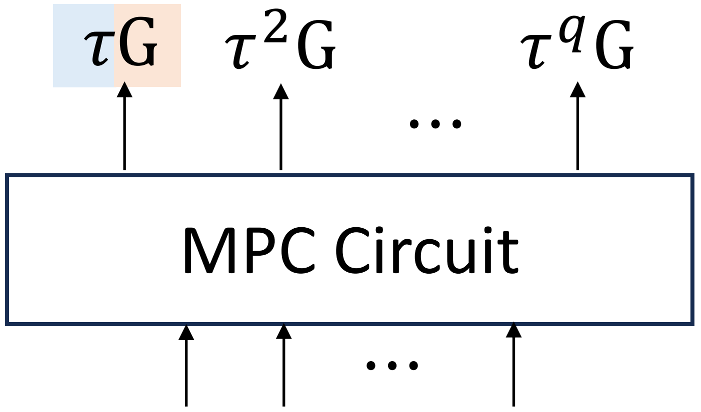
Generic MPC based approach



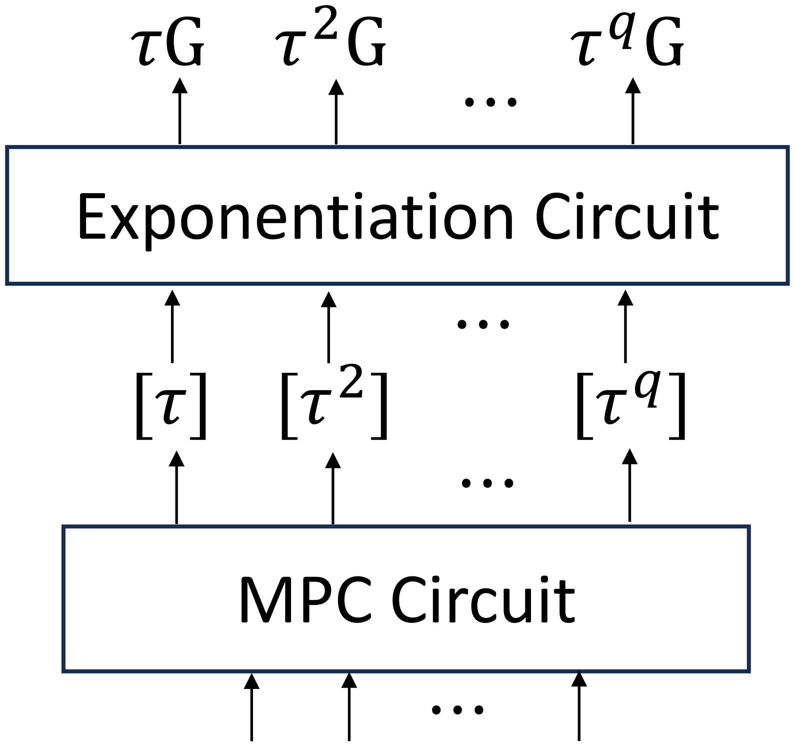
- MPC over both field $\mathbb{F}$ and group $\mathbb{G}$



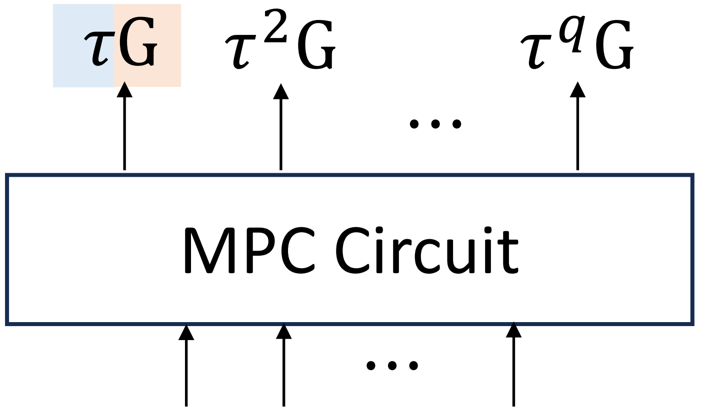
Generic MPC based approach



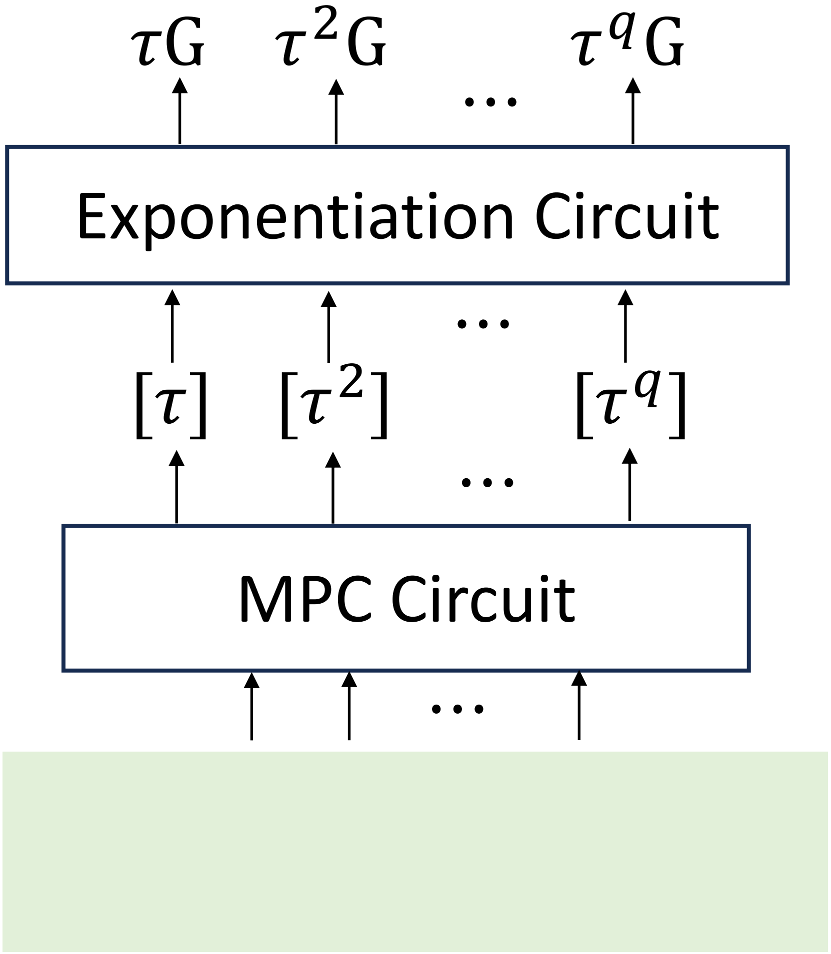
- MPC over both field $\mathbb{F}$ and group $\mathbb{G}$



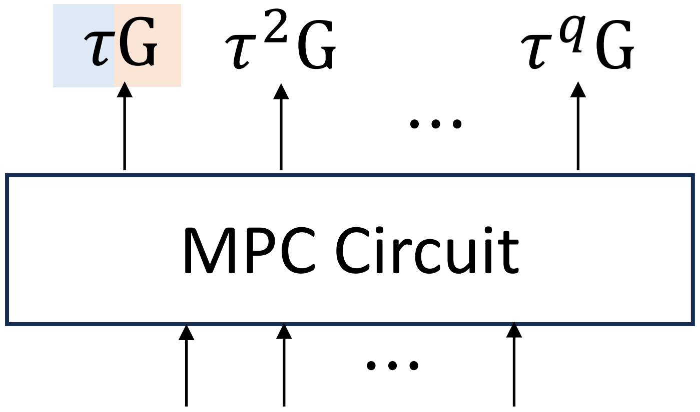
Generic MPC based approach



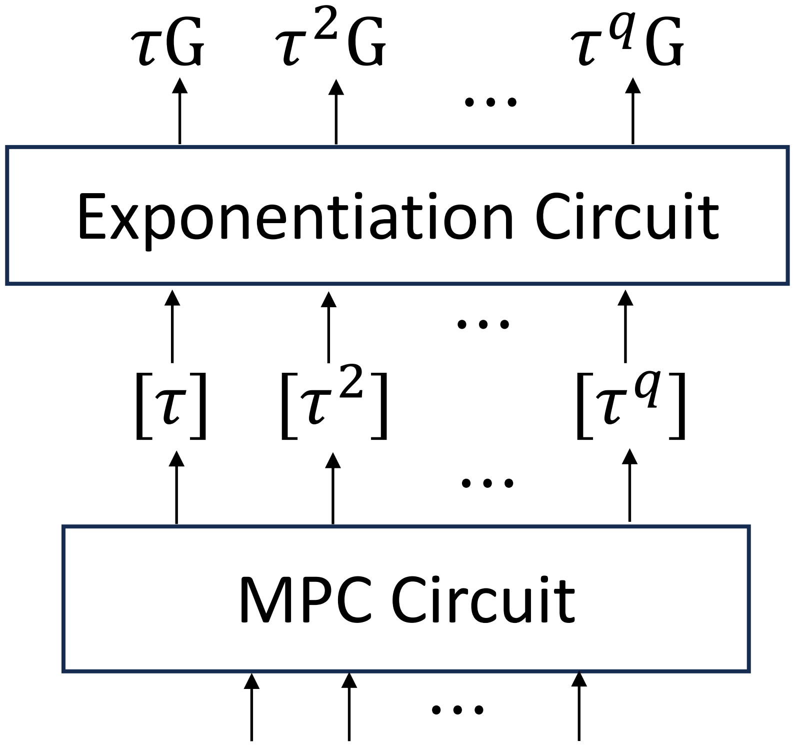
- MPC over both field $\mathbb{F}$ and group $\mathbb{G}$



Generic MPC based approach

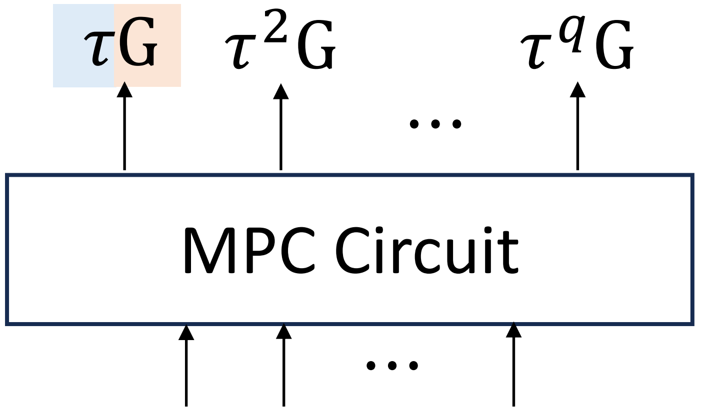


- MPC over both field $\mathbb{F}$ and group $\mathbb{G}$

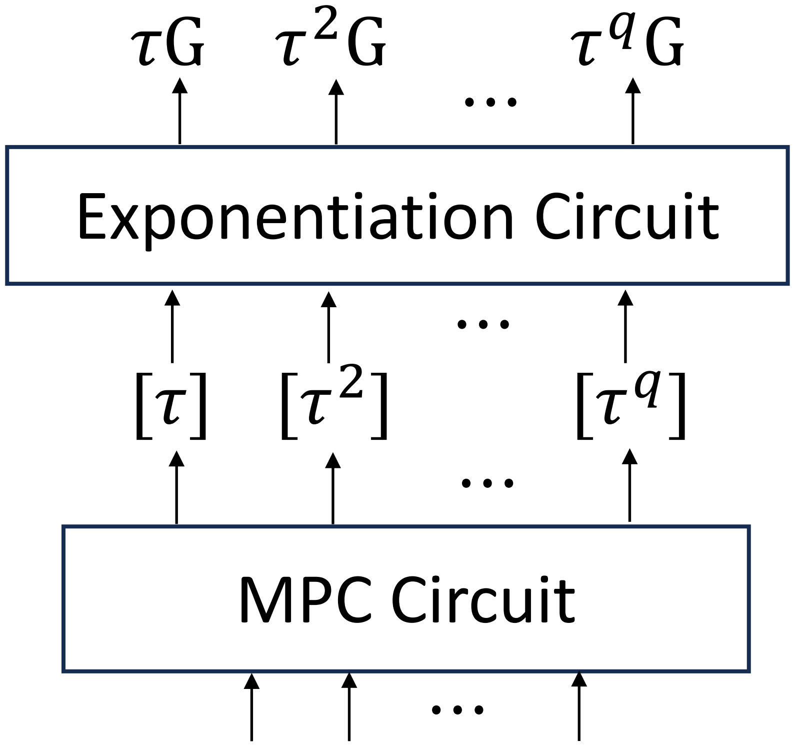


+ MPC over only field $\mathbb{F}$

Generic MPC based approach

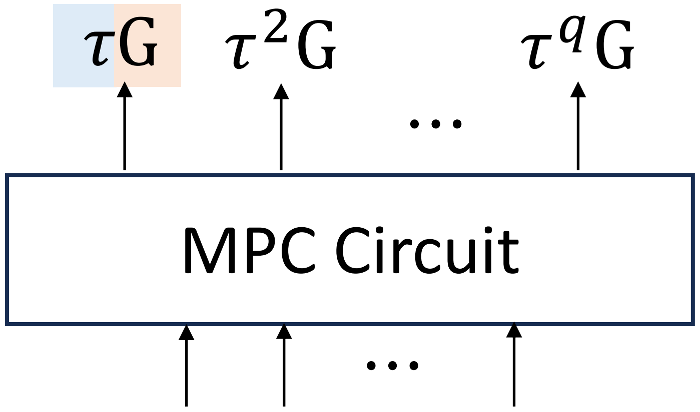


- MPC over both field $\mathbb{F}$ and group $\mathbb{G}$

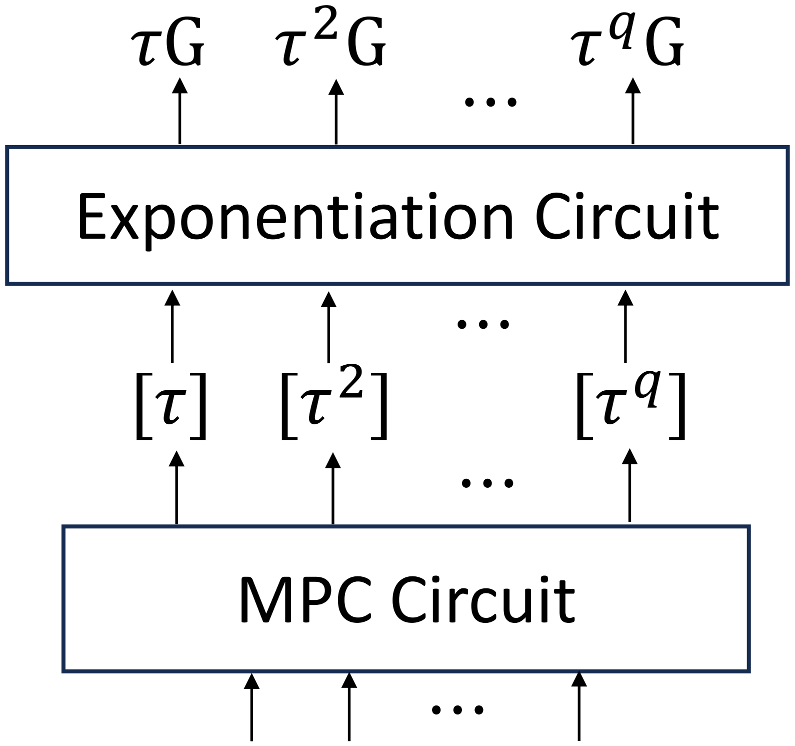


+ MPC over only field $\mathbb{F}$
+ $O(\log q)$ rounds

Generic MPC based approach



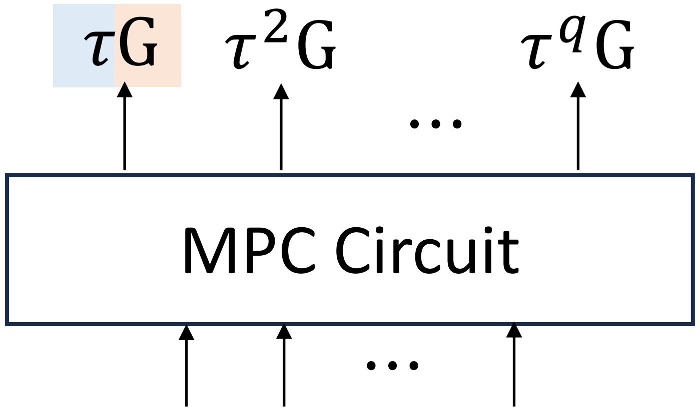
- MPC over both field $\mathbb{F}$ and group $\mathbb{G}$



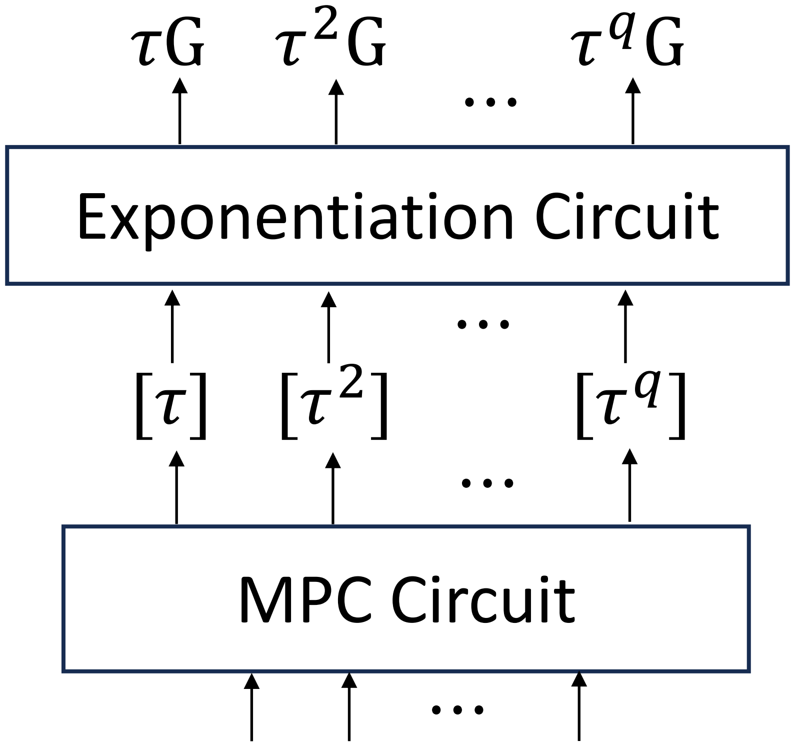
+ MPC over only field $\mathbb{F}$
+ $O(\log q)$ rounds

- $O(q)$ multiplication units

Generic MPC based approach



- MPC over both field $\mathbb{F}$ and group $\mathbb{G}$



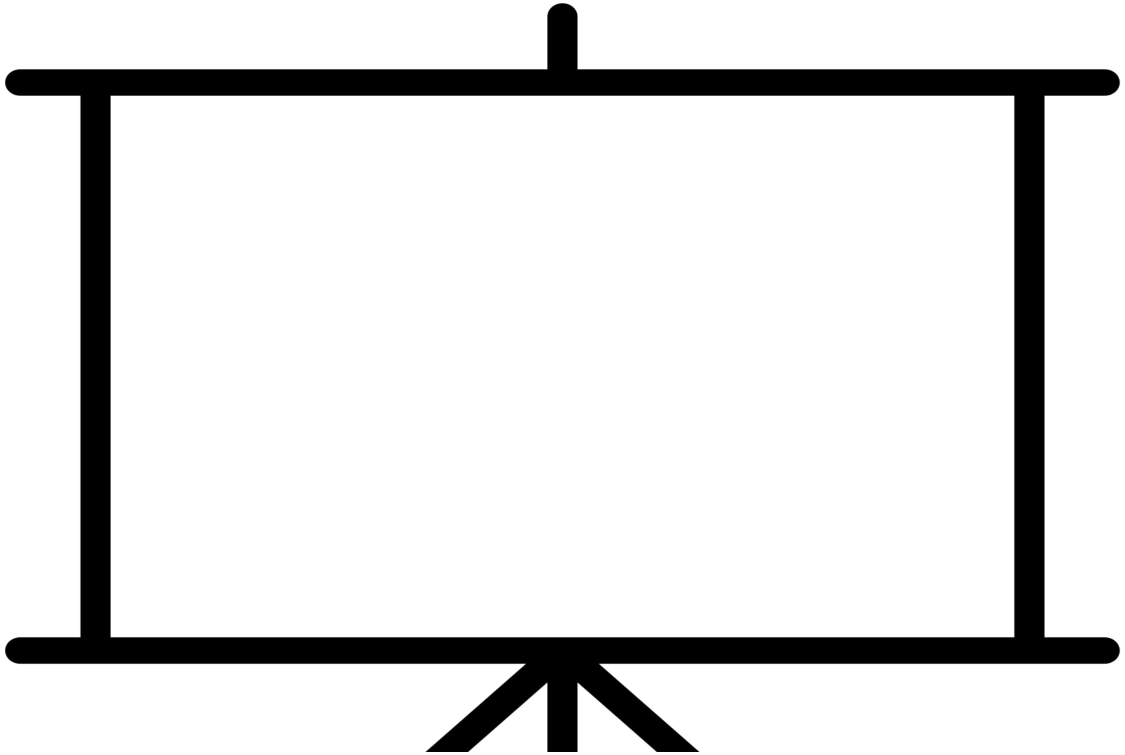
+ MPC over only field $\mathbb{F}$
+ $O(\log q)$ rounds

- $O(q)$ multiplication units

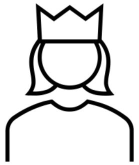
Multiplication units are expensive, per party $\Omega(nq)$ communication costs

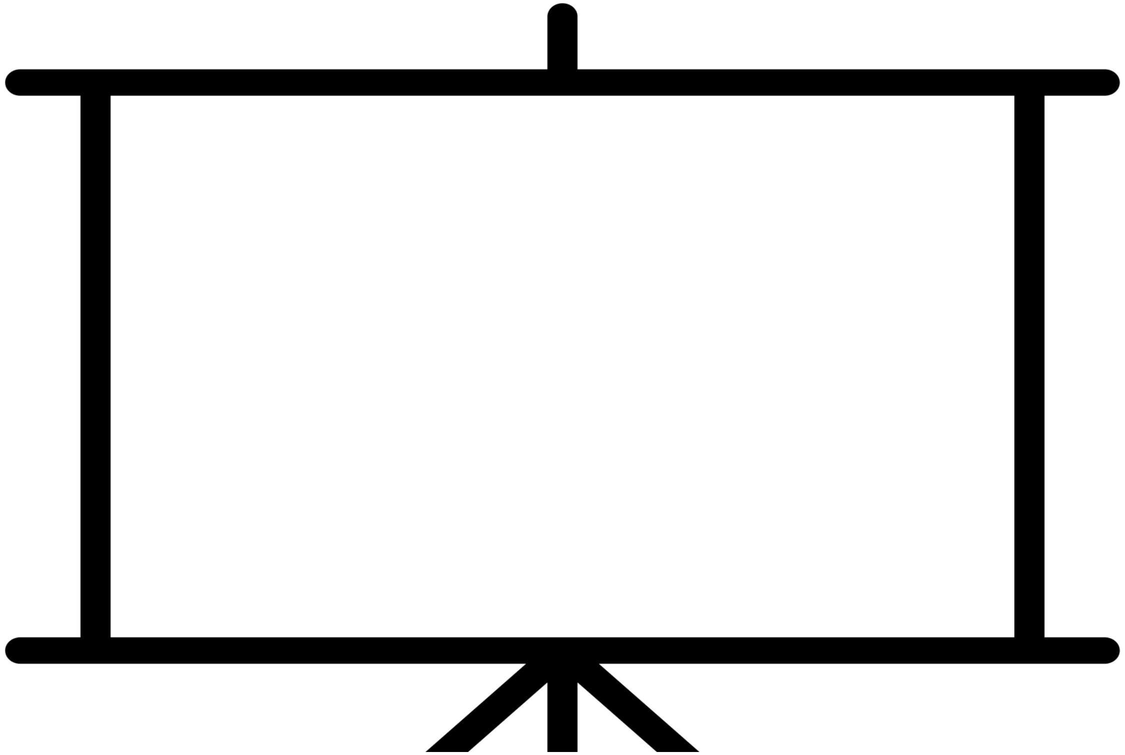
Round Robin approach

Round Robin approach

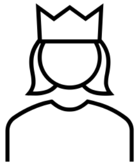


Round Robin approach

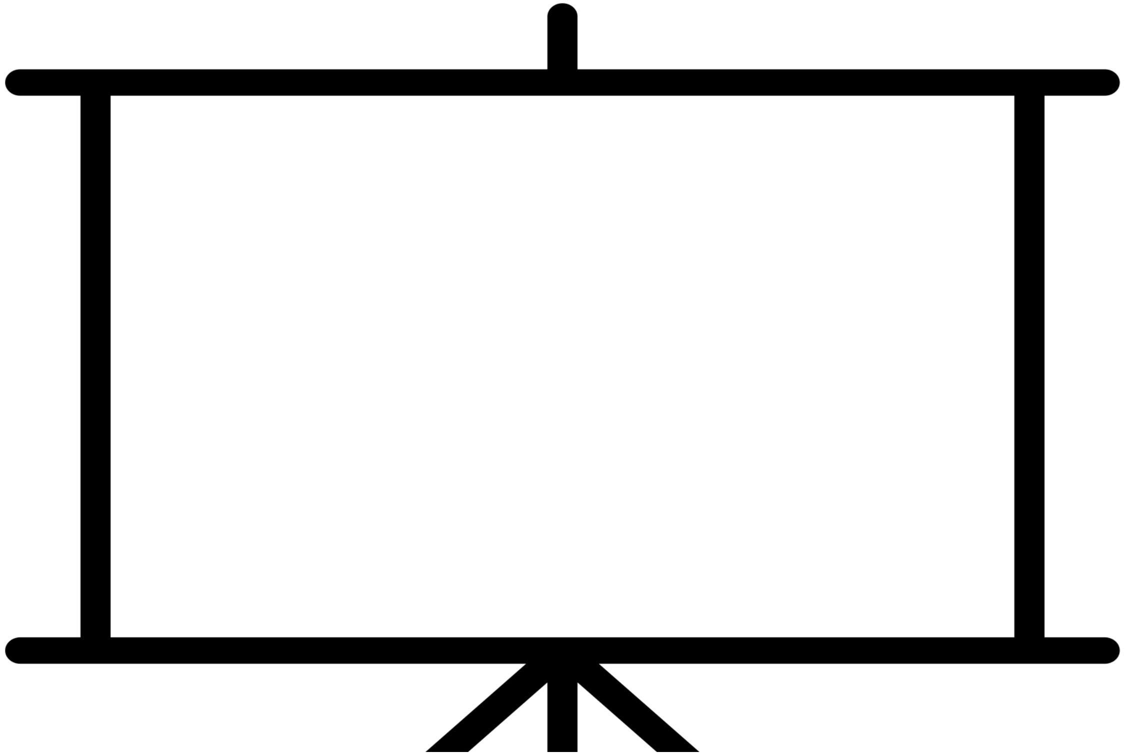




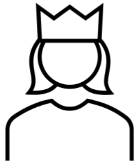
Round Robin approach



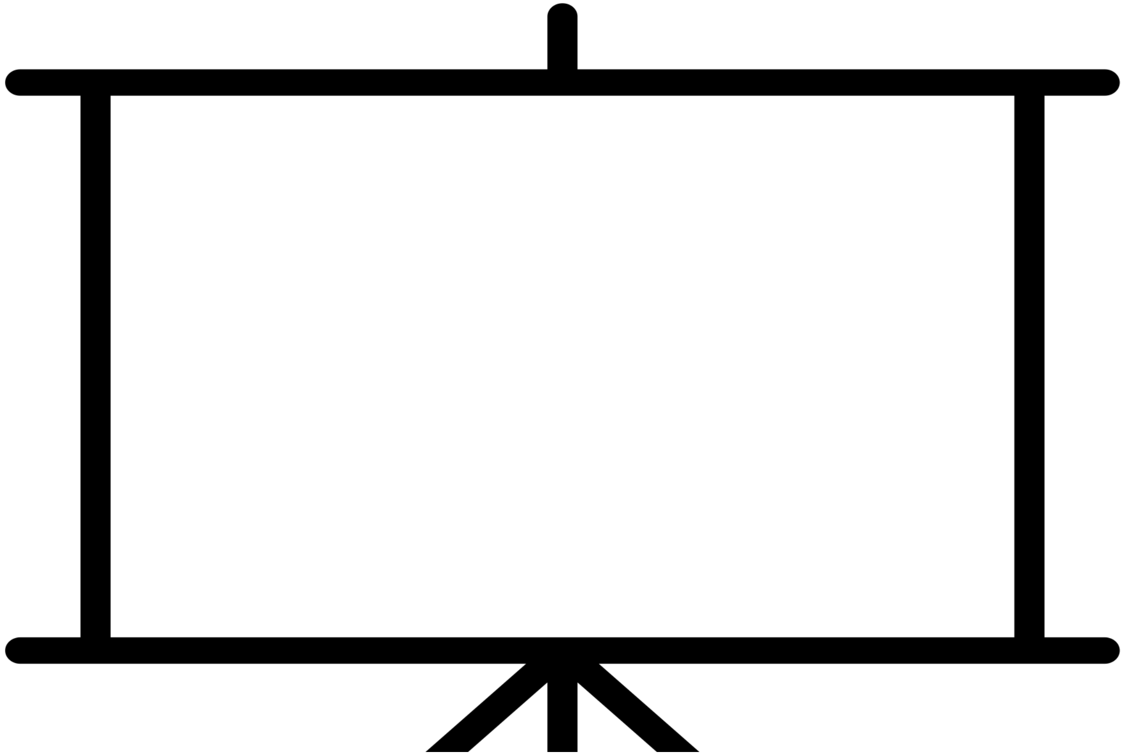
- Sample $\tau_1 \leftarrow \mathbb{F}$



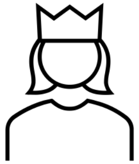
Round Robin approach



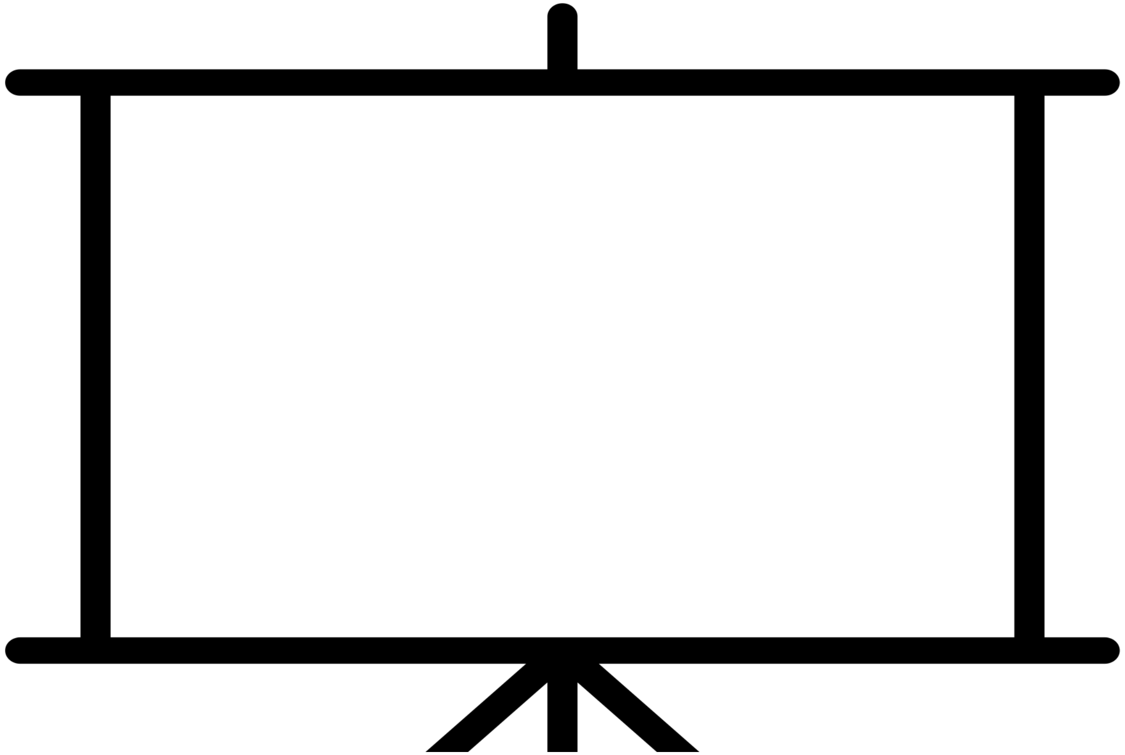
- Sample $\tau_1 \leftarrow \mathbb{F}$
- Compute $\mathrm{G}, \tau_1 \mathrm{G}, \tau_1^2 \mathrm{G}, \ldots, \tau_1^q \mathrm{G}$



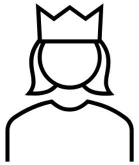
Round Robin approach



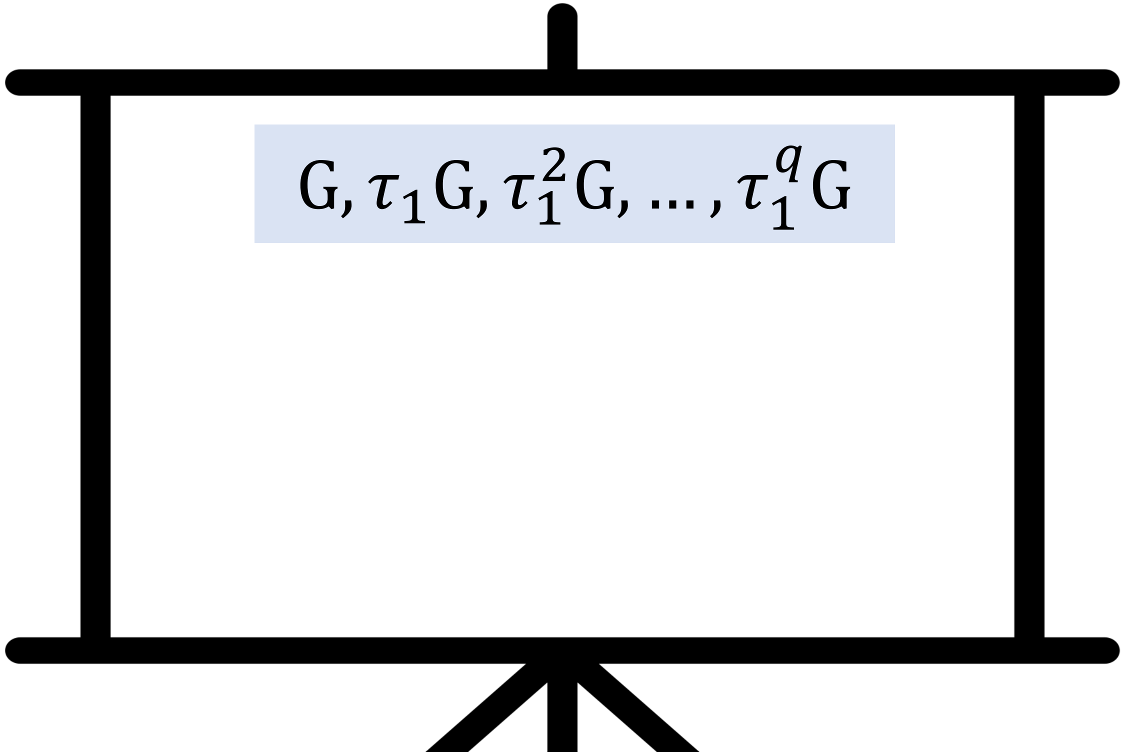
- Sample $\tau_1 \leftarrow \mathbb{F}$
- Compute $\mathrm{G}, \tau_1 \mathrm{G}, \tau_1^2 \mathrm{G}, \ldots, \tau_1^q \mathrm{G}$
- Post $\mathrm{G}, \tau_1 \mathrm{G}, \tau_1^2 \mathrm{G}, \ldots, \tau_1^q \mathrm{G}$



Round Robin approach

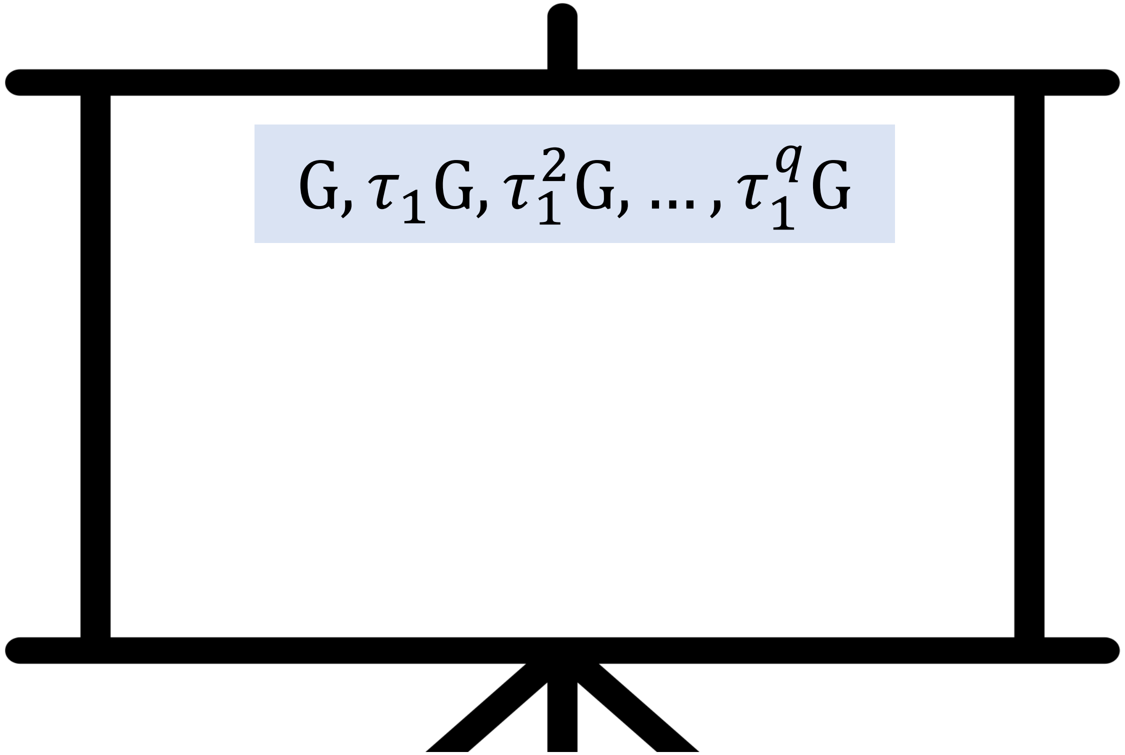


- Sample $\tau_1 \leftarrow \mathbb{F}$
- Compute $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \ldots, \tau_1^q\mathrm{G}$
- Post $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \ldots, \tau_1^q\mathrm{G}$

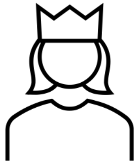


Round Robin approach

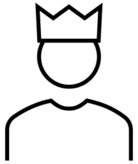
- Sample $\tau_1 \leftarrow \mathbb{F}$
- Compute $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \ldots, \tau_1^q\mathrm{G}$
- Post $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \ldots, \tau_1^q\mathrm{G}$



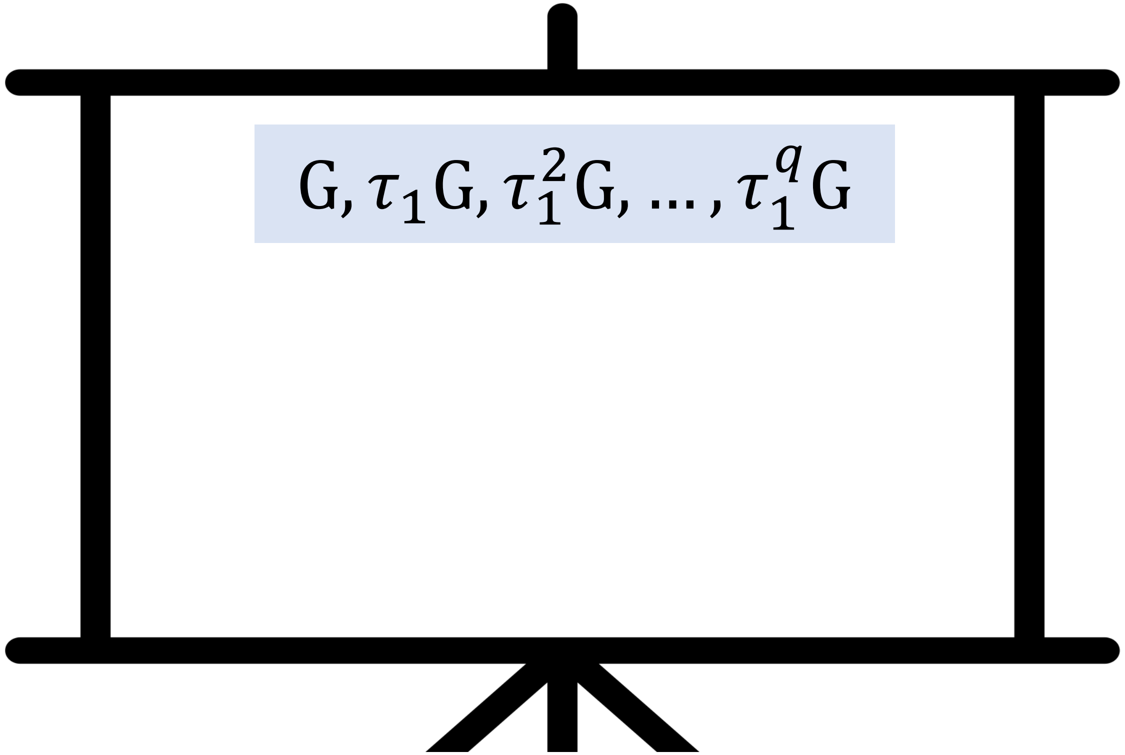
Round Robin approach



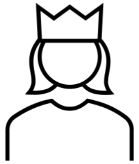
- Sample $\tau_1 \leftarrow \mathbb{F}$
- Compute $\mathrm{G}, \tau_1 \mathrm{G}, \tau_1^2 \mathrm{G}, \ldots, \tau_1^q \mathrm{G}$
- Post $\mathrm{G}, \tau_1 \mathrm{G}, \tau_1^2 \mathrm{G}, \ldots, \tau_1^q \mathrm{G}$



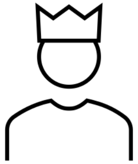
- Download $\mathrm{G}_0, \mathrm{G}_1, \mathrm{G}_2, \ldots, \mathrm{G}_q$



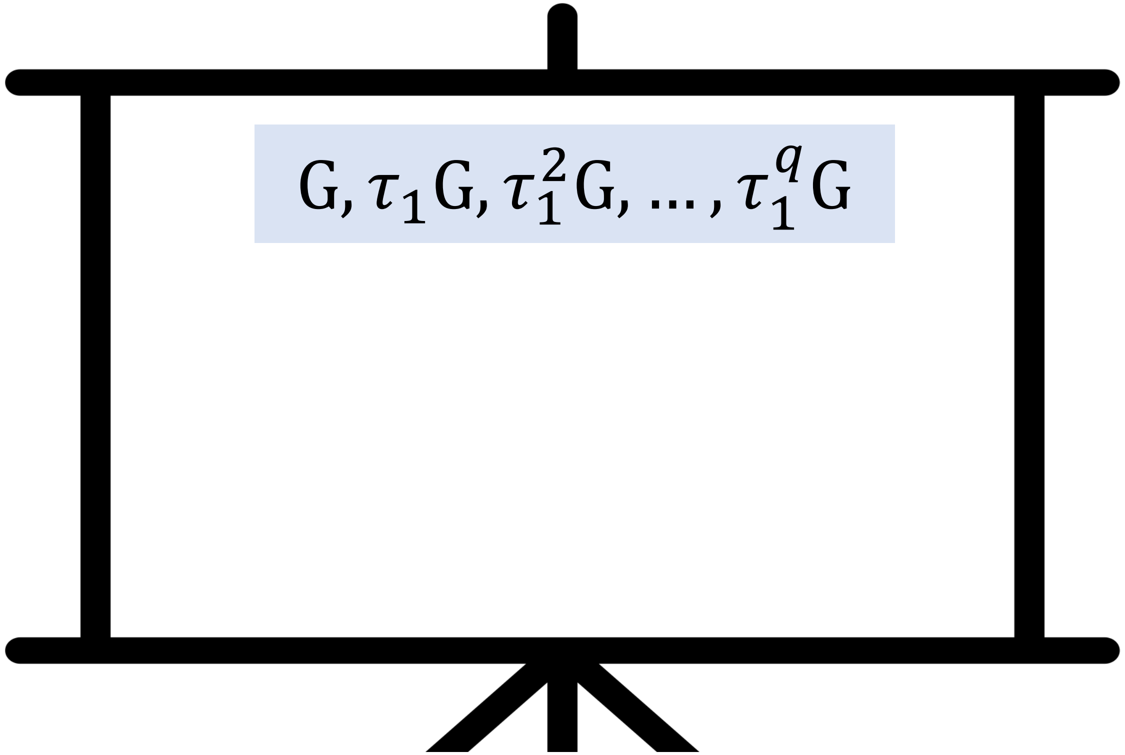
Round Robin approach



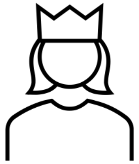
- Sample $\tau_1 \leftarrow \mathbb{F}$
- Compute $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \dots, \tau_1^q\mathrm{G}$
- Post $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \dots, \tau_1^q\mathrm{G}$



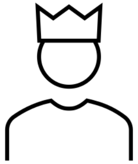
- Download $\mathrm{G}_0, \mathrm{G}_1, \mathrm{G}_2, \dots, \mathrm{G}_q$
- Sample $\tau_2 \leftarrow \mathbb{F}$;



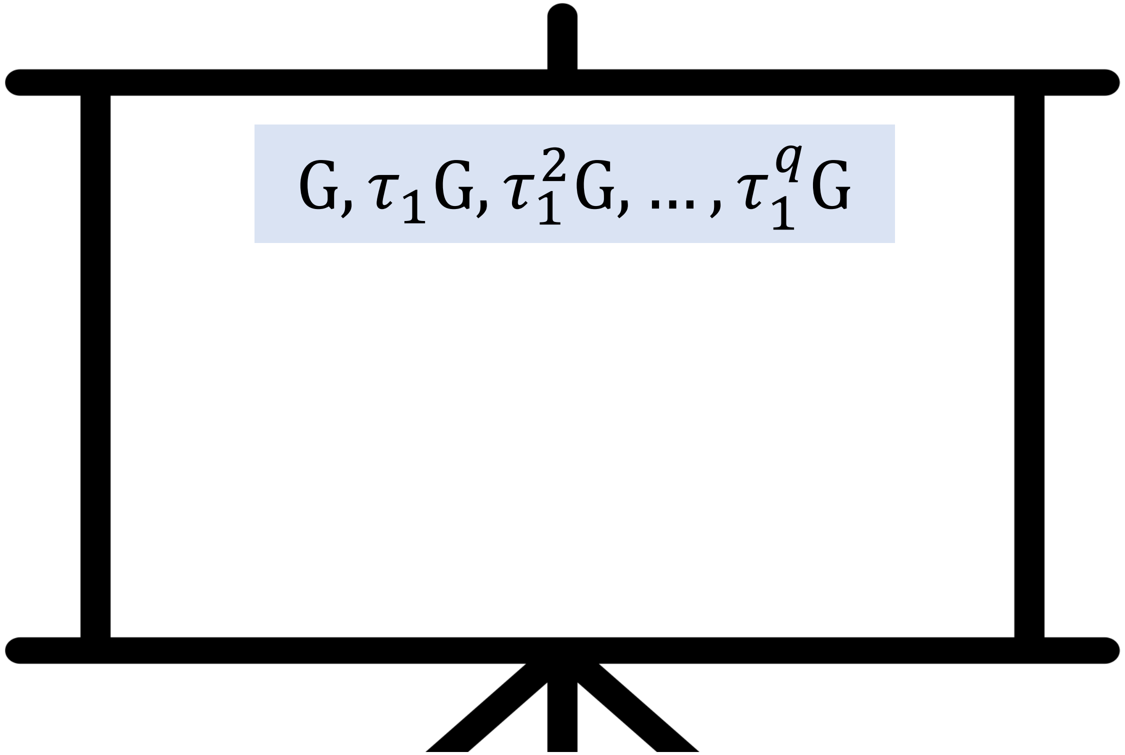
Round Robin approach



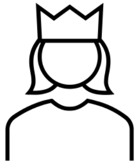
- Sample $\tau_1 \leftarrow \mathbb{F}$
- Compute $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \dots, \tau_1^q\mathrm{G}$
- Post $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \dots, \tau_1^q\mathrm{G}$



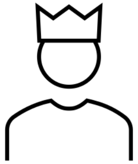
- Download $\mathrm{G}_0, \mathrm{G}_1, \mathrm{G}_2, \dots, \mathrm{G}_q$
- Sample $\tau_2 \leftarrow \mathbb{F}$;
- Compute $\mathrm{G}_0, \tau_2\mathrm{G}_1, \tau_2^2\mathrm{G}_2, \dots, \tau_2^q\mathrm{G}_q$



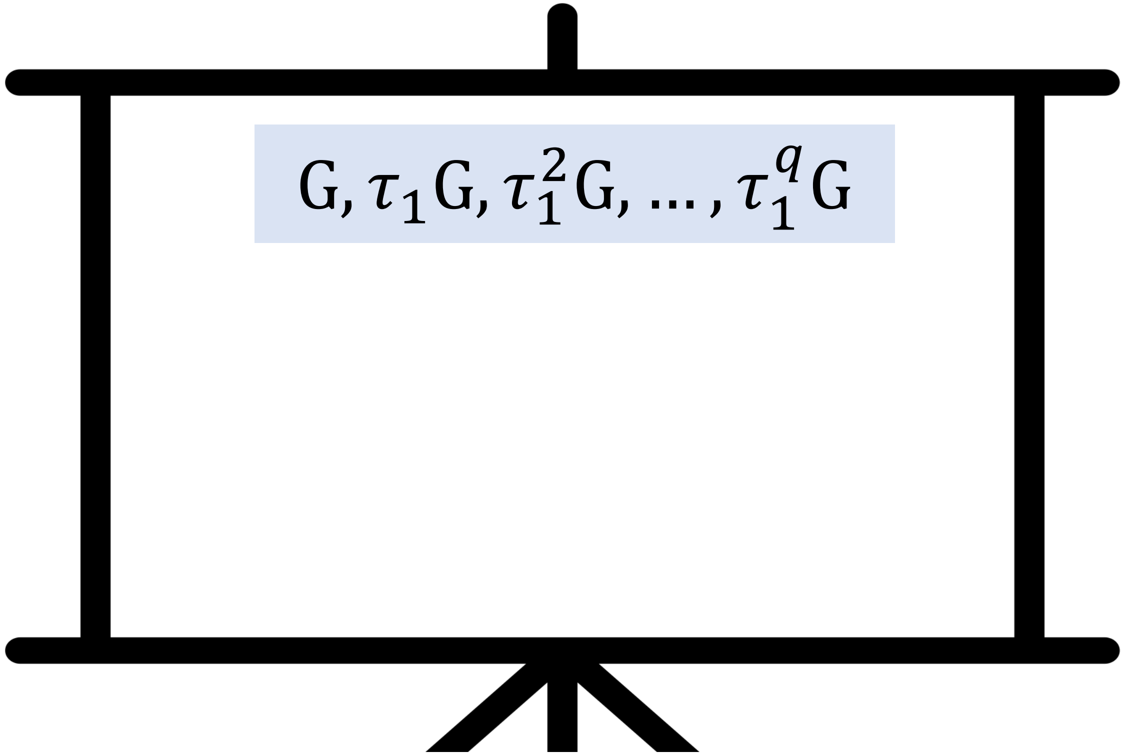
Round Robin approach



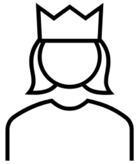
- Sample $\tau_1 \leftarrow \mathbb{F}$
- Compute $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \ldots, \tau_1^q\mathrm{G}$
- Post $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \ldots, \tau_1^q\mathrm{G}$



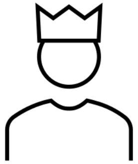
- Download $\mathrm{G}_0, \mathrm{G}_1, \mathrm{G}_2, \ldots, \mathrm{G}_q$
- Sample $\tau_2 \leftarrow \mathbb{F}$;
- Compute $\mathrm{G}_0, \tau_2\mathrm{G}_1, \tau_2^2\mathrm{G}_2, \ldots, \tau_2^q\mathrm{G}_q$
- Broadcast $\mathrm{G}_0, \tau_2\mathrm{G}_1, \tau_2^2\mathrm{G}_2, \ldots, \tau_2^q\mathrm{G}_q$



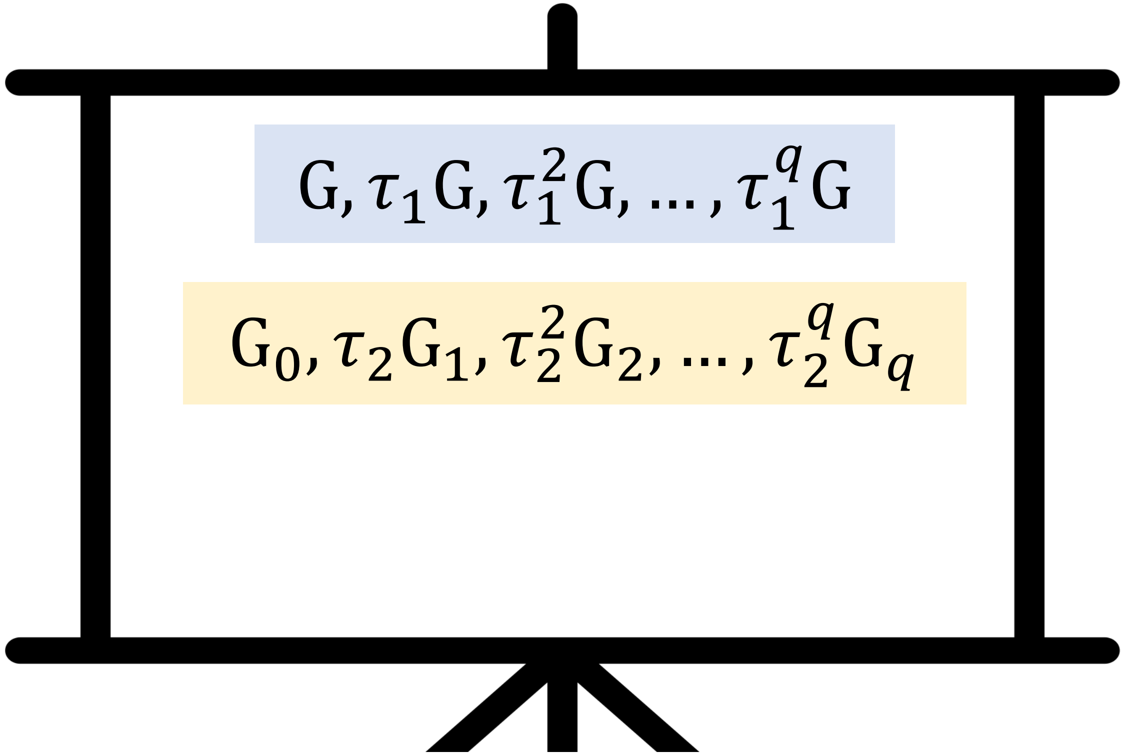
Round Robin approach



- Sample $\tau_1 \leftarrow \mathbb{F}$
- Compute $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \ldots, \tau_1^q\mathrm{G}$
- Post $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \ldots, \tau_1^q\mathrm{G}$



- Download $\mathrm{G}_0, \mathrm{G}_1, \mathrm{G}_2, \ldots, \mathrm{G}_q$
- Sample $\tau_2 \leftarrow \mathbb{F}$;
- Compute $\mathrm{G}_0, \tau_2\mathrm{G}_1, \tau_2^2\mathrm{G}_2, \ldots, \tau_2^q\mathrm{G}_q$
- Broadcast $\mathrm{G}_0, \tau_2\mathrm{G}_1, \tau_2^2\mathrm{G}_2, \ldots, \tau_2^q\mathrm{G}_q$

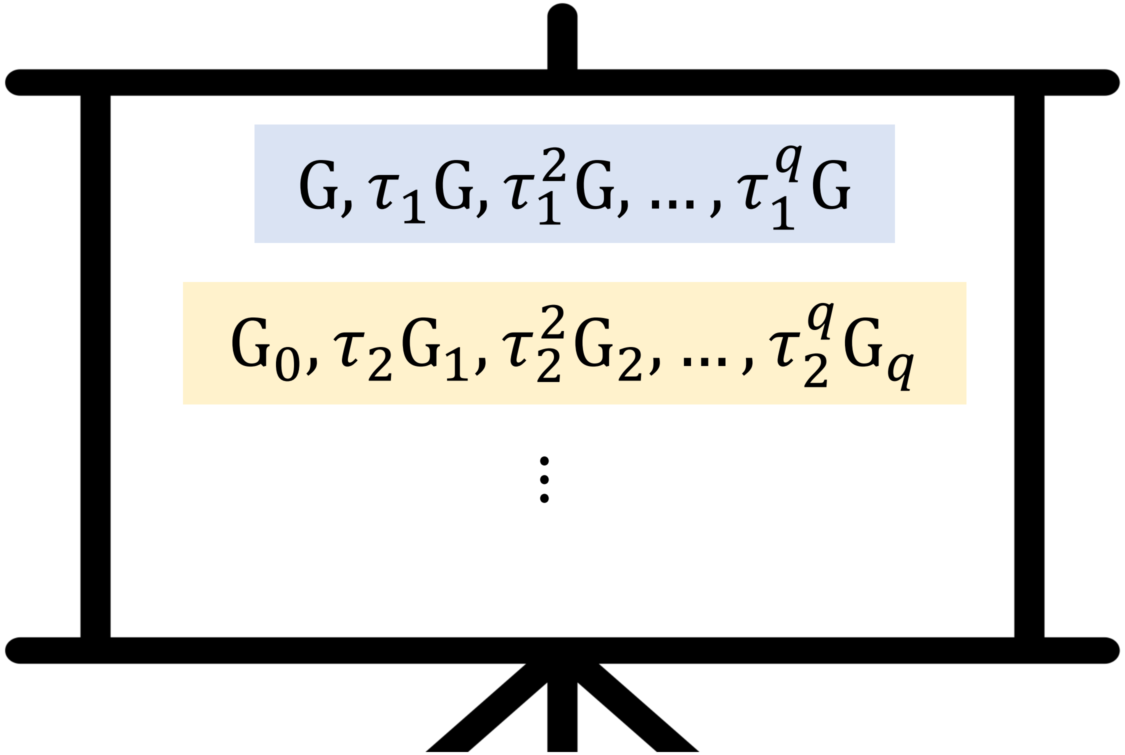


Round Robin approach

- Sample $\tau_1 \leftarrow \mathbb{F}$
- Compute $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \dots, \tau_1^q\mathrm{G}$
- Post $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \dots, \tau_1^q\mathrm{G}$

- Download $\mathrm{G}_0, \mathrm{G}_1, \mathrm{G}_2, \dots, \mathrm{G}_q$
- Sample $\tau_2 \leftarrow \mathbb{F}$;
- Compute $\mathrm{G}_0, \tau_2\mathrm{G}_1, \tau_2^2\mathrm{G}_2, \dots, \tau_2^q\mathrm{G}_q$
- Broadcast $\mathrm{G}_0, \tau_2\mathrm{G}_1, \tau_2^2\mathrm{G}_2, \dots, \tau_2^q\mathrm{G}_q$

$\vdots$

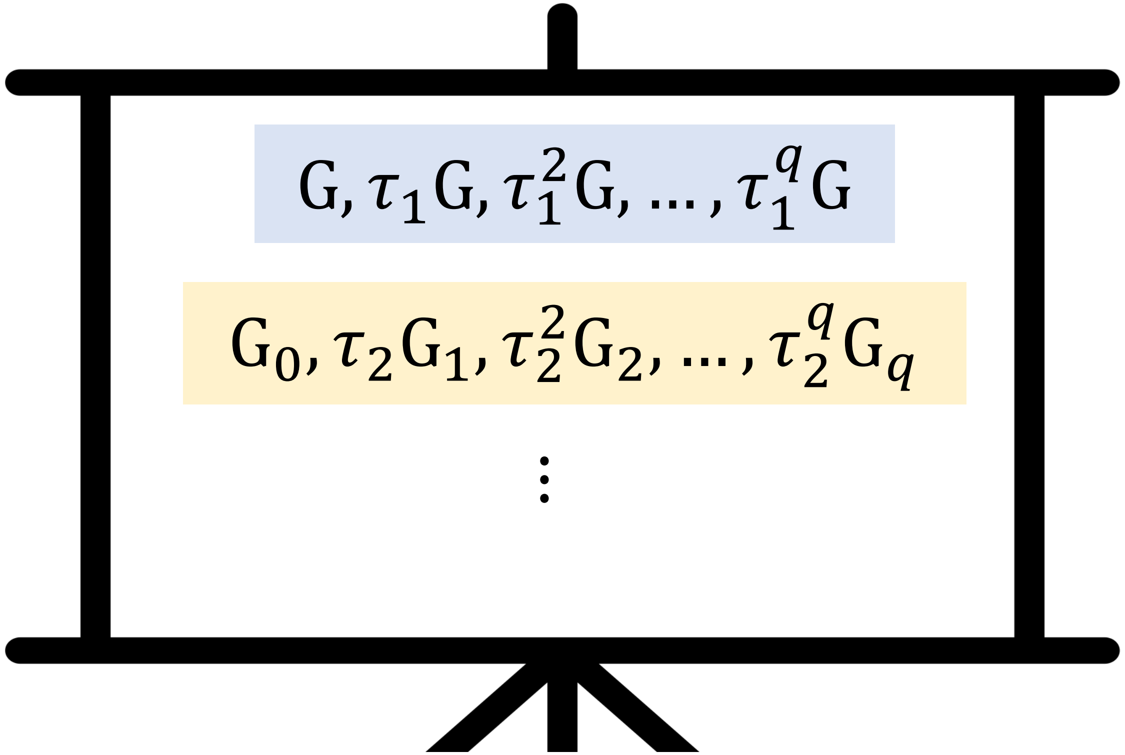


Round Robin approach

- Sample $\tau_1 \leftarrow \mathbb{F}$
- Compute $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \dots, \tau_1^q\mathrm{G}$
- Post $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \dots, \tau_1^q\mathrm{G}$

- Download $\mathrm{G}_0, \mathrm{G}_1, \mathrm{G}_2, \dots, \mathrm{G}_q$
- Sample $\tau_2 \leftarrow \mathbb{F}$;
- Compute $\mathrm{G}_0, \tau_2\mathrm{G}_1, \tau_2^2\mathrm{G}_2, \dots, \tau_2^q\mathrm{G}_q$
- Broadcast $\mathrm{G}_0, \tau_2\mathrm{G}_1, \tau_2^2\mathrm{G}_2, \dots, \tau_2^q\mathrm{G}_q$

$\vdots$



Final output: $\mathrm{G}, (\tau_1\tau_2\cdots\tau_n)\mathrm{G},\ (\tau_1\tau_2\cdots\tau_n)^2\mathrm{G},\ \cdots\ (\tau_1\tau_2\cdots\tau_n)^q\mathrm{G}$

Round Robin approach

- Sample $\tau_1 \leftarrow \mathbb{F}$
- Compute $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \dots, \tau_1^q\mathrm{G}$
- Post $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \dots, \tau_1^q\mathrm{G}$

- Download $\mathrm{G}_0, \mathrm{G}_1, \mathrm{G}_2, \dots, \mathrm{G}_q$
- Sample $\tau_2 \leftarrow \mathbb{F}$;
- Compute $\mathrm{G}_0, \tau_2\mathrm{G}_1, \tau_2^2\mathrm{G}_2, \dots, \tau_2^q\mathrm{G}_q$
- Broadcast $\mathrm{G}_0, \tau_2\mathrm{G}_1, \tau_2^2\mathrm{G}_2, \dots, \tau_2^q\mathrm{G}_q$

$\vdots$

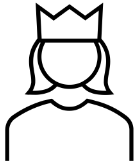
$\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \dots, \tau_1^q\mathrm{G}$

$\mathrm{G}_0, \tau_2\mathrm{G}_1, \tau_2^2\mathrm{G}_2, \dots, \tau_2^q\mathrm{G}_q$

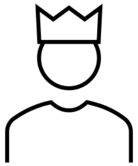
$\vdots$

Final output: $\mathrm{G}, (\tau_1\tau_2\cdots\tau_n)\mathrm{G},\ (\tau_1\tau_2\cdots\tau_n)^2\mathrm{G},\ \cdots\ (\tau_1\tau_2\cdots\tau_n)^q\mathrm{G}$

Round Robin approach

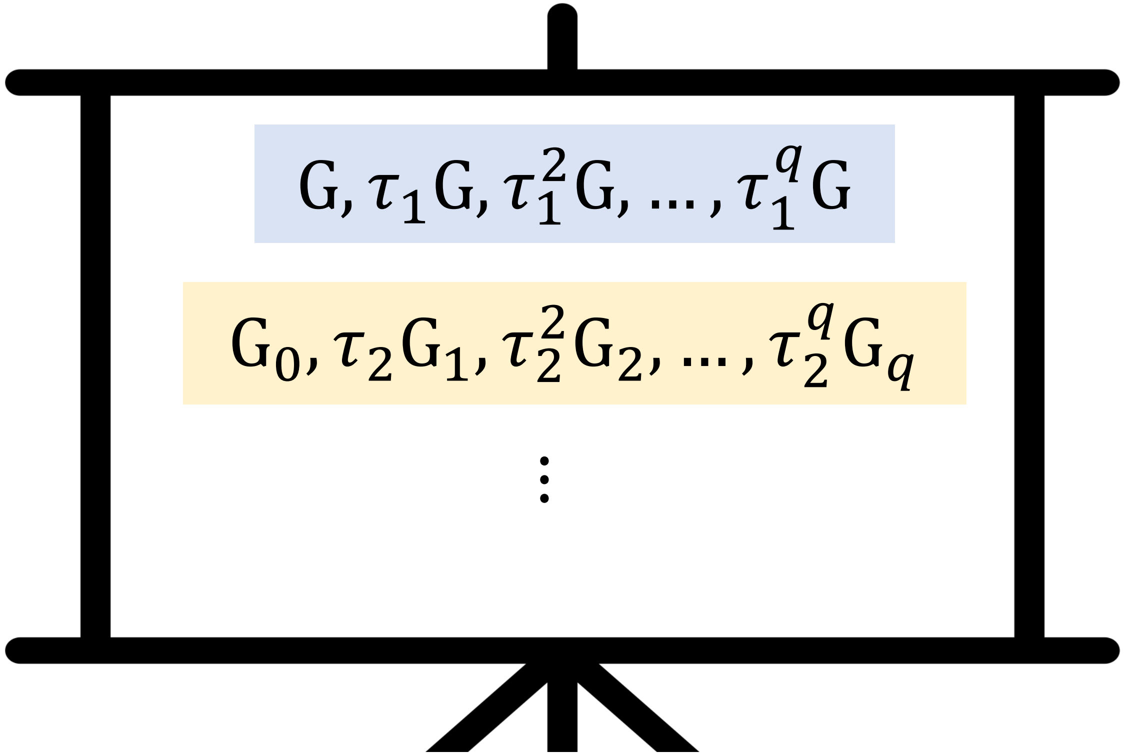


- Sample $\tau_1 \leftarrow \mathbb{F}$
- Compute $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \ldots, \tau_1^q\mathrm{G}$
- Post $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \ldots, \tau_1^q\mathrm{G}$



- Download $\mathrm{G}_0, \mathrm{G}_1, \mathrm{G}_2, \ldots, \mathrm{G}_q$
- Sample $\tau_2 \leftarrow \mathbb{F}$;
- Compute $\mathrm{G}_0, \tau_2\mathrm{G}_1, \tau_2^2\mathrm{G}_2, \ldots, \tau_2^q\mathrm{G}_q$
- Broadcast $\mathrm{G}_0, \tau_2\mathrm{G}_1, \tau_2^2\mathrm{G}_2, \ldots, \tau_2^q\mathrm{G}_q$

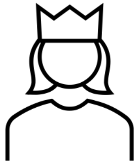
$\vdots$



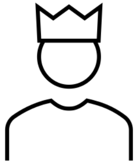
Final output: $\mathrm{G}, (\tau_1\tau_2\cdots\tau_n)\mathrm{G},\ (\tau_1\tau_2\cdots\tau_n)^2\mathrm{G},\ \cdots\ (\tau_1\tau_2\cdots\tau_n)^q\mathrm{G}$

+ Parties need not be fixed a priori

Round Robin approach

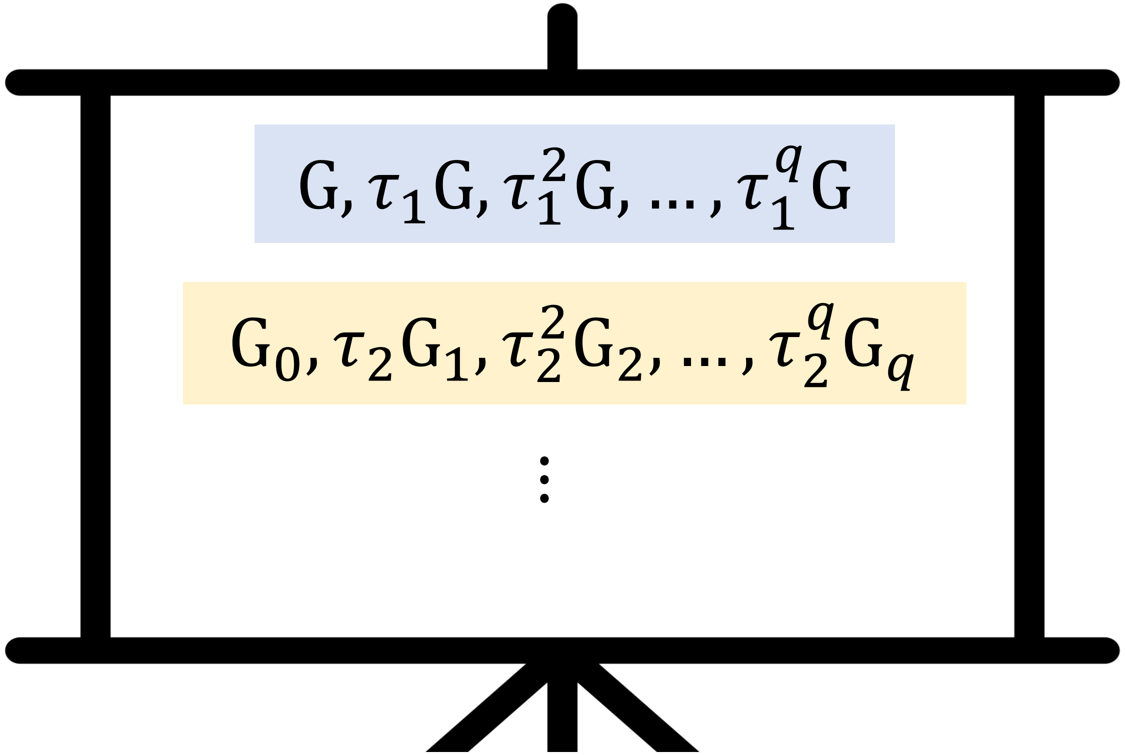


- Sample $\tau_1 \leftarrow \mathbb{F}$
- Compute $\mathrm{G}, \tau_1 \mathrm{G}, \tau_1^2 \mathrm{G}, \dots, \tau_1^q \mathrm{G}$
- Post $\mathrm{G}, \tau_1 \mathrm{G}, \tau_1^2 \mathrm{G}, \dots, \tau_1^q \mathrm{G}$



- Download $\mathrm{G}_0, \mathrm{G}_1, \mathrm{G}_2, \dots, \mathrm{G}_q$
- Sample $\tau_2 \leftarrow \mathbb{F}$;
- Compute $\mathrm{G}_0, \tau_2 \mathrm{G}_1, \tau_2^2 \mathrm{G}_2, \dots, \tau_2^q \mathrm{G}_q$
- Broadcast $\mathrm{G}_0, \tau_2 \mathrm{G}_1, \tau_2^2 \mathrm{G}_2, \dots, \tau_2^q \mathrm{G}_q$

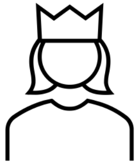
$\vdots$



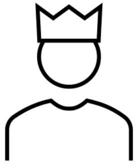
Final output: $\mathrm{G}, (\tau_1 \tau_2 \cdots \tau_n)\mathrm{G},\ (\tau_1 \tau_2 \cdots \tau_n)^2 \mathrm{G},\ \cdots\ (\tau_1 \tau_2 \cdots \tau_n)^q \mathrm{G}$

+ Parties need not be fixed a priori
+ Only one honest party is needed

Round Robin approach

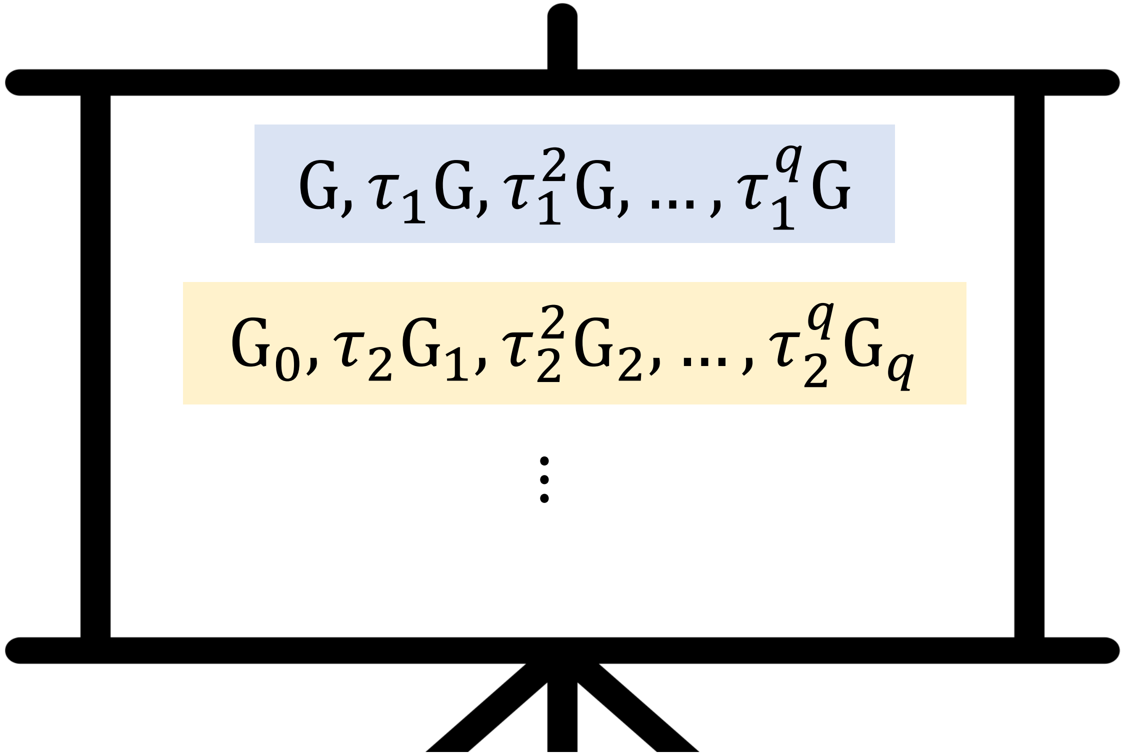


- Sample $\tau_1 \leftarrow \mathbb{F}$
- Compute $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \ldots, \tau_1^q\mathrm{G}$
- Post $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \ldots, \tau_1^q\mathrm{G}$



- Download $\mathrm{G}_0, \mathrm{G}_1, \mathrm{G}_2, \ldots, \mathrm{G}_q$
- Sample $\tau_2 \leftarrow \mathbb{F}$;
- Compute $\mathrm{G}_0, \tau_2\mathrm{G}_1, \tau_2^2\mathrm{G}_2, \ldots, \tau_2^q\mathrm{G}_q$
- Broadcast $\mathrm{G}_0, \tau_2\mathrm{G}_1, \tau_2^2\mathrm{G}_2, \ldots, \tau_2^q\mathrm{G}_q$

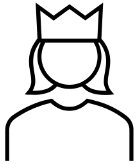
$\vdots$



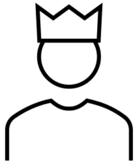
Final output: $\mathrm{G}, (\tau_1\tau_2\cdots\tau_n)\mathrm{G},\ (\tau_1\tau_2\cdots\tau_n)^2\mathrm{G},\ \cdots\ (\tau_1\tau_2\cdots\tau_n)^q\mathrm{G}$

+ Parties need not be fixed a priori
+ Only one honest party is needed

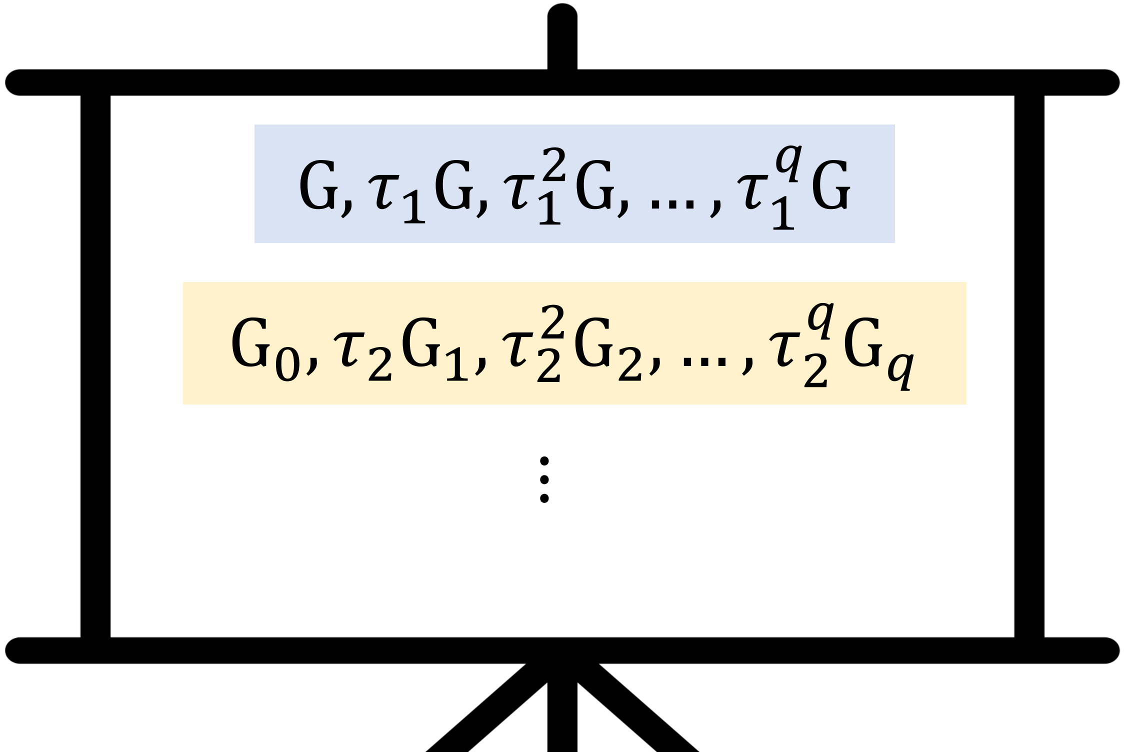
Round Robin approach



- Sample $\tau_1 \leftarrow \mathbb{F}$
- Compute $G, \tau_1 G, \tau_1^2 G, \dots, \tau_1^q G$
- Post $G, \tau_1 G, \tau_1^2 G, \dots, \tau_1^q G$



- Download $G_0, G_1, G_2, \dots, G_q$
- Sample $\tau_2 \leftarrow \mathbb{F}$;
- Compute $G_0, \tau_2 G_1, \tau_2^2 G_2, \dots, \tau_2^q G_q$
- Broadcast $G_0, \tau_2 G_1, \tau_2^2 G_2, \dots, \tau_2^q G_q$

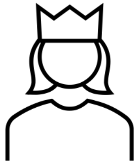


$\vdots$

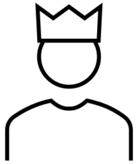
Final output: $G, (\tau_1 \tau_2 \cdots \tau_n) G,\ (\tau_1 \tau_2 \cdots \tau_n)^2 G,\ \cdots\ (\tau_1 \tau_2 \cdots \tau_n)^q G$

+ Parties need not be fixed a priori
+ Only one honest party is needed

Round Robin approach

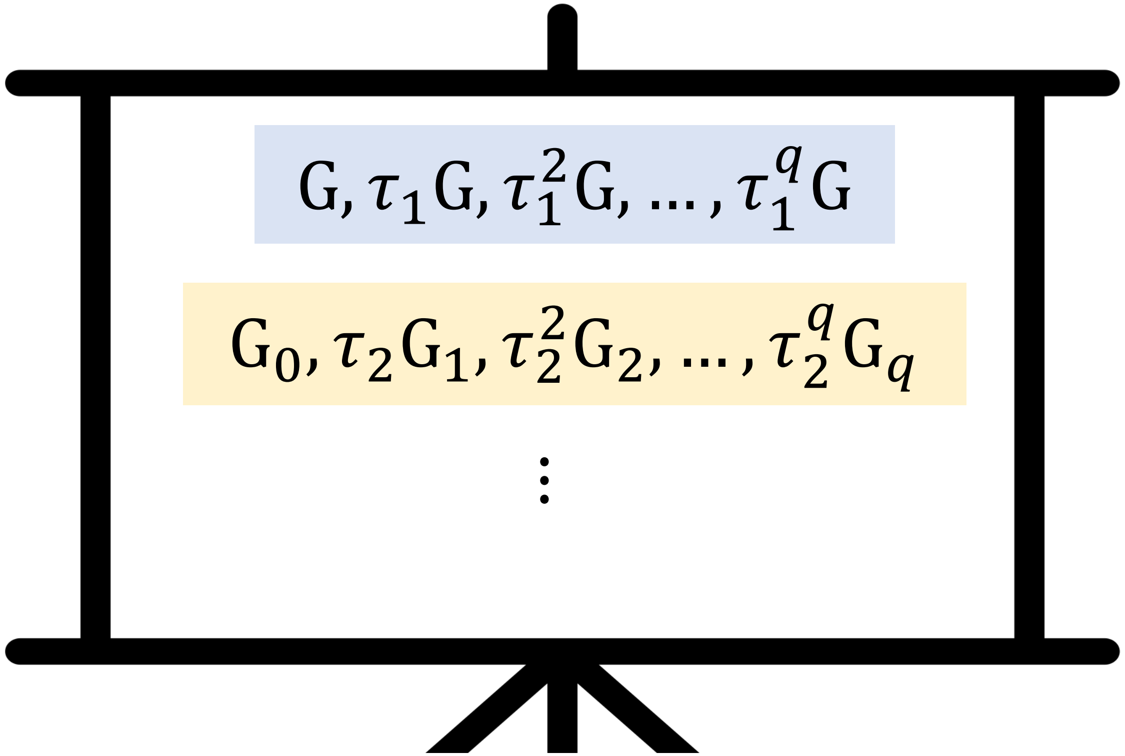


- Sample $\tau_1 \leftarrow \mathbb{F}$
- Compute $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \dots, \tau_1^q\mathrm{G}$
- Post $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \dots, \tau_1^q\mathrm{G}$



- Download $\mathrm{G}_0, \mathrm{G}_1, \mathrm{G}_2, \dots, \mathrm{G}_q$
- Sample $\tau_2 \leftarrow \mathbb{F}$;
- Compute $\mathrm{G}_0, \tau_2\mathrm{G}_1, \tau_2^2\mathrm{G}_2, \dots, \tau_2^q\mathrm{G}_q$
- Broadcast $\mathrm{G}_0, \tau_2\mathrm{G}_1, \tau_2^2\mathrm{G}_2, \dots, \tau_2^q\mathrm{G}_q$

$\vdots$

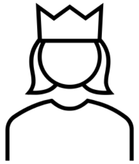


Final output: $\mathrm{G}, (\tau_1\tau_2\cdots\tau_n)\mathrm{G},\ (\tau_1\tau_2\cdots\tau_n)^2\mathrm{G},\ \cdots\ (\tau_1\tau_2\cdots\tau_n)^q\mathrm{G}$

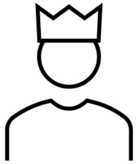
+ Parties need not be fixed a priori
+ Only one honest party is needed

- Require $\Omega(n)$ sequential broadcasts

Round Robin approach

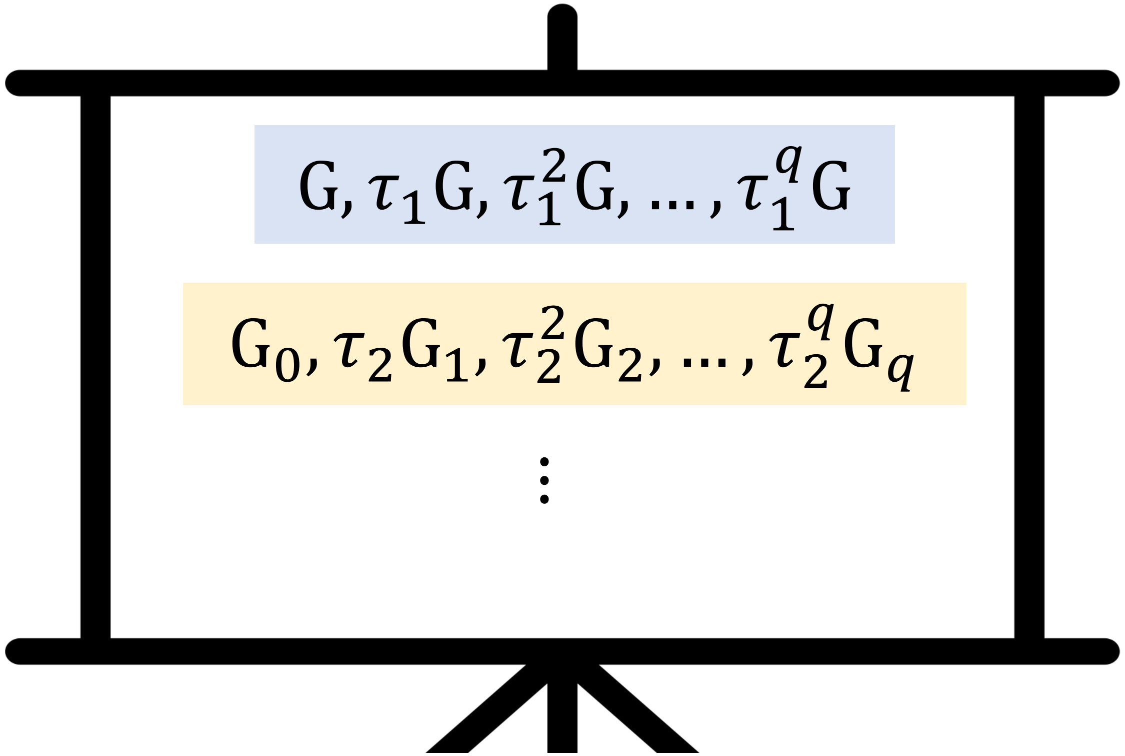


- Sample $\tau_1 \leftarrow \mathbb{F}$
- Compute $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \dots, \tau_1^q\mathrm{G}$
- Post $\mathrm{G}, \tau_1\mathrm{G}, \tau_1^2\mathrm{G}, \dots, \tau_1^q\mathrm{G}$



- Download $\mathrm{G}_0, \mathrm{G}_1, \mathrm{G}_2, \dots, \mathrm{G}_q$
- Sample $\tau_2 \leftarrow \mathbb{F}$;
- Compute $\mathrm{G}_0, \tau_2\mathrm{G}_1, \tau_2^2\mathrm{G}_2, \dots, \tau_2^q\mathrm{G}_q$
- Broadcast $\mathrm{G}_0, \tau_2\mathrm{G}_1, \tau_2^2\mathrm{G}_2, \dots, \tau_2^q\mathrm{G}_q$

$\vdots$



Final output: $\mathrm{G}, (\tau_1\tau_2 \cdots \tau_n)\mathrm{G},\ (\tau_1\tau_2 \cdots \tau_n)^2\mathrm{G},\ \cdots\ (\tau_1\tau_2 \cdots \tau_n)^q\mathrm{G}$

+ Parties need not be fixed a priori
+ Only one honest party is needed

- Require $\Omega(n)$ sequential broadcasts
- Does not work in asynchrony

Our Approach

Our Approach: High-level

Our Approach: High-level

Specialized asynchronous MPC for generating Powers of Tau

Our Approach: High-level

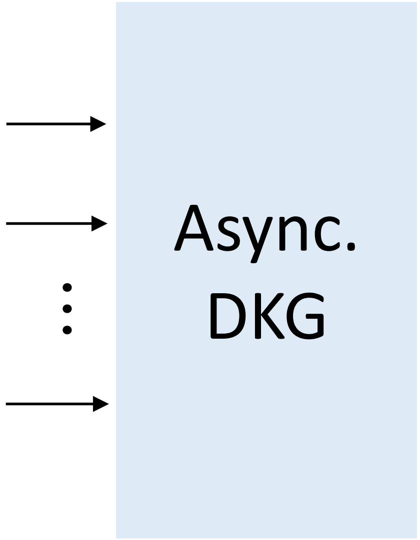
Specialized asynchronous MPC for generating Powers of Tau

Three phases:

Our Approach: High-level

Specialized asynchronous MPC for generating Powers of Tau

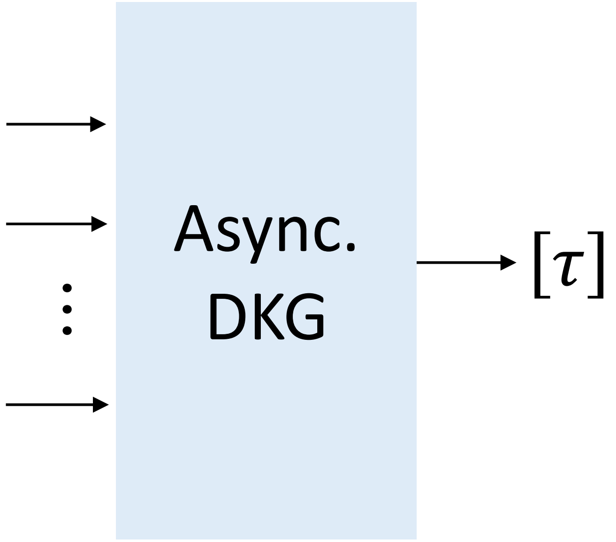
Three phases:



Our Approach: High-level

Specialized asynchronous MPC for generating Powers of Tau

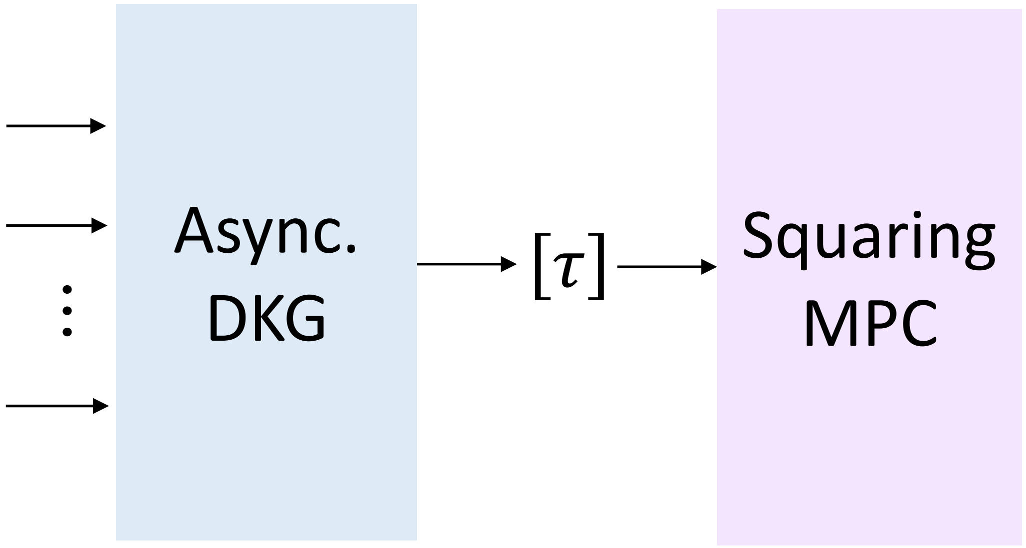
Three phases:



Our Approach: High-level

Specialized asynchronous MPC for generating Powers of Tau

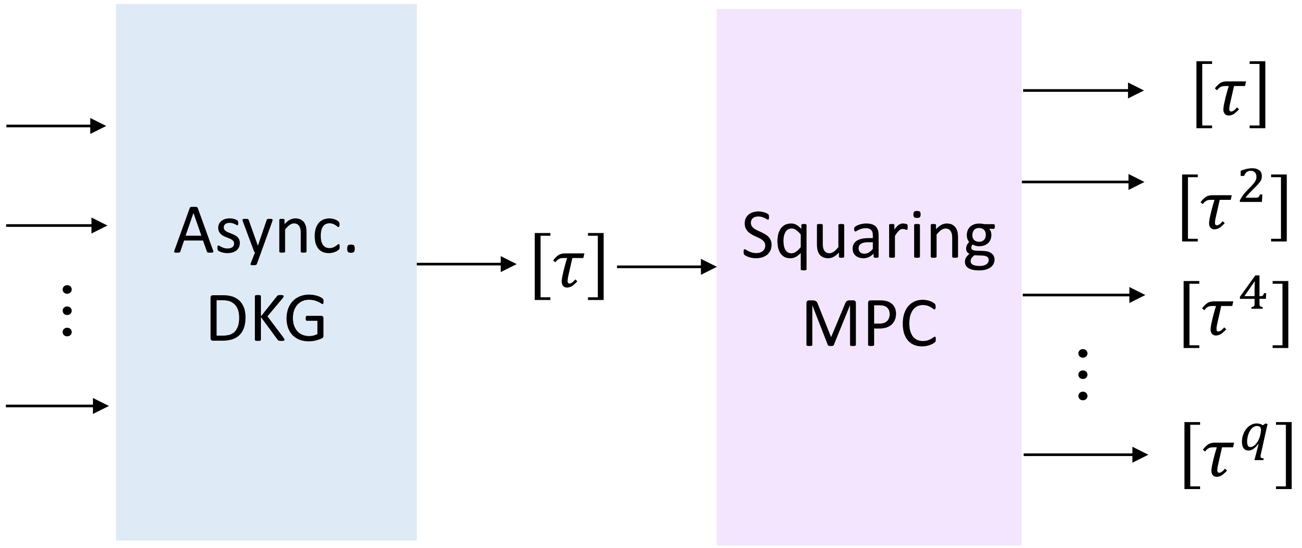
Three phases:



Our Approach: High-level

Specialized asynchronous MPC for generating Powers of Tau

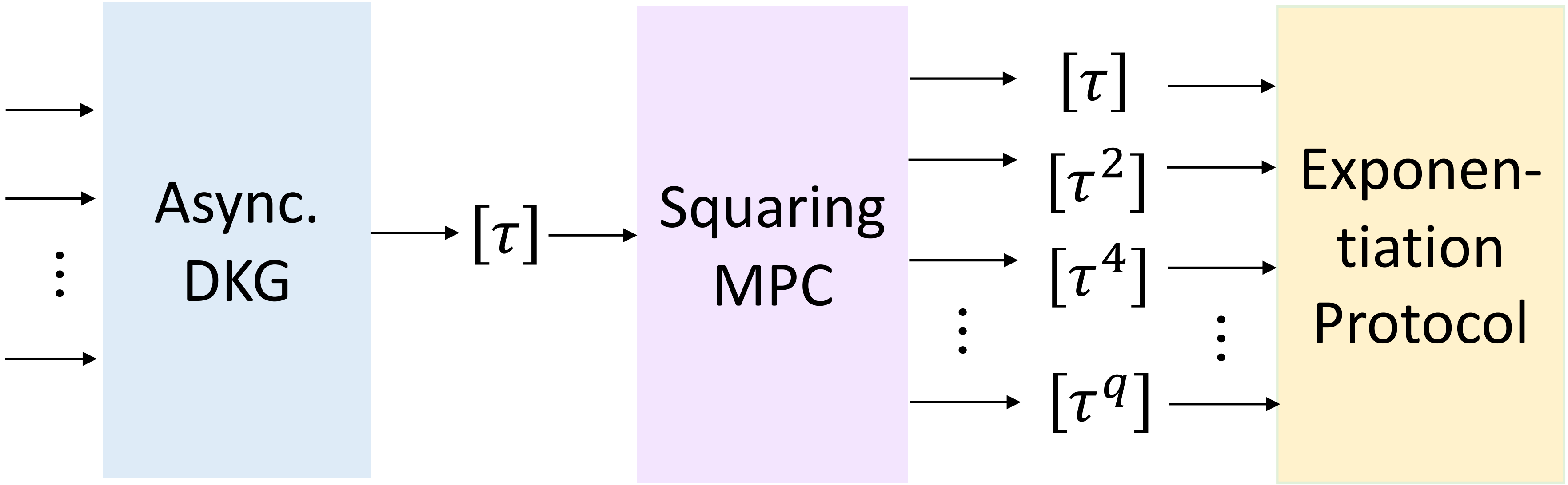
Three phases:



Our Approach: High-level

Specialized asynchronous MPC for generating Powers of Tau

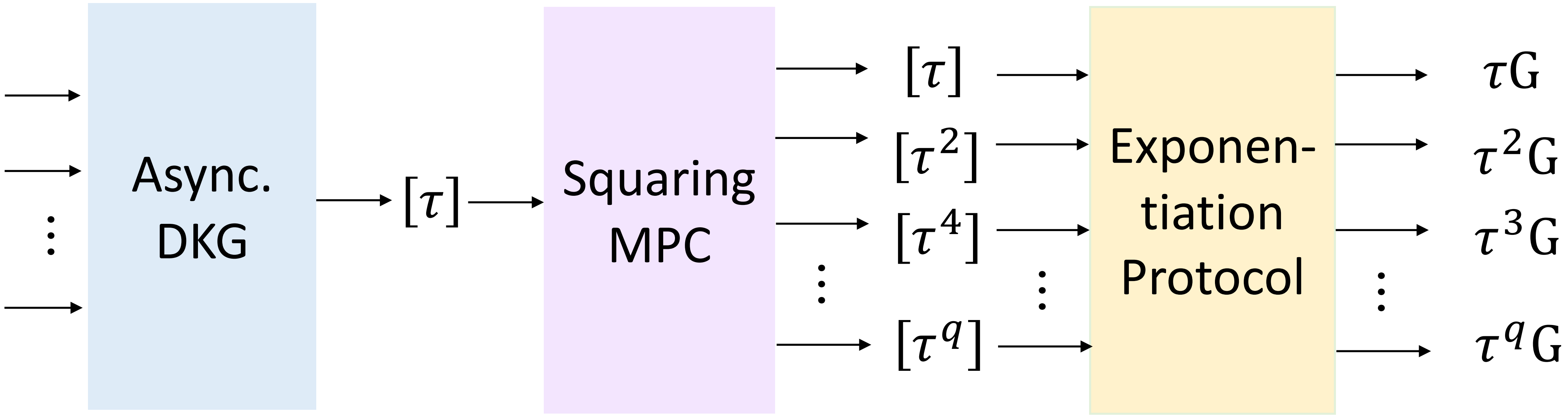
Three phases:



Our Approach: High-level

Specialized asynchronous MPC for generating Powers of Tau

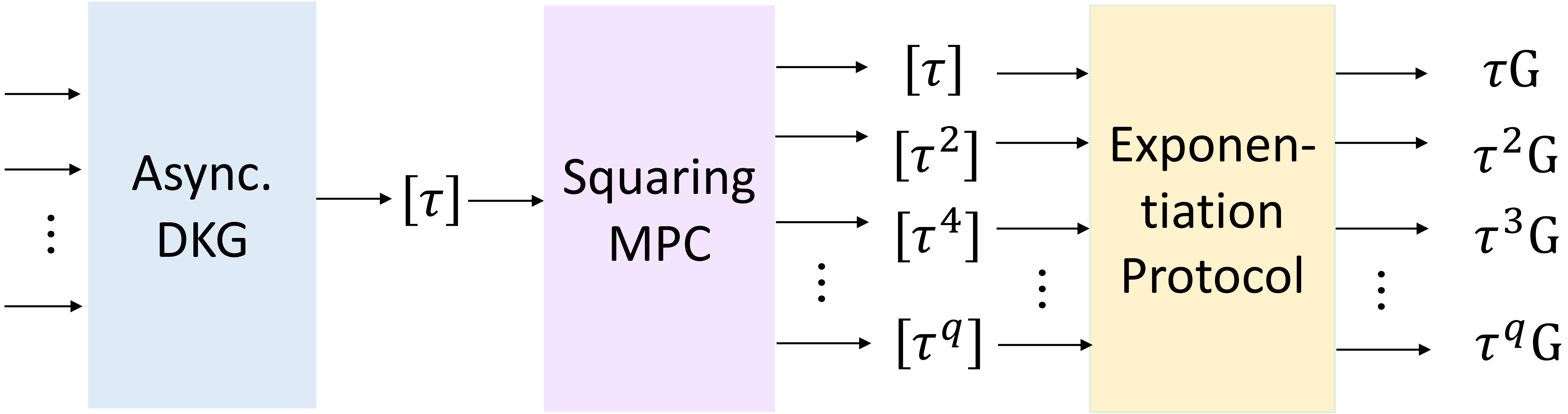
Three phases:



Our Approach: High-level

Specialized asynchronous MPC for generating Powers of Tau

Three phases:

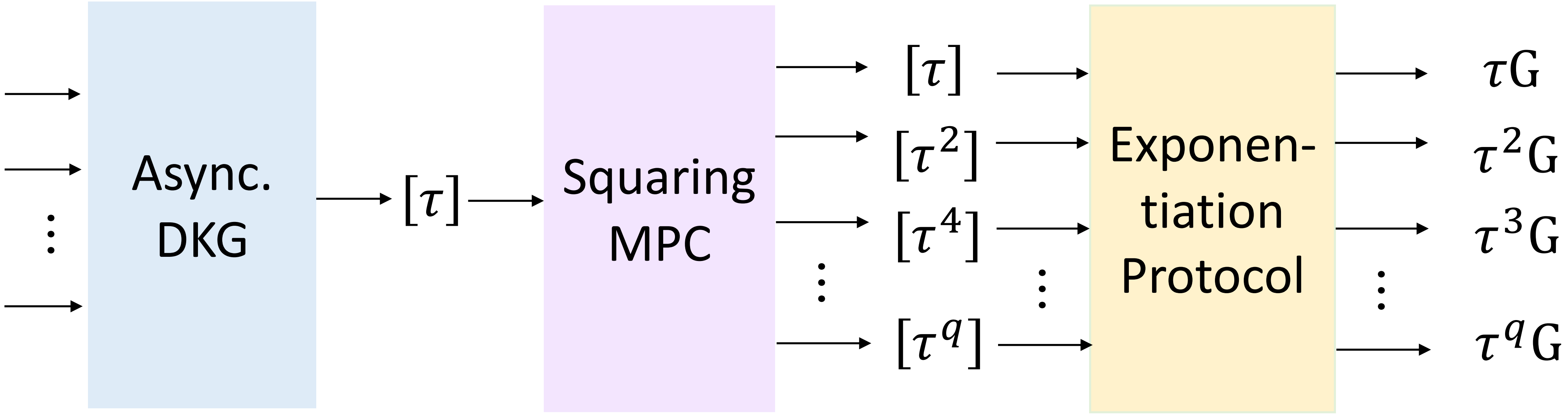


Only $O(\log q)$ multiplication units are needed

Our Approach: High-level

Specialized asynchronous MPC for generating Powers of Tau

Three phases:



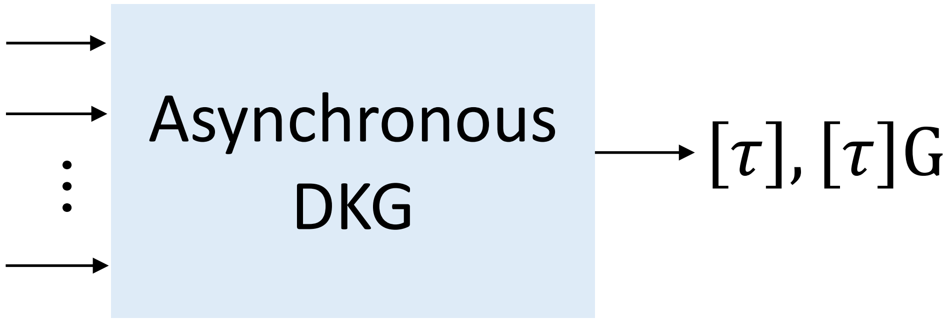
Only $\mathrm{O}(\log q)$ multiplication units are needed

All parts can be implemented with expected $\mathrm{O}(\log q + \log n)$ rounds

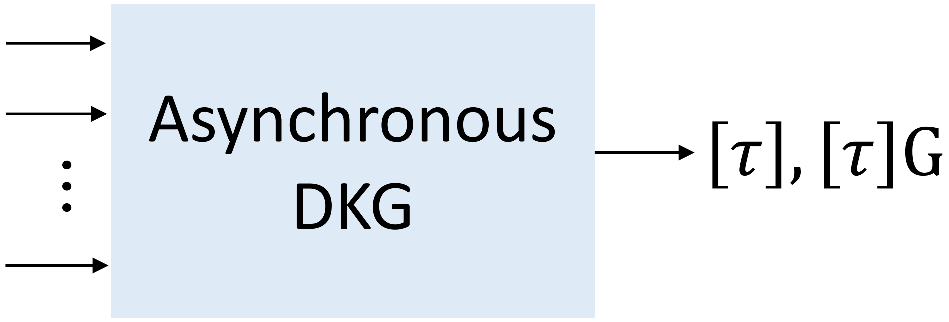
Step 1: Asynchronous DKG

Step 1: Asynchronous DKG

Step 1: Asynchronous DKG

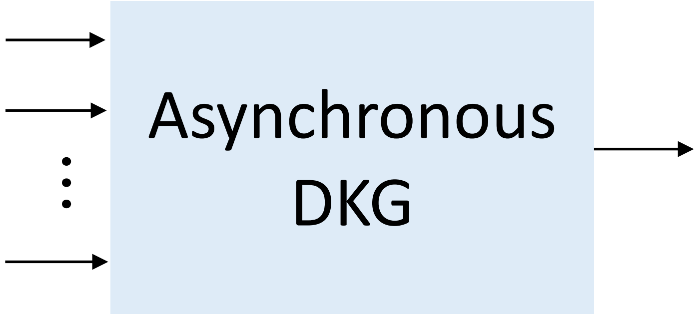


Step 1: Asynchronous DKG



where $[\tau]\mathrm{G} = [\tau]_1\mathrm{G}, [\tau]_2\mathrm{G}, \ldots [\tau]_n\mathrm{G}$

Step 1: Asynchronous DKG

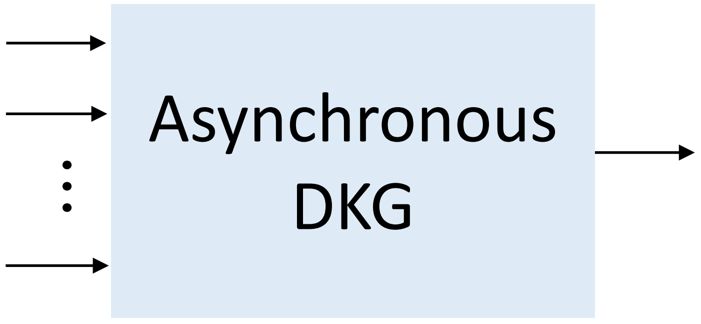


$[\tau], [\tau]\mathrm{G}$

where $[\tau]\mathrm{G} = [\tau]_1\mathrm{G}, [\tau]_2\mathrm{G}, \ldots [\tau]_n\mathrm{G}$

$[\tau]\mathrm{G}$ are also called as threshold public keys

Step 1: Asynchronous DKG



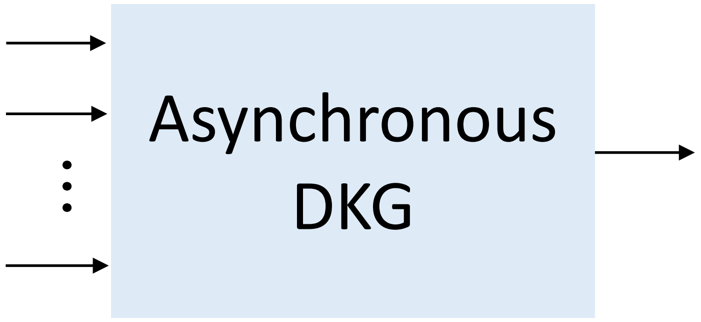
$[\tau], [\tau]\mathrm{G}$

where $[\tau]\mathrm{G} = [\tau]_1\mathrm{G}, [\tau]_2\mathrm{G}, \ldots [\tau]_n\mathrm{G}$

$[\tau]\mathrm{G}$ are also called as threshold public keys

- We use the Asynchronous DKG protocol from [DXKR’23]

Step 1: Asynchronous DKG



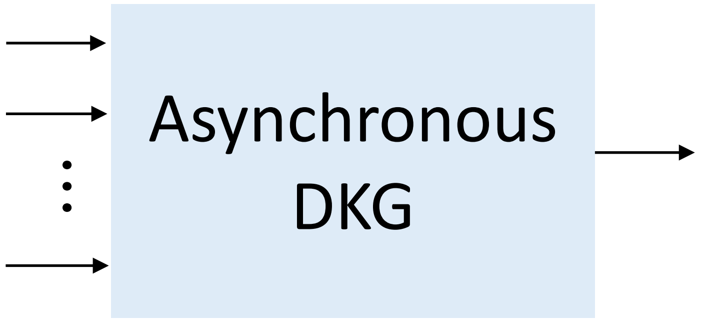
$[\tau], [\tau]\mathrm{G}$

where $[\tau]\mathrm{G} = [\tau]_1\mathrm{G}, [\tau]_2\mathrm{G}, \ldots [\tau]_n\mathrm{G}$

$[\tau]\mathrm{G}$ are also called as threshold public keys

- We use the Asynchronous DKG protocol from [DXKR’23]
  - $\mathrm{O}(n^2)$ per-party communication cost

Step 1: Asynchronous DKG



$[\tau], [\tau]\mathrm{G}$

where $[\tau]\mathrm{G} = [\tau]_1\mathrm{G}, [\tau]_2\mathrm{G}, \dots [\tau]_n\mathrm{G}$

$[\tau]\mathrm{G}$ are also called as threshold public keys

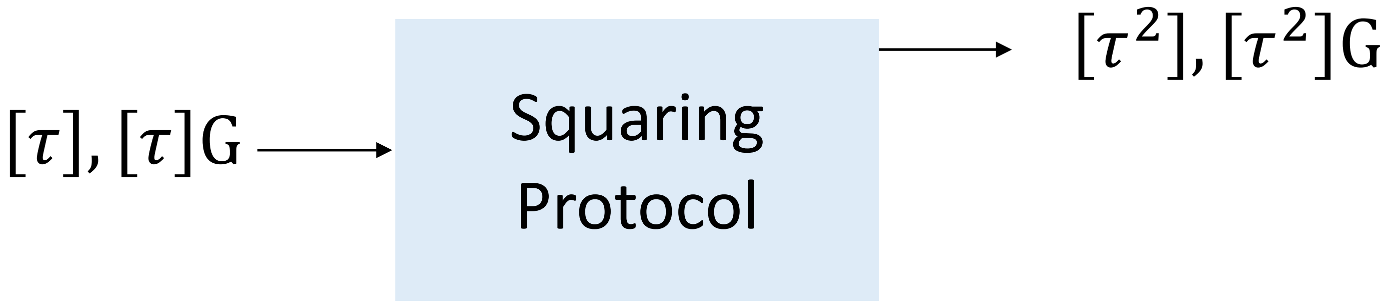
- We use the Asynchronous DKG protocol from [DXKR'23]
  - $\mathrm{O}(n^2)$ per-party communication cost
  - $\mathrm{O}(\log n)$ expected rounds

Step 2: Squaring protocol

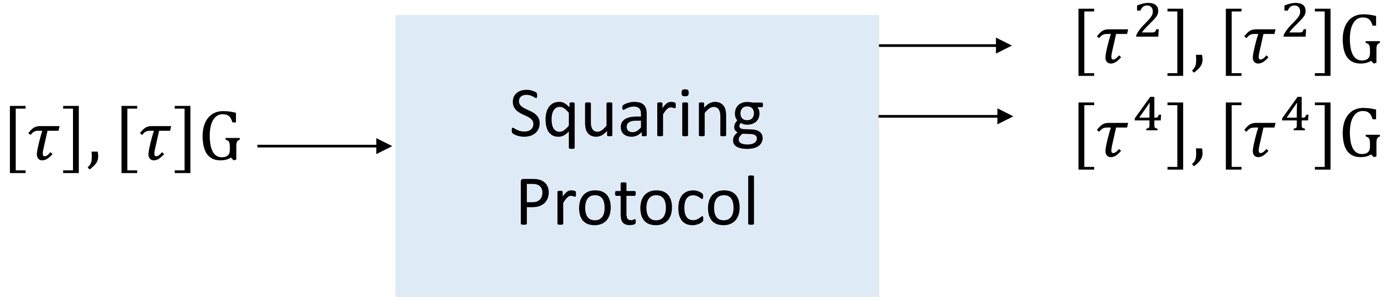
Step 2: Squaring protocol

$[\tau], [\tau]\mathrm{G} \longrightarrow$ Squaring Protocol

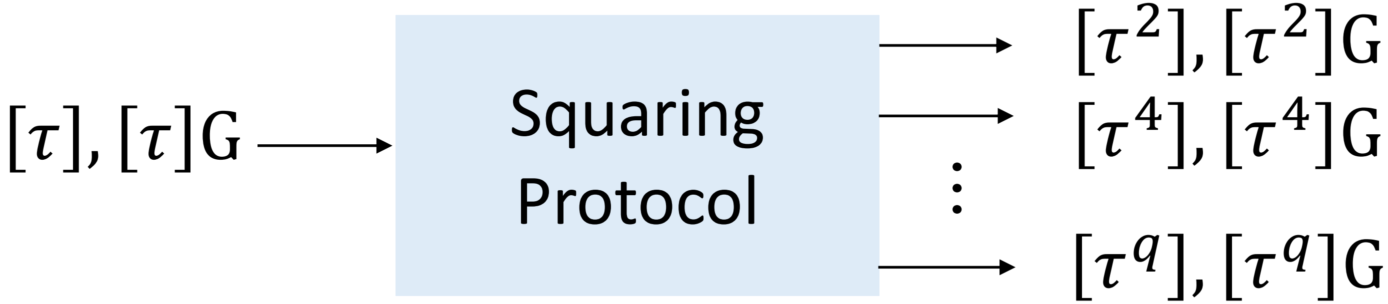
Step 2: Squaring protocol



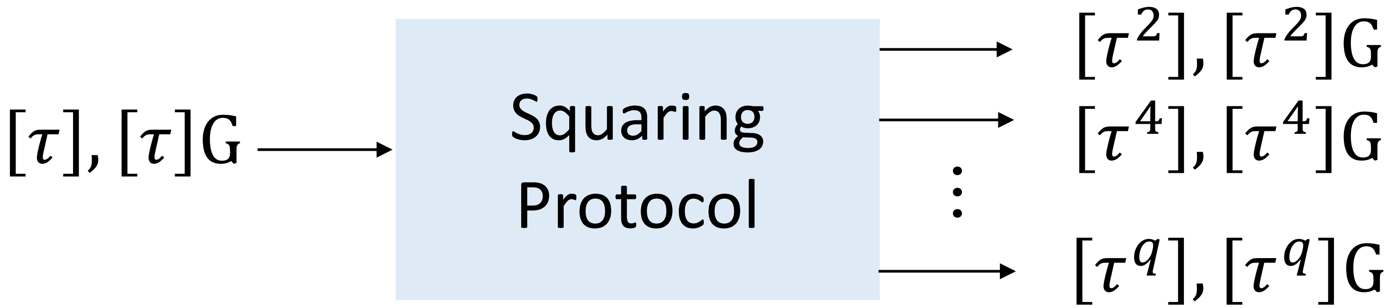
Step 2: Squaring protocol



Step 2: Squaring protocol

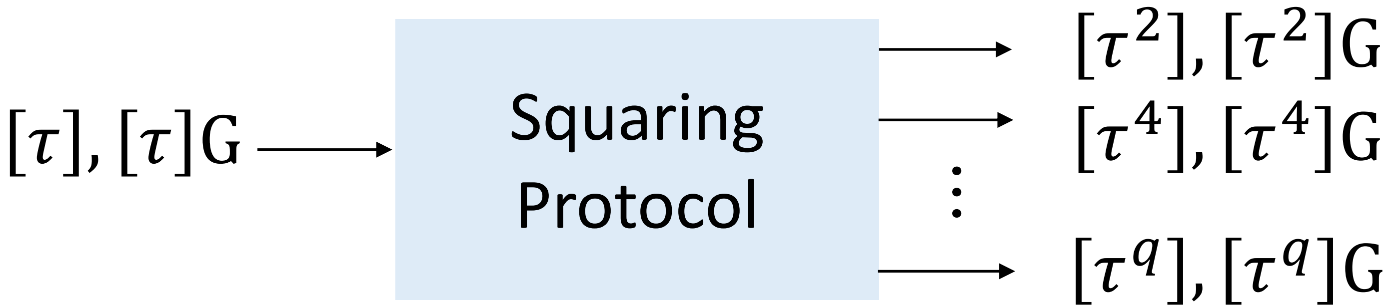


Step 2: Squaring protocol



Double sharing-based MPC multiplication

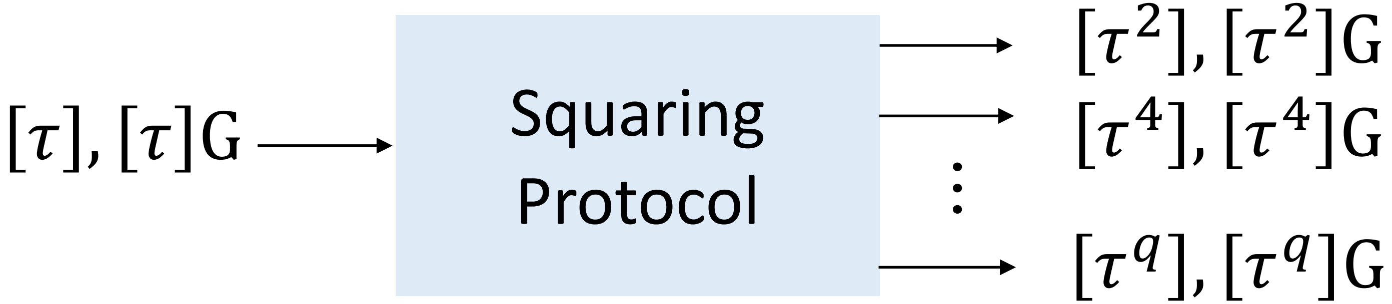
Step 2: Squaring protocol



Double sharing-based MPC multiplication

- Let $[a]^t$ and $[a]^{2t}$ be degree t and $2t$ sharing of a $a \leftarrow \mathbb{F}$

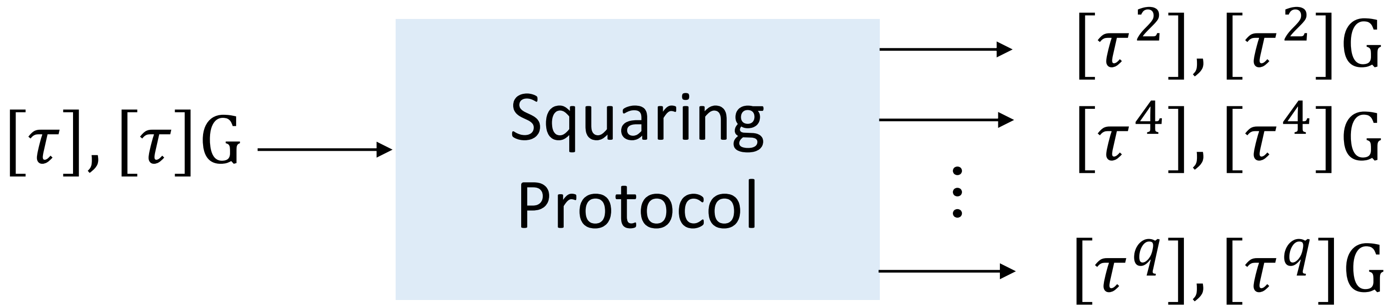
Step 2: Squaring protocol



Double sharing-based MPC multiplication

- Let $[a]^t$ and $[a]^{2t}$ be degree t and $2t$ sharing of a $a \leftarrow \mathbb{F}$
- Let $[a]^t\mathrm{G}$ and $[a]^{2t}\mathrm{G}$ be threshold public keys

Step 2: Squaring protocol



Double sharing-based MPC multiplication

- Let $[a]^t$ and $[a]^{2t}$ be degree t and $2t$ sharing of a $a \leftarrow \mathbb{F}$
- Let $[a]^t\mathrm{G}$ and $[a]^{2t}\mathrm{G}$ be threshold public keys

$([\tau], [a]^{2t})$

Step 2: Squaring protocol

$[\tau], [\tau]G \longrightarrow$ **Squaring Protocol** $\longrightarrow [\tau^2], [\tau^2]G$; $[\tau^4], [\tau^4]G$; $\vdots$; $[\tau^q], [\tau^q]G$

Double sharing-based MPC multiplication

- Let $[a]^t$ and $[a]^{2t}$ be degree t and $2t$ sharing of a $a \leftarrow \mathbb{F}$
- Let $[a]^t\mathrm{G}$ and $[a]^{2t}\mathrm{G}$ be threshold public keys

$$([\tau], [a]^{2t}) \longrightarrow \boxed{\mathrm{Reveal}([\tau][\tau] + [a]^{2t})}$$

Step 2: Squaring protocol

$[\tau], [\tau]G \longrightarrow$ **Squaring Protocol** $\longrightarrow [\tau^2], [\tau^2]G$

$\longrightarrow [\tau^4], [\tau^4]G$

$\vdots$

$\longrightarrow [\tau^q], [\tau^q]G$

Double sharing-based MPC multiplication

- Let $[a]^t$ and $[a]^{2t}$ be degree t and $2t$ sharing of a $a \leftarrow \mathbb{F}$
- Let $[a]^t G$ and $[a]^{2t} G$ be threshold public keys

$$([\tau], [a]^{2t}) \longrightarrow \boxed{\text{Reveal}([\tau][\tau] + [a]^{2t})} \longrightarrow z$$

Step 2: Squaring protocol

$[\tau], [\tau]\mathrm{G} \longrightarrow$ **Squaring Protocol** $\longrightarrow$

$[\tau^2], [\tau^2]\mathrm{G}$

$[\tau^4], [\tau^4]\mathrm{G}$

$\vdots$

$[\tau^q], [\tau^q]\mathrm{G}$

Double sharing-based MPC multiplication

- Let $[a]^t$ and $[a]^{2t}$ be degree t and $2t$ sharing of a $a \leftarrow \mathbb{F}$
- Let $[a]^t\mathrm{G}$ and $[a]^{2t}\mathrm{G}$ be threshold public keys

$$([\tau], [a]^{2t}) \longrightarrow \mathrm{Reveal}([\tau][\tau] + [a]^{2t}) \longrightarrow z$$

Step 2: Squaring protocol

$$[\tau], [\tau]\mathrm{G} \longrightarrow \boxed{\text{Squaring Protocol}} \longrightarrow \begin{matrix} [\tau^2], [\tau^2]\mathrm{G} \\ [\tau^4], [\tau^4]\mathrm{G} \\ \vdots \\ [\tau^q], [\tau^q]\mathrm{G} \end{matrix}$$

Double sharing-based MPC multiplication

- Let $[a]^t$ and $[a]^{2t}$ be degree t and $2t$ sharing of a $a \leftarrow \mathbb{F}$
- Let $[a]^t\mathrm{G}$ and $[a]^{2t}\mathrm{G}$ be threshold public keys

$$([\tau], [a]^{2t}) \longrightarrow \boxed{\mathrm{Reveal}([\tau][\tau] + [a]^{2t})} \longrightarrow z$$

Compute $[\tau^2]_i \coloneqq z - [a]_i^t$

Step 2: Squaring protocol

$[\tau], [\tau]\mathrm{G} \longrightarrow$ Squaring Protocol $\longrightarrow [\tau^2], [\tau^2]\mathrm{G}$; $[\tau^4], [\tau^4]\mathrm{G}$; $\vdots$; $[\tau^q], [\tau^q]\mathrm{G}$

Double sharing-based MPC multiplication

- Let $[a]^t$ and $[a]^{2t}$ be degree t and $2t$ sharing of a $a \leftarrow \mathbb{F}$
- Let $[a]^t\mathrm{G}$ and $[a]^{2t}\mathrm{G}$ be threshold public keys

$$([\tau], [a]^{2t}) \longrightarrow \mathrm{Reveal}([\tau][\tau] + [a]^{2t}) \longrightarrow z$$

Compute $[\tau^2]_i \coloneqq z - [a]_i^t$
Compute $[\tau^2]\mathrm{G} \coloneqq (z - [a]^t) \cdot \mathrm{G}$

Step 2: Squaring protocol

$$[\tau], [\tau]\mathrm{G} \longrightarrow \boxed{\text{Squaring Protocol}} \longrightarrow \begin{array}{l} [\tau^2], [\tau^2]\mathrm{G} \\ [\tau^4], [\tau^4]\mathrm{G} \\ \vdots \\ [\tau^q], [\tau^q]\mathrm{G} \end{array}$$

Double sharing-based MPC multiplication

- Let $[a]^t$ and $[a]^{2t}$ be degree t and $2t$ sharing of a $a \leftarrow \mathbb{F}$
- Let $[a]^t\mathrm{G}$ and $[a]^{2t}\mathrm{G}$ be threshold public keys

$$([\tau], [a]^{2t}) \longrightarrow \boxed{\mathrm{Reveal}([\tau][\tau] + [a]^{2t})} \longrightarrow z$$

Compute $[\tau^2]_i \coloneqq z - [a]_i^t$
Compute $[\tau^2]\mathrm{G} \coloneqq (z - [a]^t) \cdot \mathrm{G}$

Double sharing generation from [DXKR'23]

Step 2: Squaring protocol

$[\tau], [\tau]\mathrm{G} \longrightarrow$ Squaring Protocol $\longrightarrow [\tau^2], [\tau^2]\mathrm{G}$

$\longrightarrow [\tau^4], [\tau^4]\mathrm{G}$

$\vdots$

$\longrightarrow [\tau^q], [\tau^q]\mathrm{G}$

Double sharing-based MPC multiplication

- Let $[a]^t$ and $[a]^{2t}$ be degree t and $2t$ sharing of a $a \leftarrow \mathbb{F}$
- Let $[a]^t\mathrm{G}$ and $[a]^{2t}\mathrm{G}$ be threshold public keys

$$([\tau], [a]^{2t}) \longrightarrow \mathrm{Reveal}([\tau][\tau] + [a]^{2t}) \longrightarrow z$$

Compute $[\tau^2]_i := z - [a]_i^t$
Compute $[\tau^2]\mathrm{G} := (z - [a]^t) \cdot \mathrm{G}$

Double sharing generation from [DXKR'23]

- Per party per unit communication cost of $O(n^2)$

Step 2: Squaring protocol

$[\tau], [\tau]\mathrm{G} \longrightarrow$ **Squaring Protocol** $\longrightarrow$

$[\tau^2], [\tau^2]\mathrm{G}$

$[\tau^4], [\tau^4]\mathrm{G}$

$\vdots$

$[\tau^q], [\tau^q]\mathrm{G}$

Double sharing-based MPC multiplication

- Let $[a]^t$ and $[a]^{2t}$ be degree t and $2t$ sharing of a $a \leftarrow \mathbb{F}$
- Let $[a]^t\mathrm{G}$ and $[a]^{2t}\mathrm{G}$ be threshold public keys

$([\tau], [a]^{2t}) \longrightarrow \mathrm{Reveal}([\tau][\tau] + [a]^{2t}) \longrightarrow z$

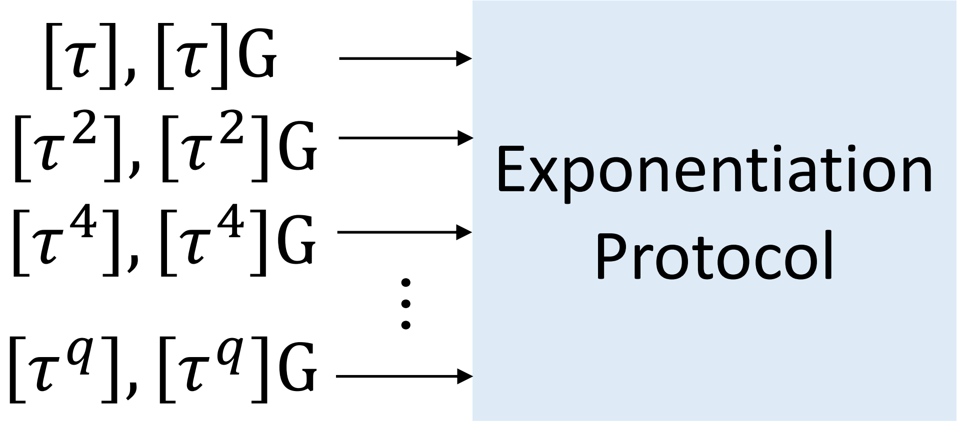
Compute $[\tau^2]_i := z - [a]_i^t$
Compute $[\tau^2]\mathrm{G} := (z - [a]^t) \cdot \mathrm{G}$

Double sharing generation from [DXKR'23]

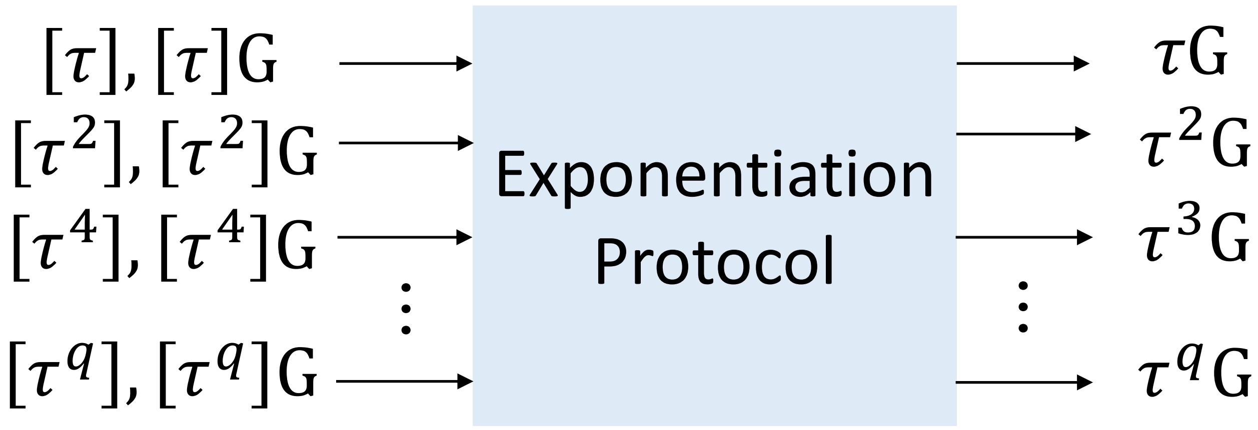
- Per party per unit communication cost of $O(n^2)$
- Per party total communication cost of $O(n^2 \log q)$

Step 3: All exponents

Step 3: All exponents



Step 3: All exponents



Step 3: All exponents

$[\tau], [\tau]G \longrightarrow$ Exponentiation Protocol $\longrightarrow \tau G$

$[\tau^2], [\tau^2]G \longrightarrow$ $\longrightarrow \tau^2 G$

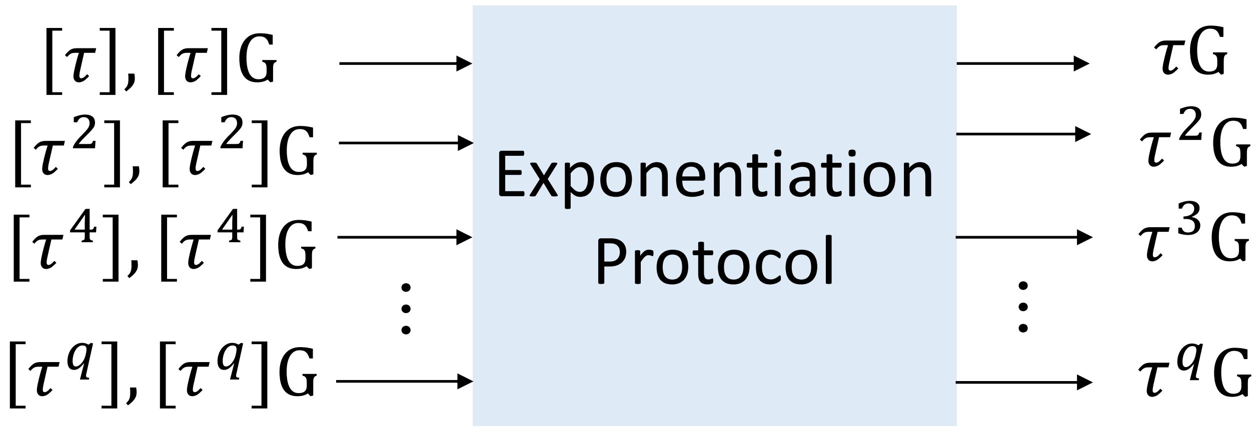
$[\tau^4], [\tau^4]G \longrightarrow$ $\longrightarrow \tau^3 G$

$\vdots$ $\vdots$

$[\tau^q], [\tau^q]G \longrightarrow$ $\longrightarrow \tau^q G$

Example: $\tau^5 G$

Step 3: All exponents



Example: $\tau^5\mathrm{G} = (\tau^{2\cdot 2}\tau)\mathrm{G}$

Step 3: All exponents

Input		Output
$[\tau], [\tau]\mathrm{G}$	Exponentiation Protocol	$\tau\mathrm{G}$
$[\tau^2], [\tau^2]\mathrm{G}$		$\tau^2\mathrm{G}$
$[\tau^4], [\tau^4]\mathrm{G}$		$\tau^3\mathrm{G}$
$\vdots$		$\vdots$
$[\tau^q], [\tau^q]\mathrm{G}$		$\tau^q\mathrm{G}$

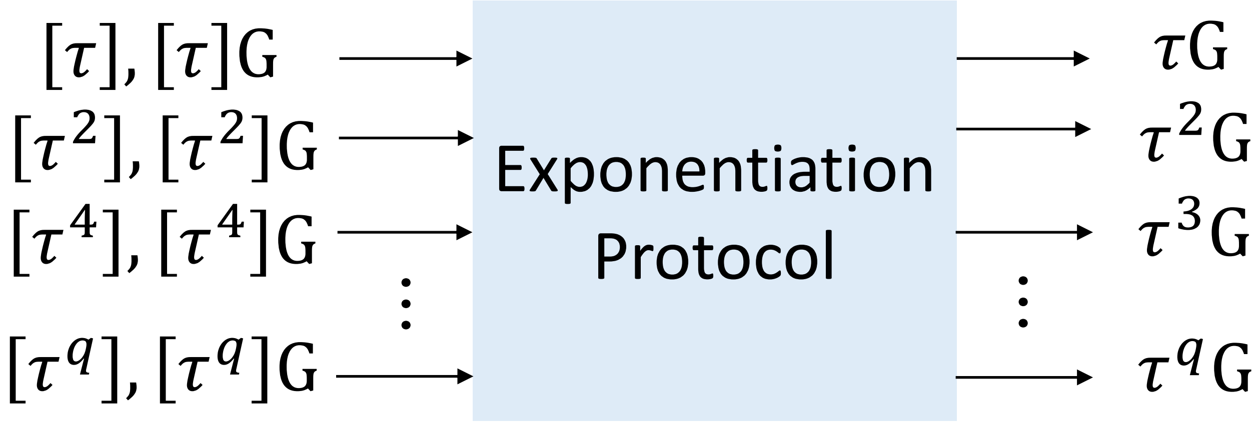
Example: $\tau^5\mathrm{G} = (\tau^{2\cdot 2}\tau)\mathrm{G} = \tau^2(\tau^2\mathrm{G}) \cdot (\tau\mathrm{G})$

Step 3: All exponents

$$
\begin{array}{ccccc}
[\tau], [\tau]\mathrm{G} & \longrightarrow & & \longrightarrow & \tau\mathrm{G} \\
[\tau^2], [\tau^2]\mathrm{G} & \longrightarrow & \text{Exponentiation} & \longrightarrow & \tau^2\mathrm{G} \\
[\tau^4], [\tau^4]\mathrm{G} & \longrightarrow & \text{Protocol} & \longrightarrow & \tau^3\mathrm{G} \\
 & \vdots & & \vdots & \\
[\tau^q], [\tau^q]\mathrm{G} & \longrightarrow & & \longrightarrow & \tau^q\mathrm{G}
\end{array}
$$

Example: $\tau^5\mathrm{G} = (\tau^{2\cdot 2}\tau)\mathrm{G} = \tau^2(\tau^2\mathrm{G}) \cdot (\tau\mathrm{G})$

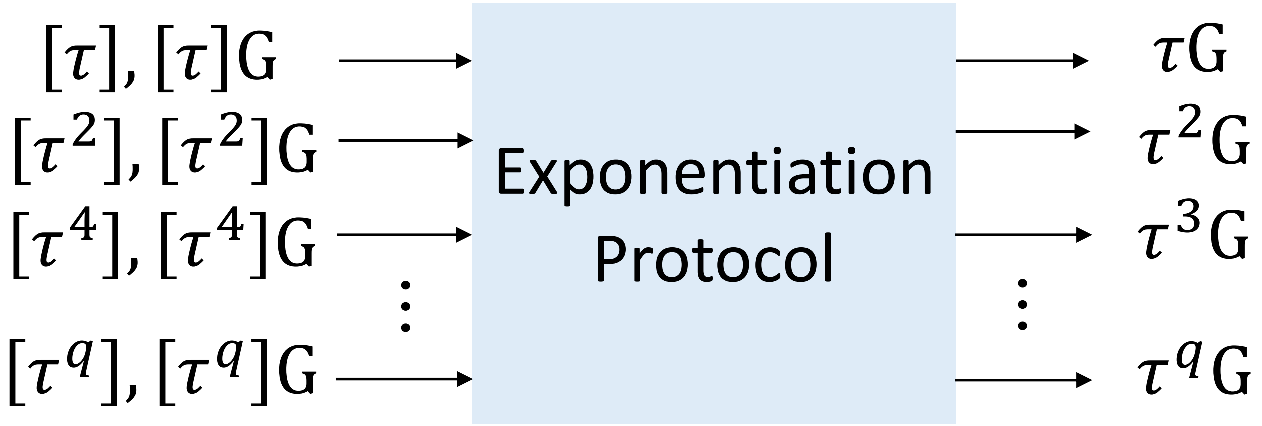
Step 3: All exponents



Example: $\tau^5\mathrm{G} = (\tau^{2\cdot 2}\tau)\mathrm{G} \ = \tau^2(\tau^2\mathrm{G}) \cdot (\tau\mathrm{G})$

Protocol:

Step 3: All exponents



Example: $\tau^5 G = (\tau^{2 \cdot 2}\tau)G \; = \tau^2(\tau^2 G) \cdot (\tau G)$

Protocol:

1. Each node i publishes $[\tau^2]_i(\tau^2 G)$

Step 3: All exponents

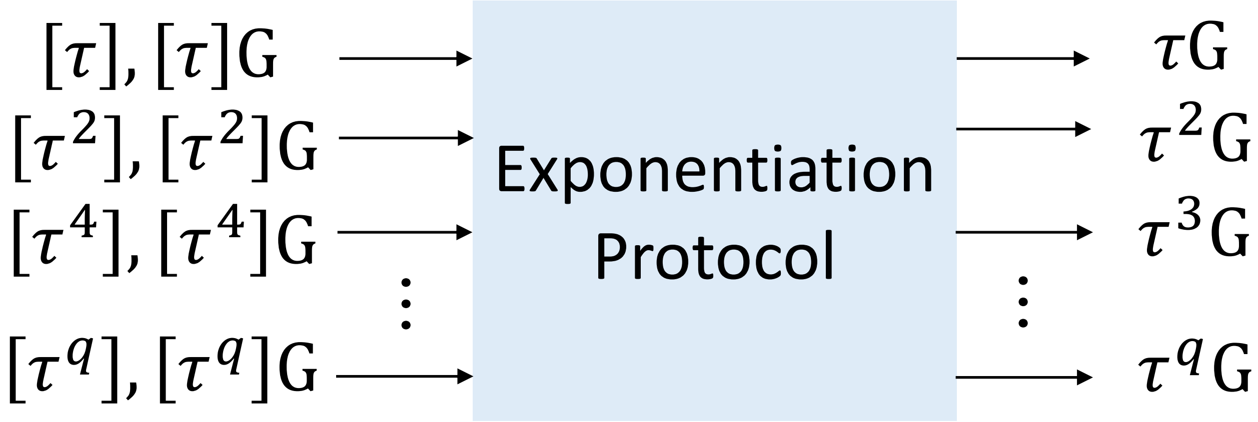
$$\begin{array}{rcl}
[\tau], [\tau]\mathrm{G} \longrightarrow & & \longrightarrow \tau\mathrm{G} \\
[\tau^2], [\tau^2]\mathrm{G} \longrightarrow & \text{Exponentiation} & \longrightarrow \tau^2\mathrm{G} \\
[\tau^4], [\tau^4]\mathrm{G} \longrightarrow & \text{Protocol} & \longrightarrow \tau^3\mathrm{G} \\
\vdots & & \vdots \\
[\tau^q], [\tau^q]\mathrm{G} \longrightarrow & & \longrightarrow \tau^q\mathrm{G}
\end{array}$$

Example: $\tau^5\mathrm{G} = (\tau^{2\cdot 2}\tau)\mathrm{G} \ = \tau^2(\tau^2\mathrm{G}) \cdot (\tau\mathrm{G})$

Protocol:

1. Each node i publishes $[\tau^2]_i(\tau^2\mathrm{G})$
2. Interpolate $[\tau^2]_i(\tau^2\mathrm{G})$ in the exponent to compute $\tau^2(\tau^2\mathrm{G})$

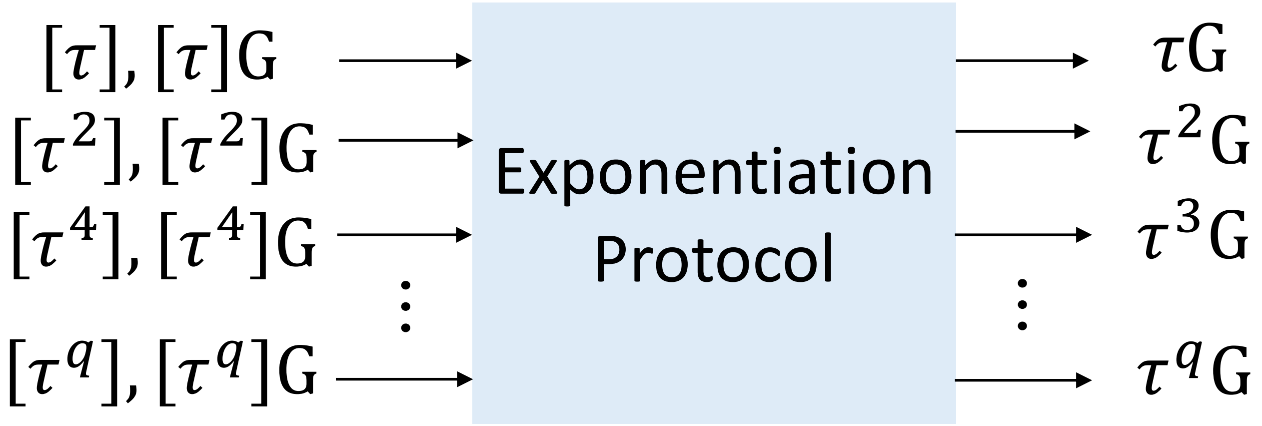
Step 3: All exponents



Example: $\tau^5\mathrm{G} = (\tau^{2\cdot 2}\tau)\mathrm{G} \ = \tau^2(\tau^2\mathrm{G}) \cdot (\tau\mathrm{G})$

Protocol:
1. Each node i publishes $[\tau^2]_i(\tau^2\mathrm{G})$
2. Interpolate $[\tau^2]_i(\tau^2\mathrm{G})$ in the exponent to compute $\tau^2(\tau^2\mathrm{G})$
3. Compute $\tau^2(\tau^2\mathrm{G}) \cdot (\tau\mathrm{G}) = \tau^5\mathrm{G}$

Step 3: All exponents

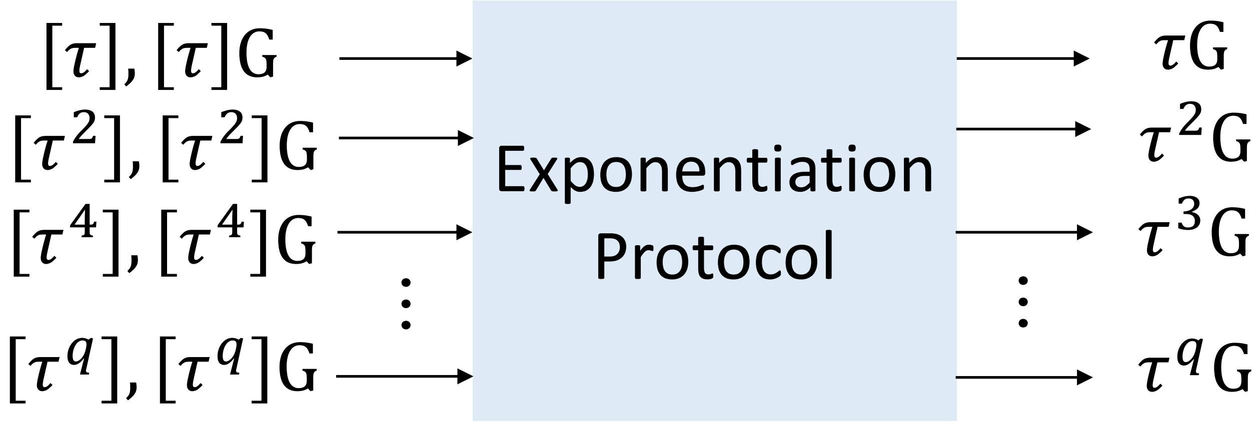


Example: $\tau^5\mathrm{G} = (\tau^{2\cdot 2}\tau)\mathrm{G} = \tau^2(\tau^2\mathrm{G}) \cdot (\tau\mathrm{G})$

Protocol:

1. Each node i publishes $[\tau^2]_i(\tau^2\mathrm{G})$
2. Interpolate $[\tau^2]_i(\tau^2\mathrm{G})$ in the exponent to compute $\tau^2(\tau^2\mathrm{G})$
3. Compute $\tau^2(\tau^2\mathrm{G}) \cdot (\tau\mathrm{G}) = \tau^5\mathrm{G}$

Step 3: All exponents



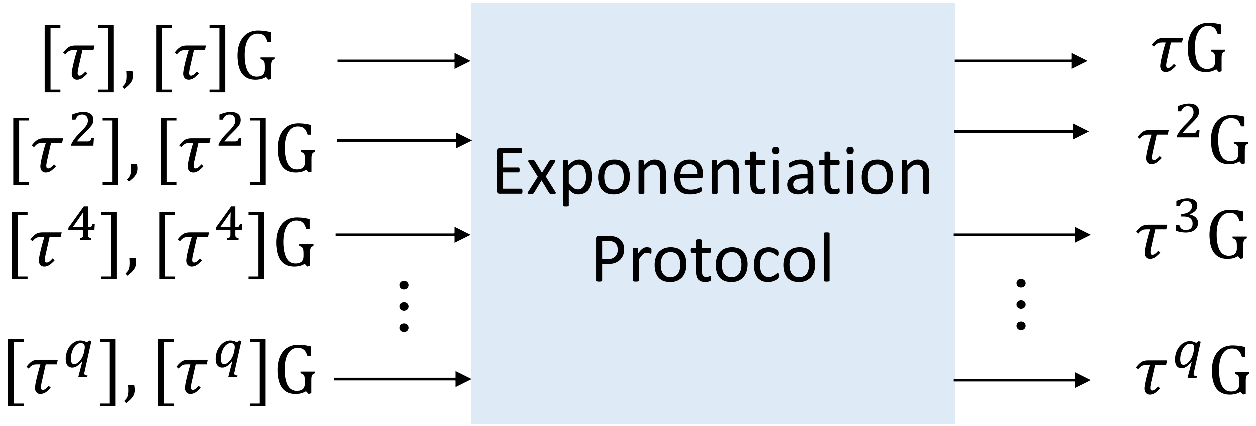
Example: $\tau^5\mathrm{G} = (\tau^{2\cdot 2}\tau)\mathrm{G} = \tau^2(\tau^2\mathrm{G}) \cdot (\tau\mathrm{G})$

Protocol:

1. Each node i publishes $[\tau^2]_i(\tau^2\mathrm{G})$
2. Interpolate $[\tau^2]_i(\tau^2\mathrm{G})$ in the exponent to compute $\tau^2(\tau^2\mathrm{G})$
3. Compute $\tau^2(\tau^2\mathrm{G}) \cdot (\tau\mathrm{G}) = \tau^5\mathrm{G}$

- Naively $\mathrm{O}(n)$ per-party communication per exponent

Step 3: All exponents



Example: $\tau^5\mathrm{G} = (\tau^{2\cdot 2}\tau)\mathrm{G} = \tau^2(\tau^2\mathrm{G}) \cdot (\tau\mathrm{G})$

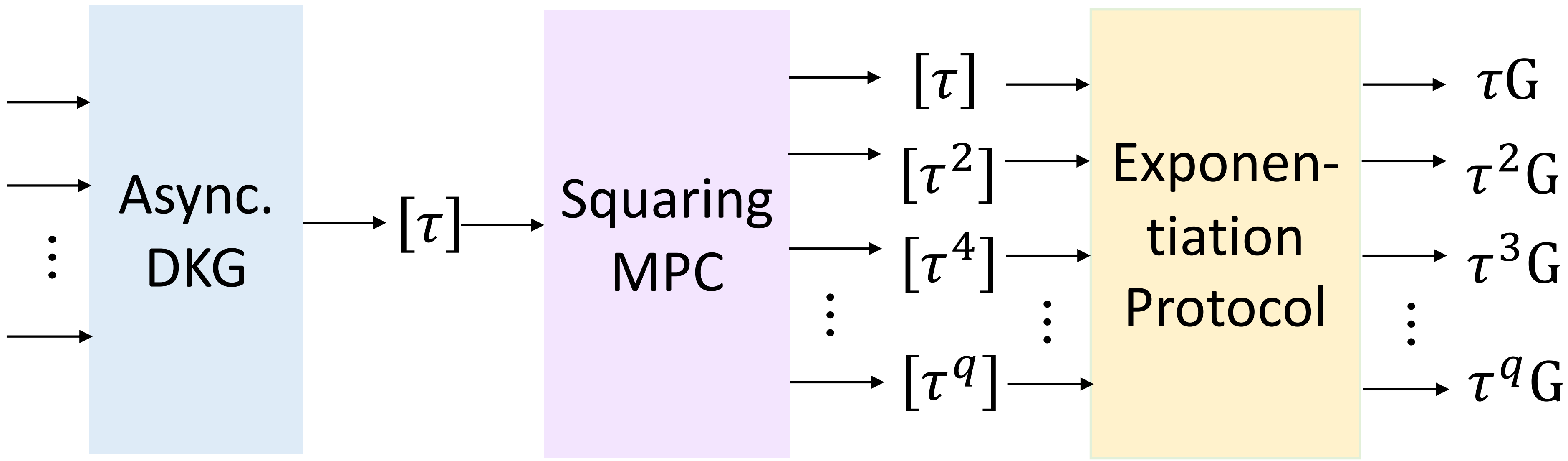
Protocol:

1. Each node i publishes $[\tau^2]_i(\tau^2\mathrm{G})$
2. Interpolate $[\tau^2]_i(\tau^2\mathrm{G})$ in the exponent to compute $\tau^2(\tau^2\mathrm{G})$
3. Compute $\tau^2(\tau^2\mathrm{G}) \cdot (\tau\mathrm{G}) = \tau^5\mathrm{G}$

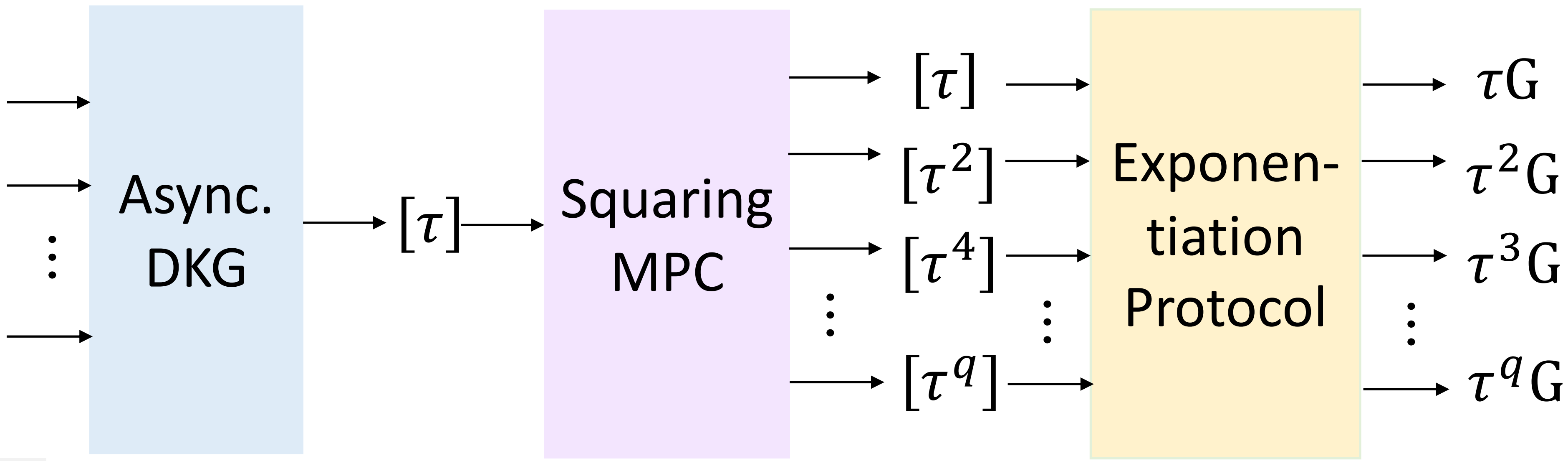
- Naively $\mathrm{O}(n)$ per-party communication per exponent
- Batch amortization optimization to get $\mathrm{O}(1)$ per-party communication cost

Putting together everything

Putting together everything



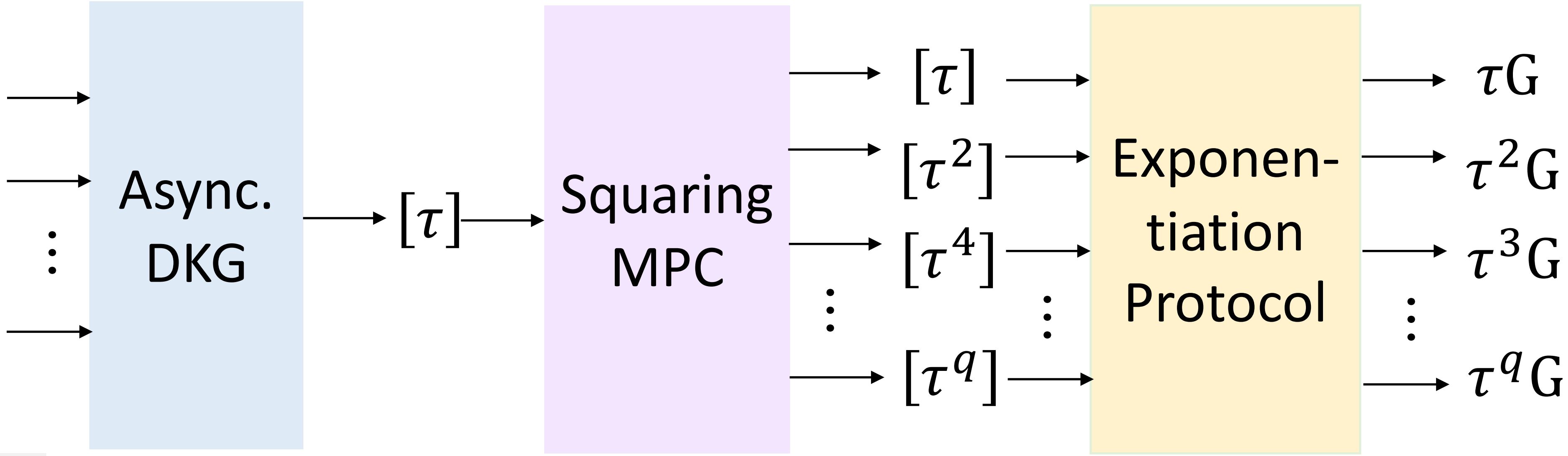
Putting together everything



Comm. Cost:

Rounds:

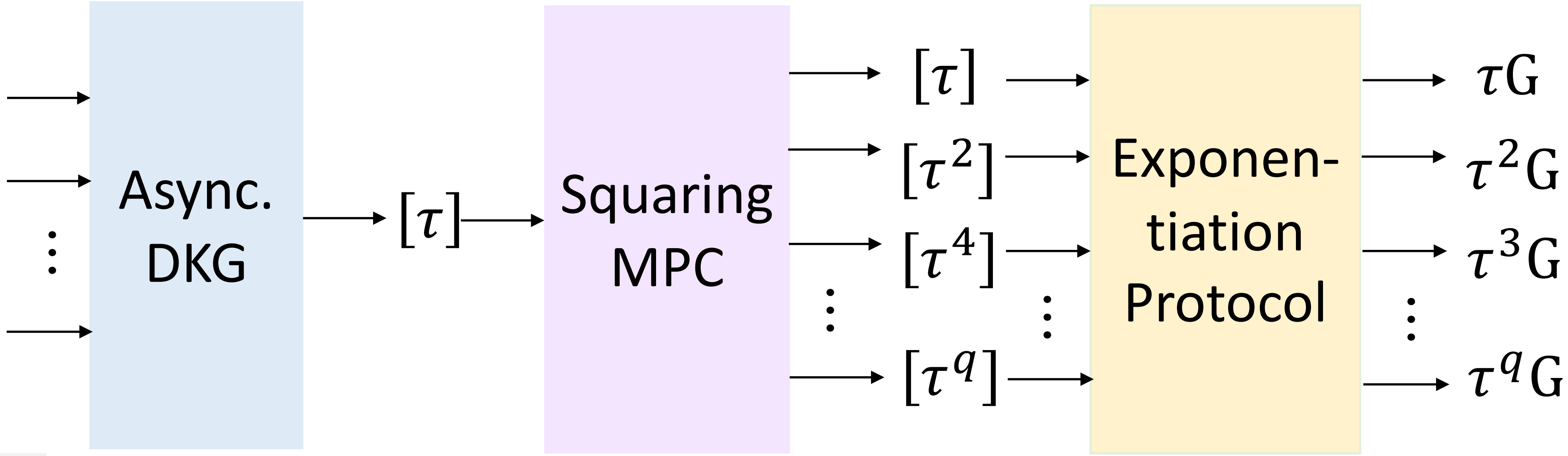
Putting together everything



Comm. Cost: $O(n^2)$

Rounds:

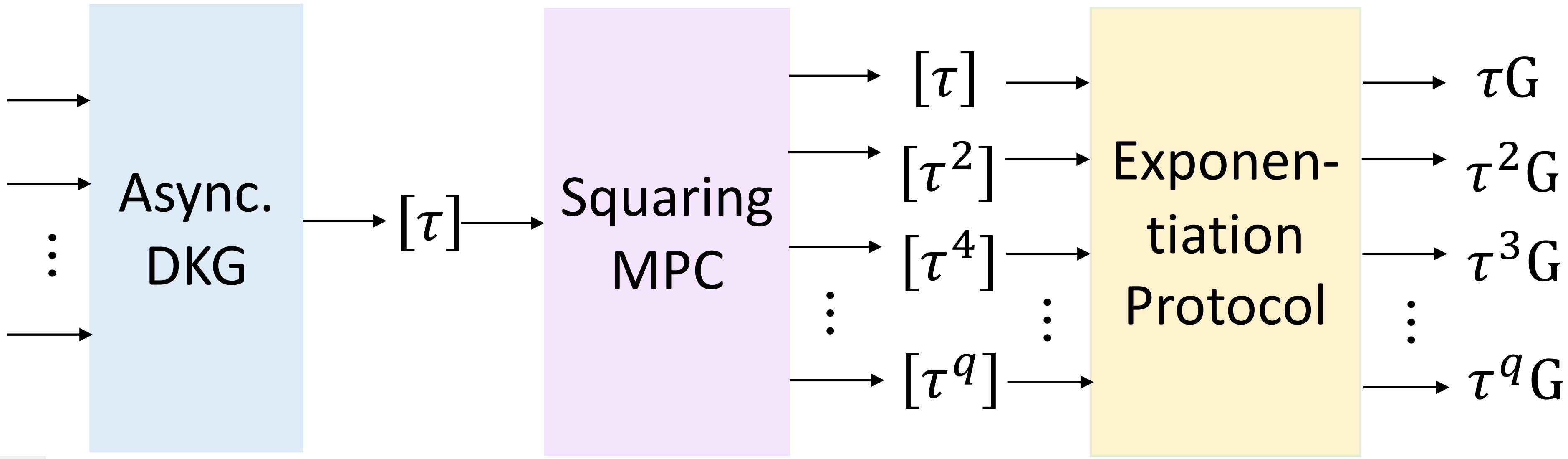
Putting together everything



Comm. Cost: $O(n^2)$

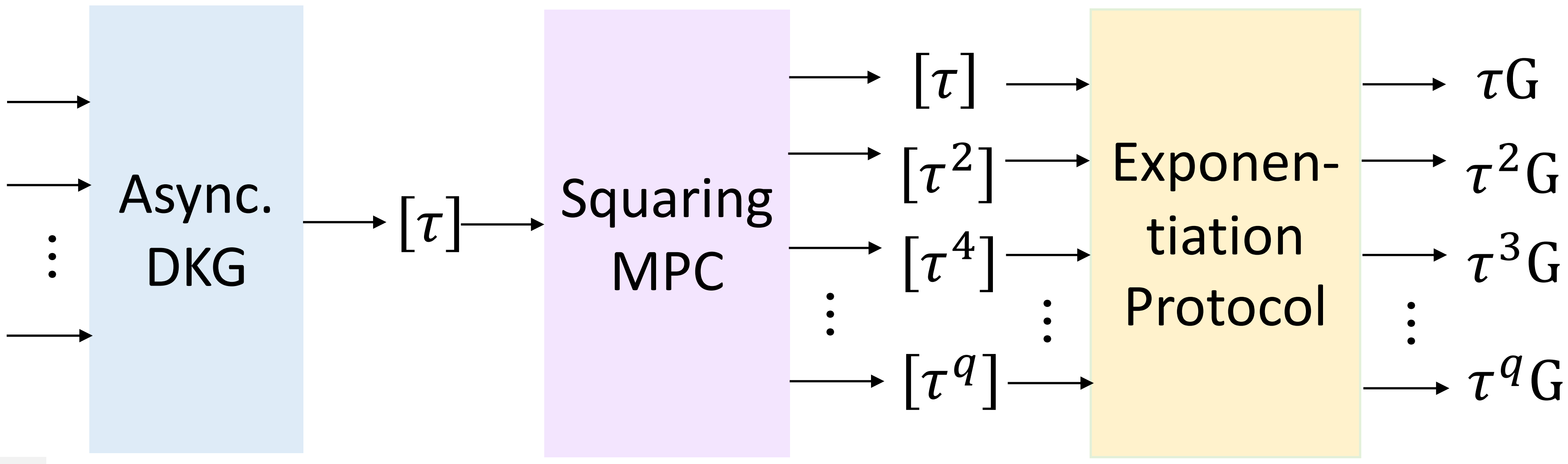
Rounds: $O(\log n)$

Putting together everything



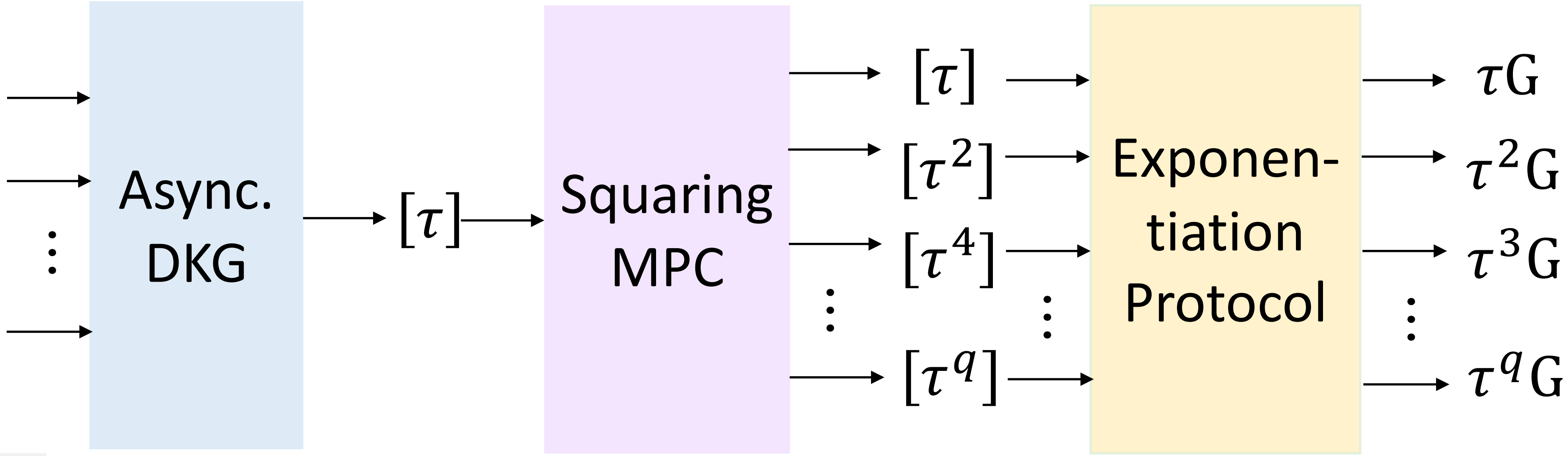
Comm. Cost:	$O(n^2)$	$O(n^2 \log q)$
Rounds:	$O(\log n)$	

Putting together everything



	Async. DKG	Squaring MPC
Comm. Cost:	$O(n^2)$	$O(n^2 \log q)$
Rounds:	$O(\log n)$	$O(\log q)$

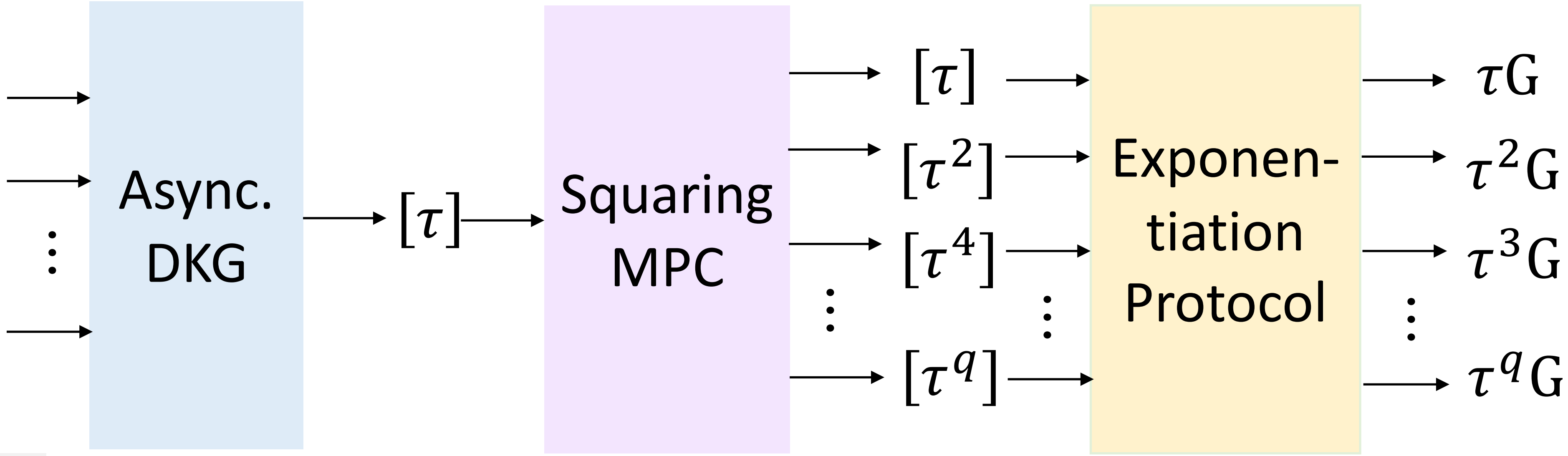
Putting together everything



Comm. Cost: $O(n^2)$ $O(n^2 \log q)$ $O(q)$

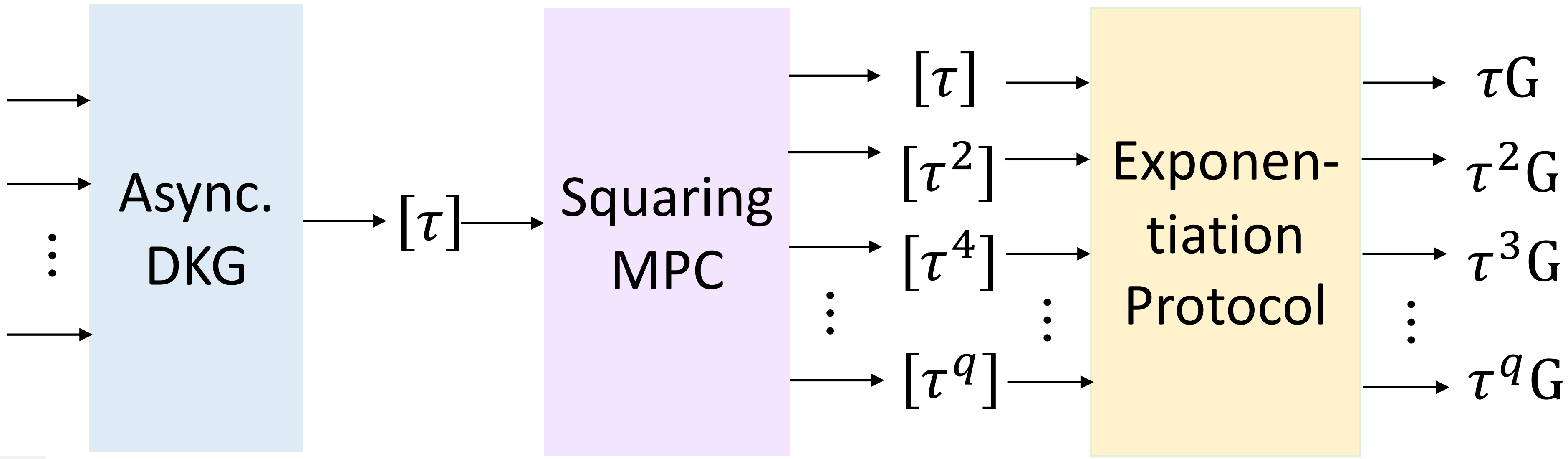
Rounds: $O(\log n)$ $O(\log q)$

Putting together everything



	Async. DKG	Squaring MPC	Exponentiation Protocol
Comm. Cost:	$O(n^2)$	$O(n^2 \log q)$	$O(q)$
Rounds:	$O(\log n)$	$O(\log q)$	$O(\log q)$

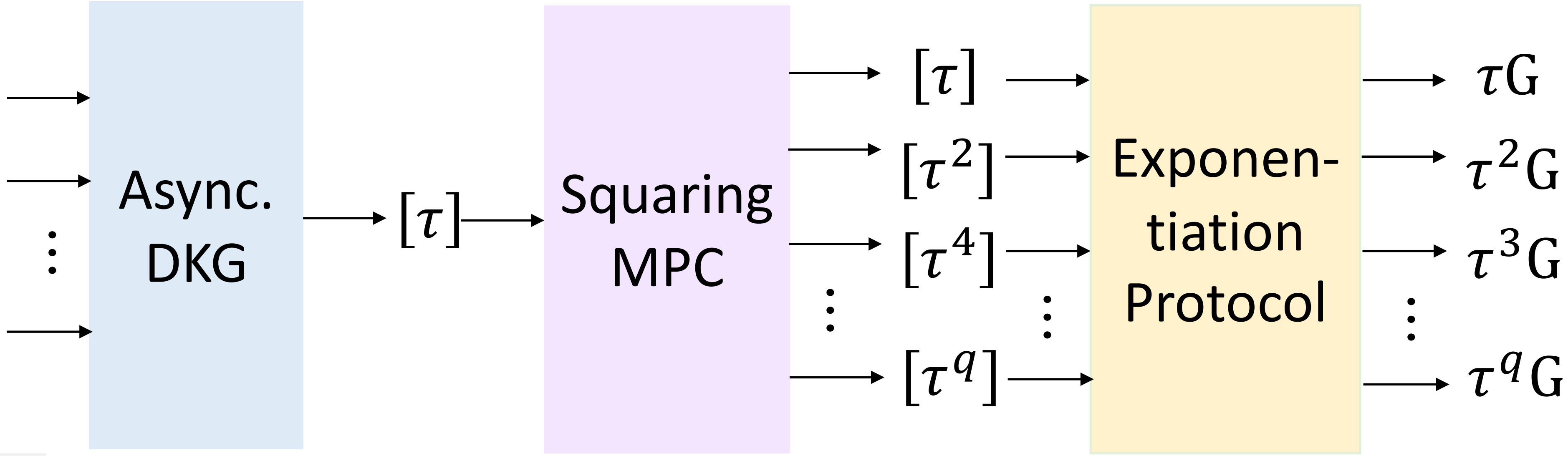
Putting together everything



Comm. Cost:	$O(n^2)$	$O(n^2 \log q)$	$O(q)$
Rounds:	$O(\log n)$	$O(\log q)$	$O(\log q)$

Total per party communication cost: $O(q + n^2 \log q)$

Putting together everything



	Async. DKG	Squaring MPC	Exponentiation Protocol
Comm. Cost:	$O(n^2)$	$O(n^2 \log q)$	$O(q)$
Rounds:	$O(\log n)$	$O(\log q)$	$O(\log q)$

Total per party communication cost: $O(q + n^2 \log q)$

Expected rounds: $O(\log n + \log q)$. Can me made $O(\log q)$

Implementation and Evaluation

Implementation Details

Implementation Details

- Implemented in python with rust for cryptography

Implementation Details

- Implemented in python with rust for cryptography
- Available at https://github.com/sourav1547/qsdh-py

Implementation Details

- Implemented in python with rust for cryptography
- Available at https://github.com/sourav1547/qsdh-py
- Evaluation with up to 128 AWS nodes

Implementation Details

- Implemented in python with rust for cryptography
- Available at https://github.com/sourav1547/qsdh-py
- Evaluation with up to 128 AWS nodes
- Round-robin protocol as baseline

Implementation Details

- Implemented in python with rust for cryptography
- Available at https://github.com/sourav1547/qsdh-py
- Evaluation with up to 128 AWS nodes
- Round-robin protocol as baseline
  - $n|M|$ as bandwidth usage of broadcast

Implementation Details

- Implemented in python with rust for cryptography
- Available at https://github.com/sourav1547/qsdh-py
- Evaluation with up to 128 AWS nodes
- Round-robin protocol as baseline
  - $n|M|$ as bandwidth usage of broadcast
  - Computation cost of broadcast is free

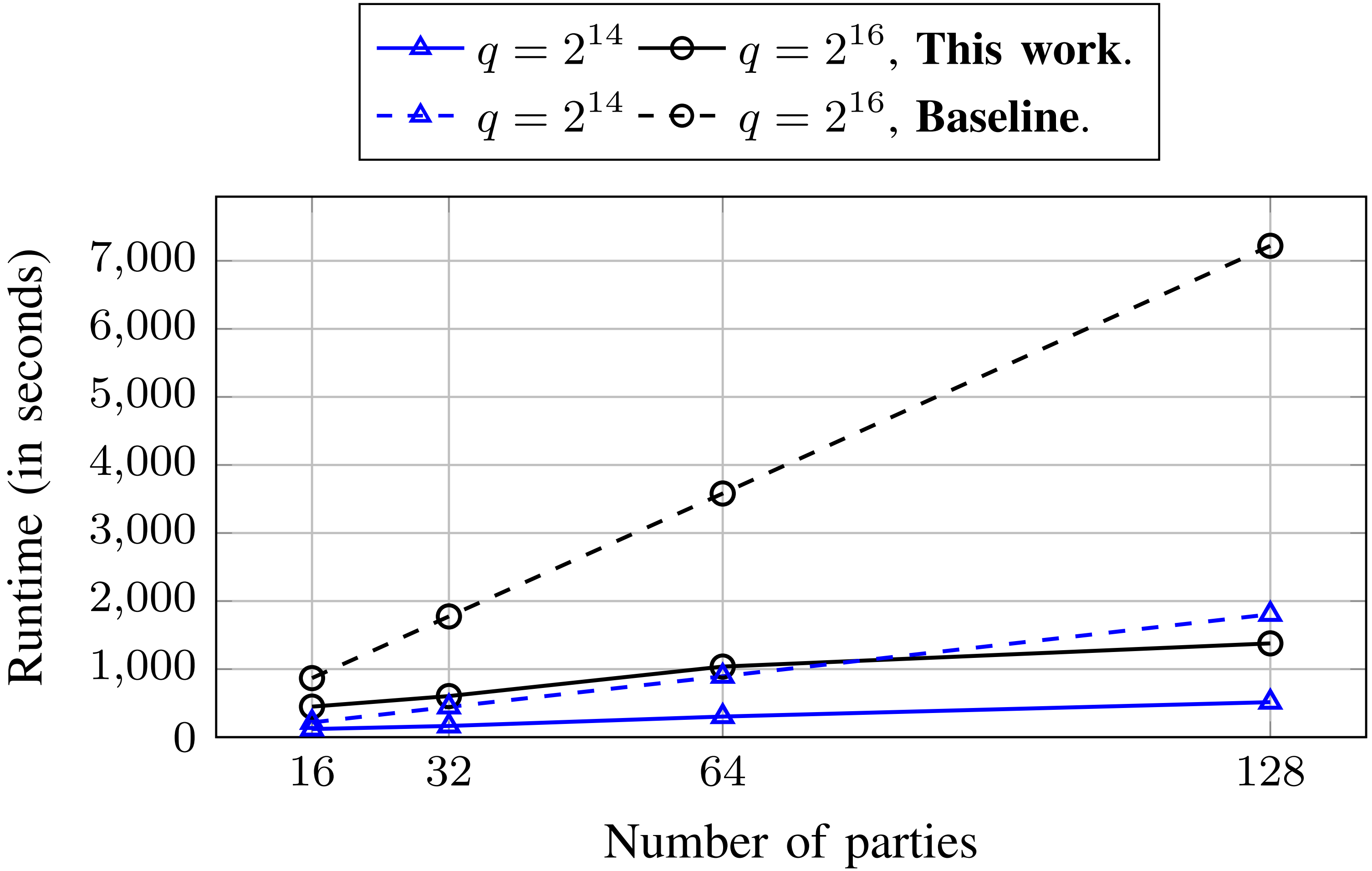
Evaluation results: Runtime

Evaluation results: Runtime

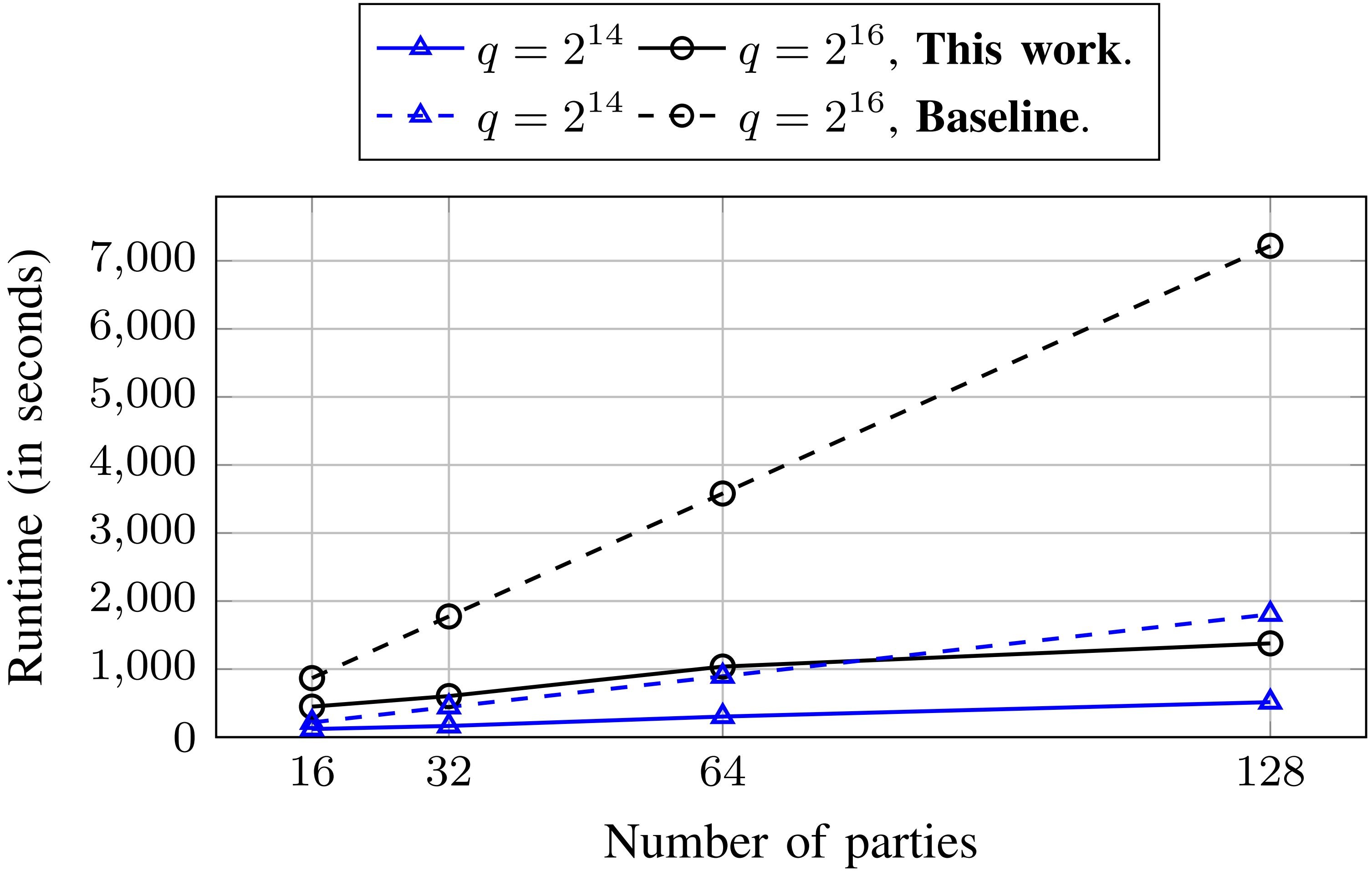
$q = 2^{14}$ $q = 2^{16}$, **This work**.

$q = 2^{14}$ $q = 2^{16}$, **Baseline**.

Evaluation results: Runtime



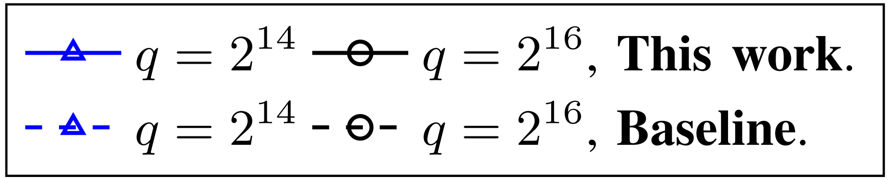
Evaluation results: Runtime



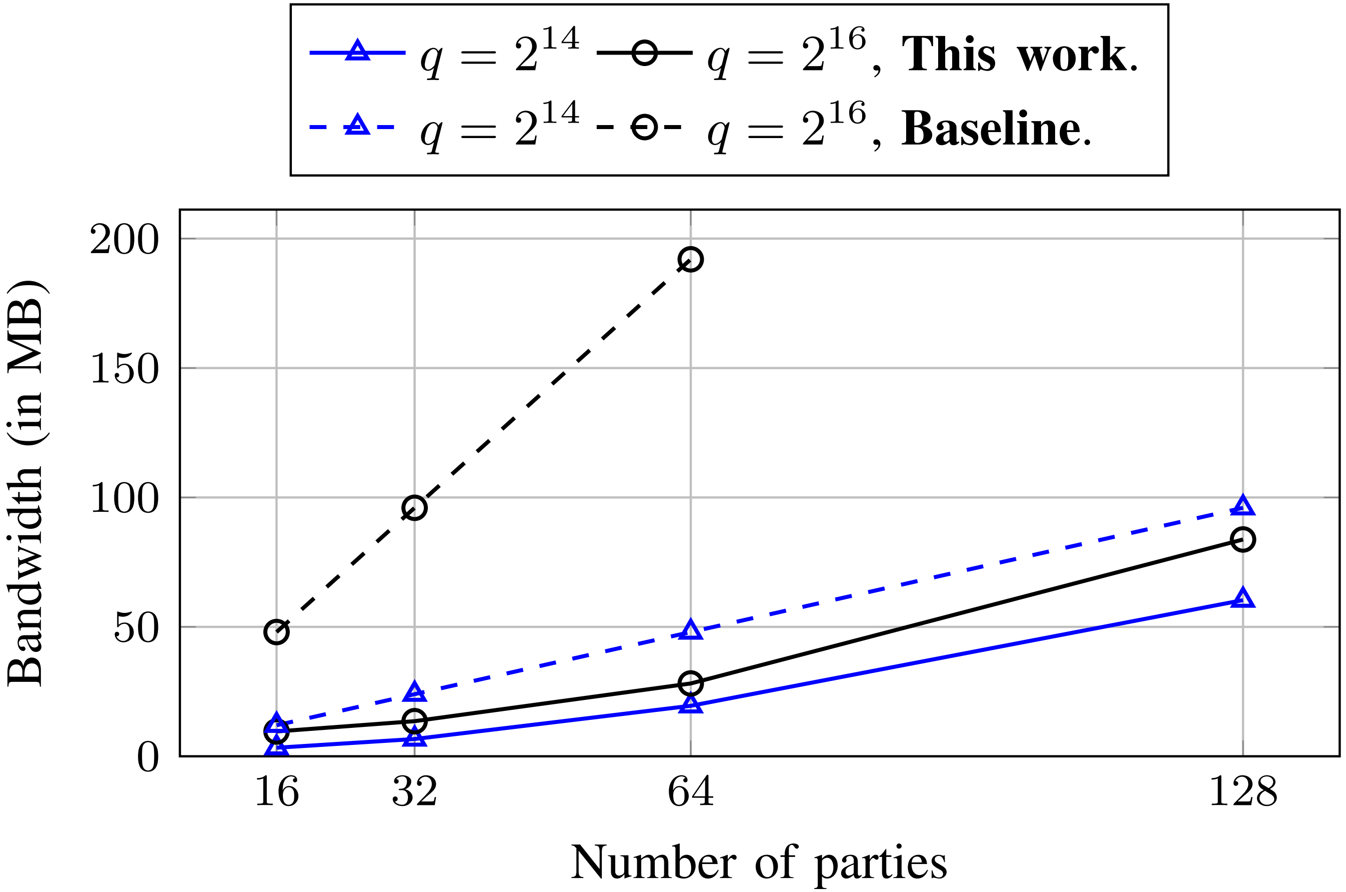
For example, with $q = 2^{16}$, Ours: 1037 seconds, Baseline: 3580 seconds (3.4×)

Evaluation results: Bandwidth Usage

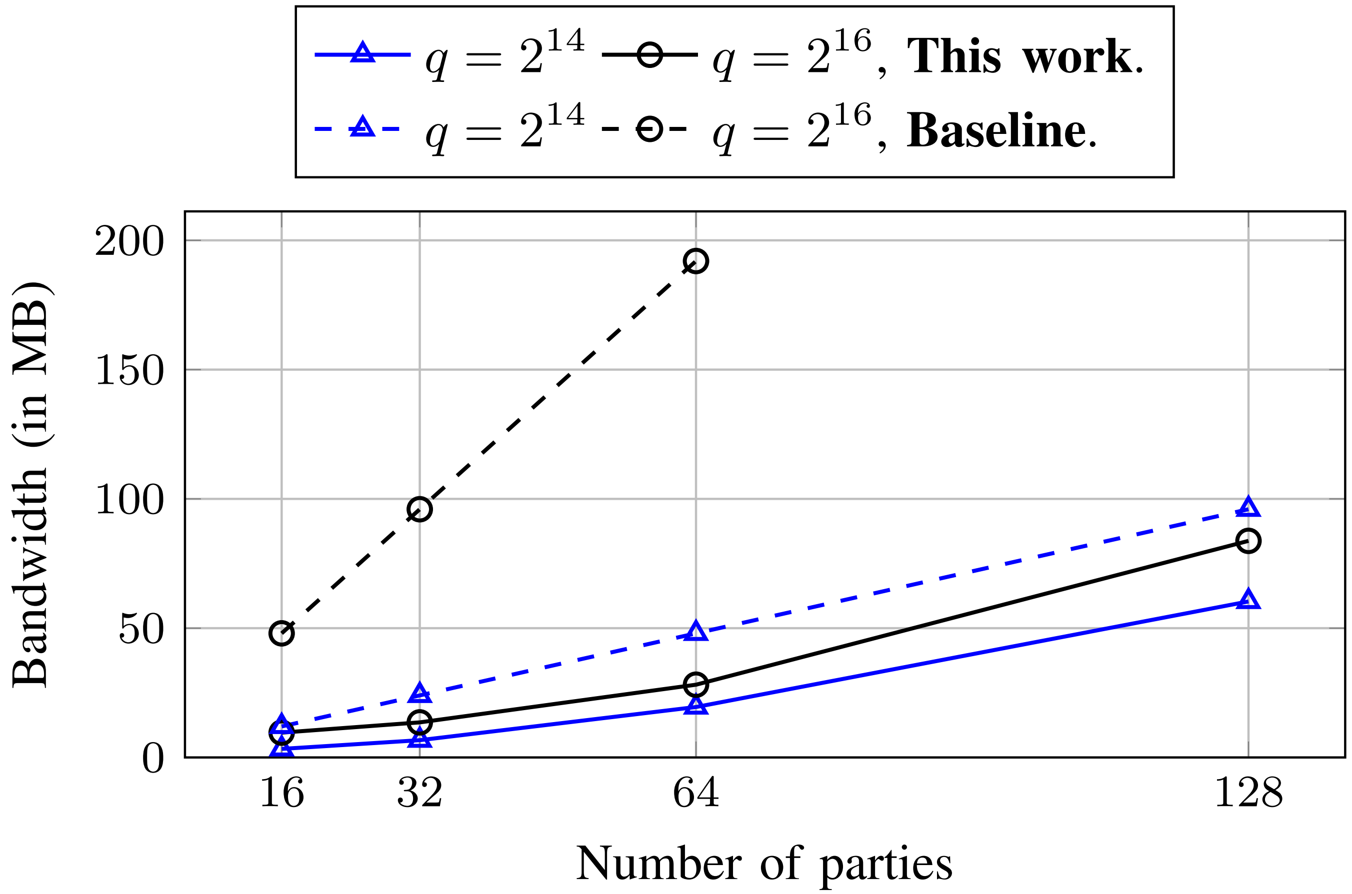
Evaluation results: Bandwidth Usage



Evaluation results: Bandwidth Usage



Evaluation results: Bandwidth Usage



For example, with $q = 2^{16}$, Ours: 13.57 MBytes, Baseline: 96 MBytes (7×)

Summary

Summary

Asynchronous protocol for generating Powers of Tau

Summary

Asynchronous protocol for generating Powers of Tau

Communication Cost (per party)	Computation Cost (per party)	Expected Number of Rounds	Cryptography Assumption
$\mathrm{O}(q + n^2 \log q)$	$\mathrm{O}(q \log n \ \mathbb{G} + n \log q \ \mathbb{P})$	$\mathrm{O}(\log q)$ + ADKG	q-SDH + ADKG

Summary

Asynchronous protocol for generating Powers of Tau

Communication Cost (per party)	Computation Cost (per party)	Expected Number of Rounds	Cryptography Assumption
$\mathrm{O}(q + n^2 \log q)$	$\mathrm{O}(q \log n \ \mathbb{G} + n \log q \ \mathbb{P})$	$\mathrm{O}(\log q)$ + ADKG	q-SDH + ADKG

See paper for:

Summary

Asynchronous protocol for generating Powers of Tau

Communication Cost (per party)	Computation Cost (per party)	Expected Number of Rounds	Cryptography Assumption
$O(q + n^2 \log q)$	$O(q \log n \, \mathbb{G} + n \log q \, \mathbb{P})$	$O(\log q)$ + ADKG	q-SDH + ADKG

See paper for:

- Batch amortization optimization

Summary

Asynchronous protocol for generating Powers of Tau

Communication Cost (per party)	Computation Cost (per party)	Expected Number of Rounds	Cryptography Assumption
$O(q + n^2 \log q)$	$O(q \log n \ \mathbb{G} + n \log q \ \mathbb{P})$	$O(\log q)$ + ADKG	q-SDH + ADKG

See paper for:

- Batch amortization optimization
- Running DKG and multiplication unit generation protocol in parallel

Summary

Asynchronous protocol for generating Powers of Tau

Communication Cost (per party)	Computation Cost (per party)	Expected Number of Rounds	Cryptography Assumption
$O(q + n^2 \log q)$	$O(q \log n \ \mathbb{G} + n \log q \ \mathbb{P})$	$O(\log q)$ + ADKG	q-SDH + ADKG

See paper for:

- Batch amortization optimization
- Running DKG and multiplication unit generation protocol in parallel
- Evaluation breakdown of each phases

Summary

Asynchronous protocol for generating Powers of Tau

Communication Cost (per party)	Computation Cost (per party)	Expected Number of Rounds	Cryptography Assumption
$\mathrm{O}(q + n^2 \log q)$	$\mathrm{O}(q \log n \ \mathbb{G} + n \log q \ \mathbb{P})$	$\mathrm{O}(\log q)$ + ADKG	q-SDH + ADKG

See paper for:

- Batch amortization optimization
- Running DKG and multiplication unit generation protocol in parallel
- Evaluation breakdown of each phases

Thank You (souravd2@Illinois.edu)