EPFL

MPC PEERSTER

KEEPING DATA PRIVATE, TOGETHER

TEAM NGO

Yuening Yang
Chen Chang Lew
Junzhen Lou

1

INTRODUCTION

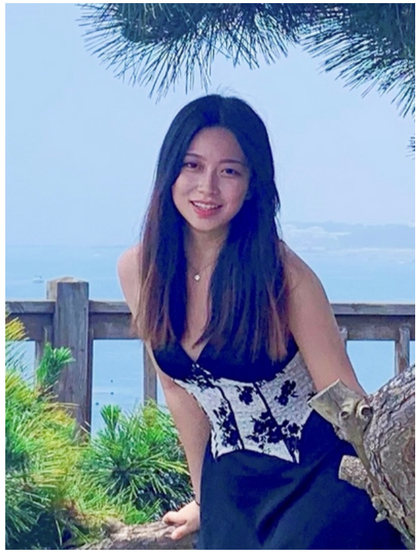
MPC PEERSTER
TEAM NGO

MOTIVATION

- Data is becoming increasingly valuable Nowadays.
- The need to preserve data privacy while maintaining availability is becoming a critical issue.
- Traditional methods for data sharing are no longer sufficient for protecting sensitive information.
- Provide a platform to raise incentives for people to use mpc

PROJECT OVERVIEW

A p2p community with access control, where participants can start an MPC, sell data for MPC to use, participate in MPC to earn fees, etc.



Junzhen Lou

MPC Protocols

Finite Field Arithmetic

Benchmark

Chen Chang Lew

MPC Progress

Benchmark

UI

Yuening Yang

Secure channel

Blockchain

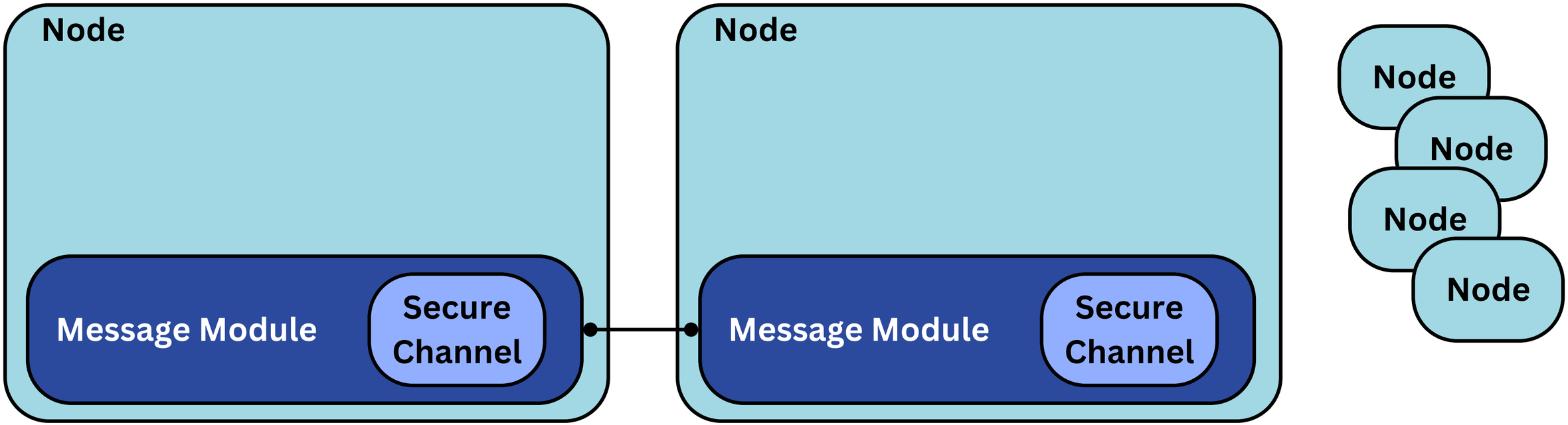
CLI

2

SYSTEM OVERVIEW

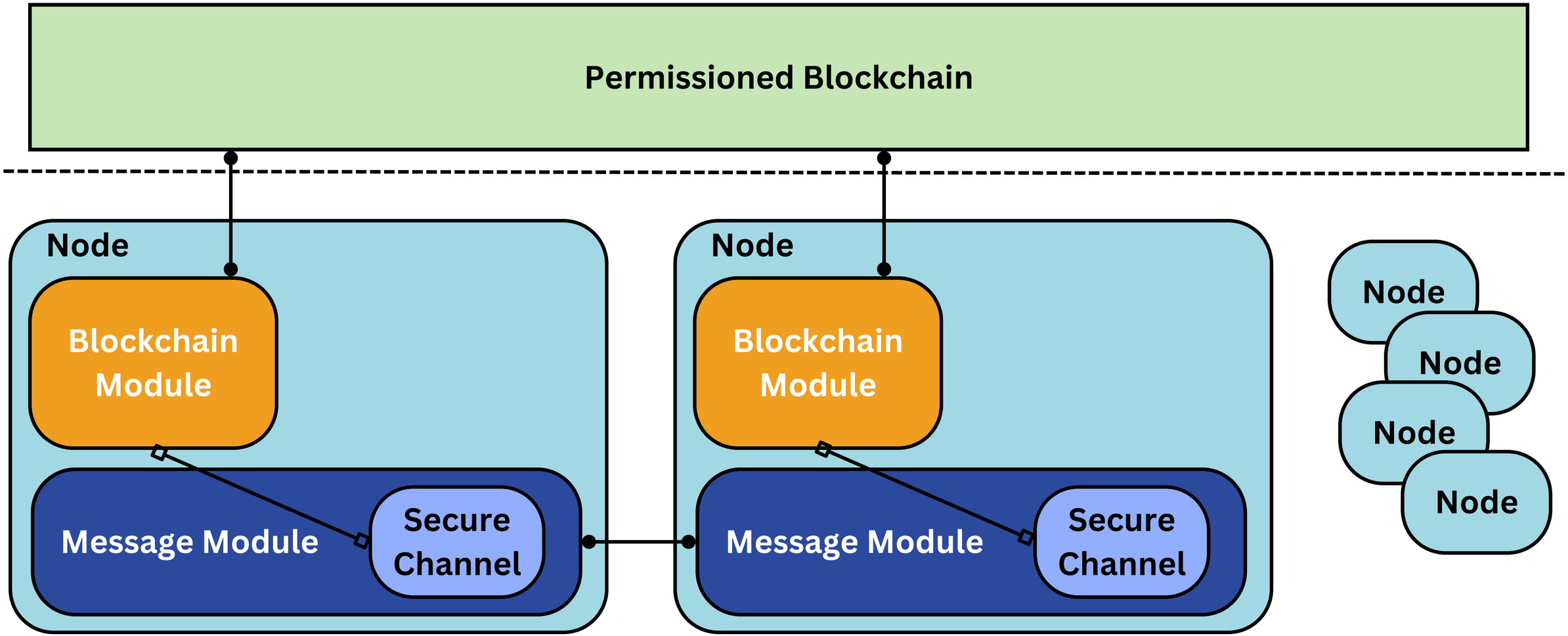
MPC PEERSTER

SYSTEM ARCHITECTURE



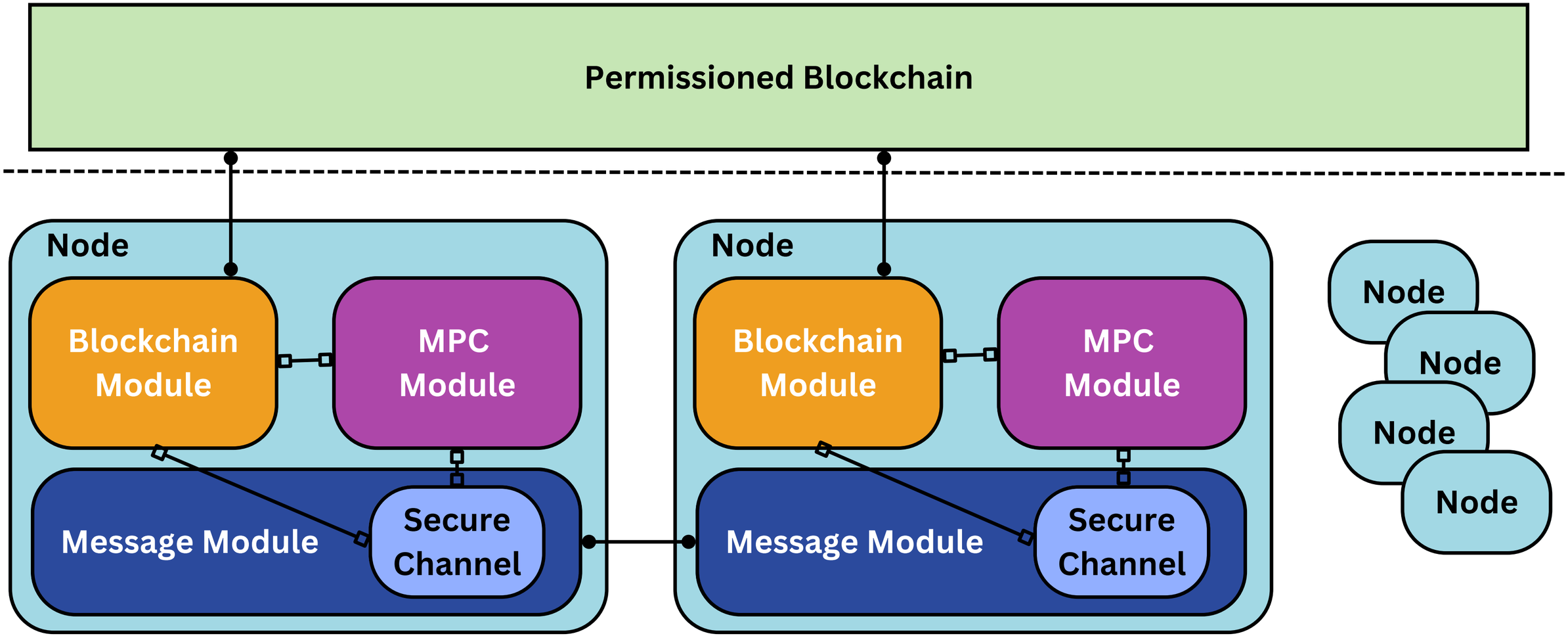
- Secure channel supports end-to-end encryption and group private broadcast

SYSTEM ARCHITECTURE



- Blockchain Module is built on top of secure channel and form a permissioned blockchain

SYSTEM ARCHITECTURE



- MPC Module is built on top of secure channel

MULTI-PARTY COMPUTATION

Overview

- Privacy-preserving distributed computation
- Secure and efficient data/information exchange
- allows parties to perform secure computation while keeping their own input private

Junzhen Lou

MULTI-PARTY COMPUTATION

Overview

- Privacy-preserving distributed computation
- Secure and efficient data/information exchange
- allows parties to perform secure computation while keeping their own input private

Model

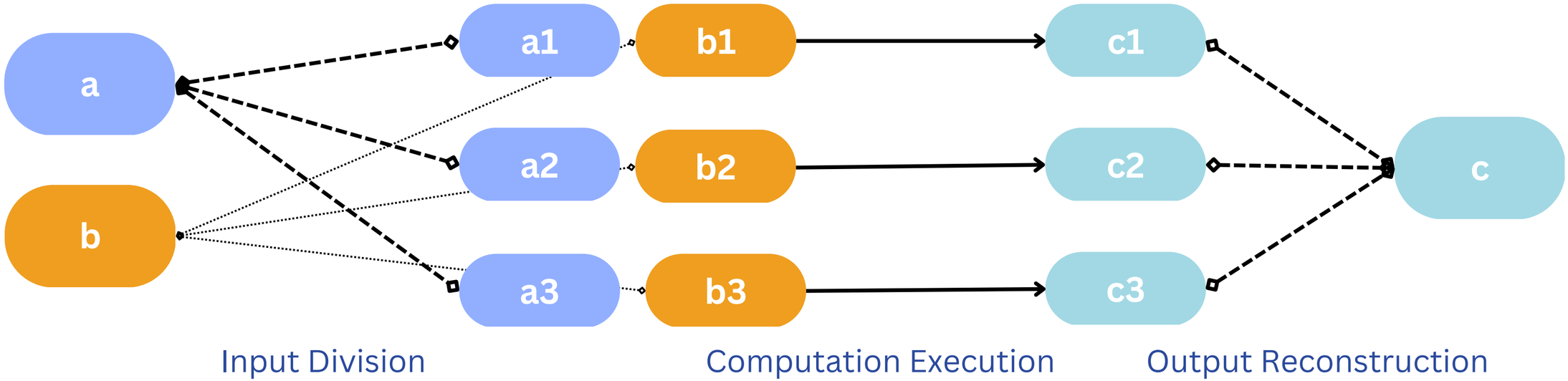
- Complete network
- **Secure channels with broadcast**
- **Synchronous communication**
- **Passive adversary**

Junzhen Lou

MULTI-PARTY COMPUTATION

Steps

- Function Identification
- **Input Division**
- **Computation Execution**
- **Output Reconstruction**

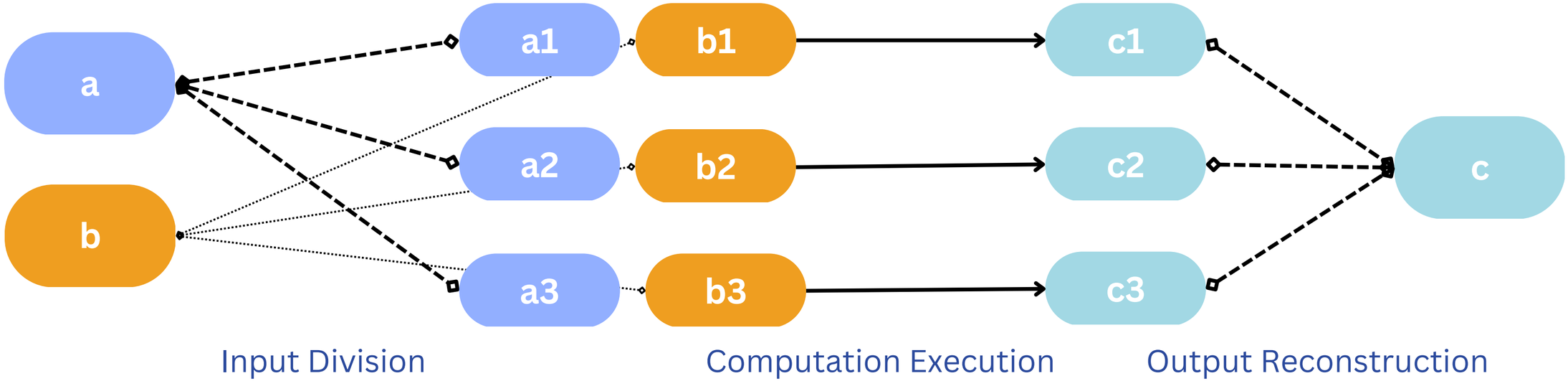


Junzhen Lou

MULTI-PARTY COMPUTATION

Steps

- Function Identification
- Input Division - **Shamir Secret Sharing Scheme**
- Computation Execution - **Secure Addition & Multiplicaiton Protocols**
- Output Reconstruction - **Lagrange Interpolation**



Junzhen Lou

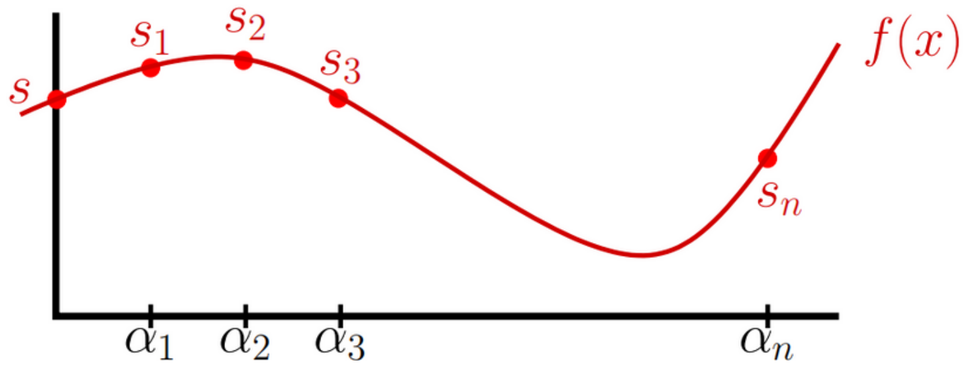
MULTI-PARTY COMPUTATION

Input Division - Shamir Secret Sharing Scheme

- Threshold secret sharing scheme
- Based on polynomials
- Linear (addictive homomorphism)

Idea

- **Random polynomial *f* of degree *d* is defined by *d + 1* points**
- ***s* = *f*(0) = secret, party *Pi* gets share *si* = *f*(*αi*) for fixed *αi***
- **Degree *d* = *k*−1 ⇒ *k* parties can reconstruct, *k*−1 cannot**



n parties, k needed for reconstruction

Junzhen Lou

MULTI-PARTY COMPUTATION

Output Reconstruction - Lagrange Interpolation

- Reconstruct a value from its shares

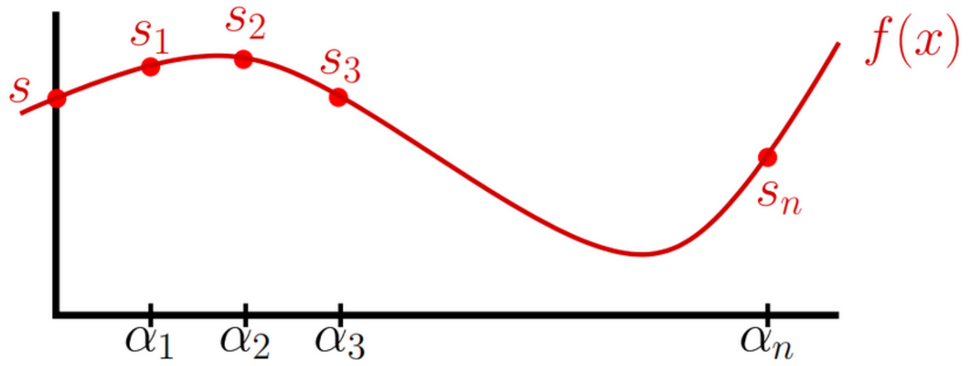
Protocol

- ∀Pi : send si to P
- ***P*: compute *s* with Lagrange interpolation**

$$s = \sum_{i=1}^{n} w_i s_i,\ w_i = \lambda_i(0) = \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{-\alpha_j}{\alpha_i - \alpha_j}.$$

Privacy

- **adversary with < *k* shares obtains no information about *s***



n parties, k needed for reconstruction

Junzhen Lou

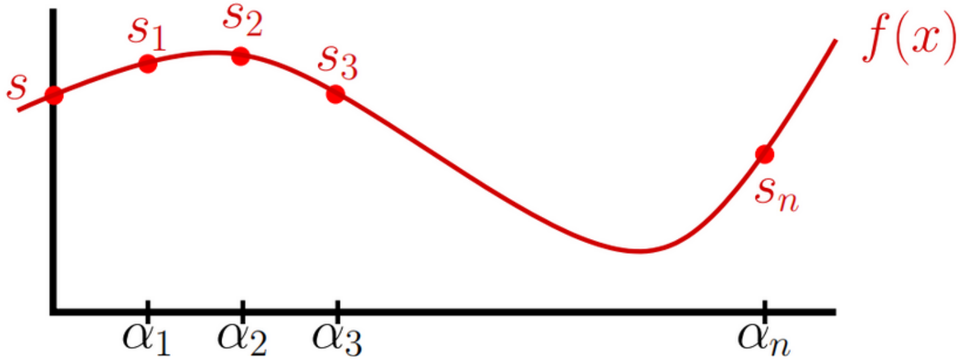
MULTI-PARTY COMPUTATION

Computation Execution - Secure Addition Protocol

- Shamir-Sharing is linear
- apply linear function on shares ⇒ apply the same function on original secret values

Protocol

- **a**, **b**, . . . shared by **a1, ..., an**, **b1, ..., bn**, etc.
- Every **Pi** computes **ci = *f*(ai , bi , . . .)**
- **c1, ..., cn** is a sharing of **c = *f*(a, b, . . .)**



n parties, k needed for reconstruction

Junzhen Lou

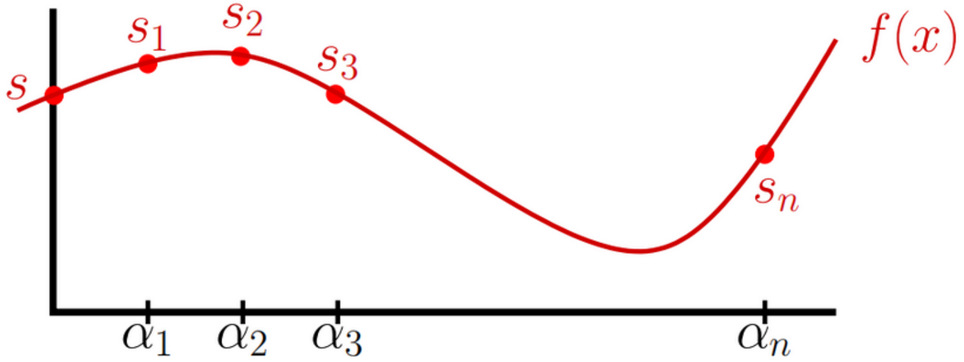
MULTI-PARTY COMPUTATION

Computation Execution - Secure Multiplication Protocol

- multiply two secrets ⇒ multiply two polynomials
- doubles the degree

Protocol

- **0. *a*, *b* are shared by *a*1, ..., *an*, *b*1, ..., *bn***
- **1. ∀ *Pi* : compute *di* = *ai* * *bi***
- **2. ∀ *Pi* : share *di* → *di*1, . . . , *din***
- **3. ∀ *Pj* : compute *cj* = *L*(*d*1*j* , . . . , *dnj*),**
- *L()*: Lagrange Interpolation

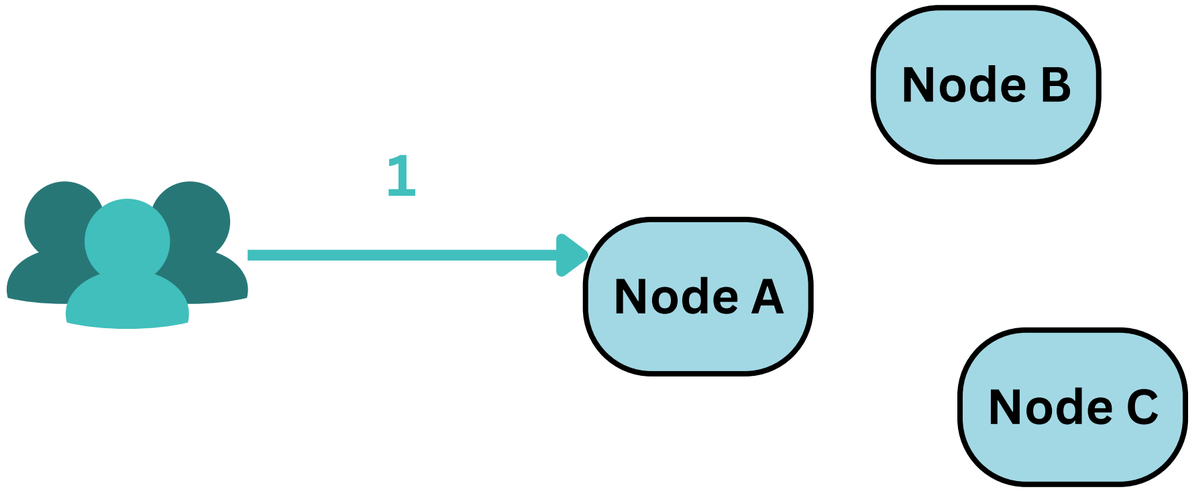


n parties, k needed for reconstruction

Junzhen Lou

MPC INTERGRATION

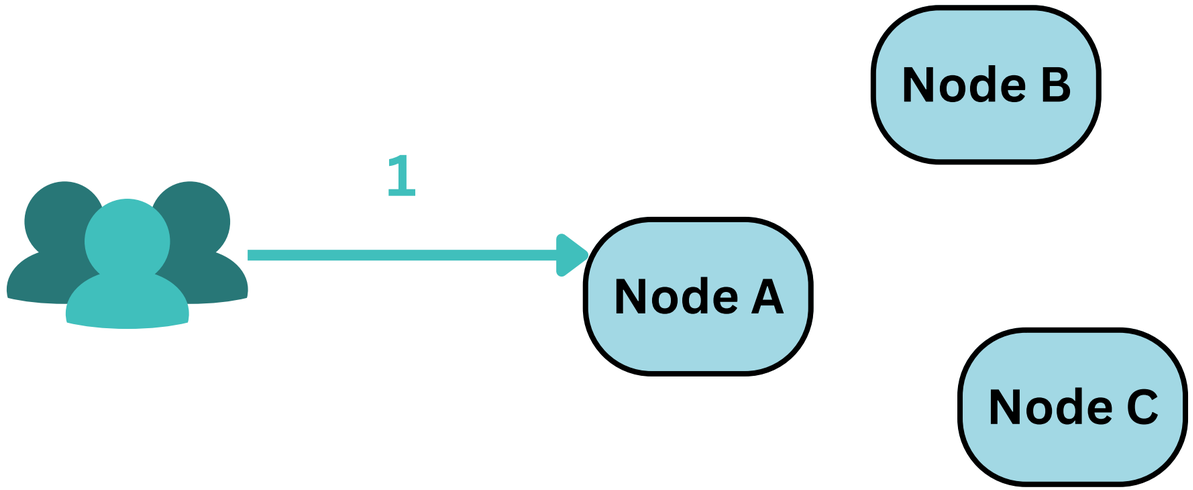
1. Users send an Expression for MPC to count. Ex: a+c



Chen Chang Lew

MPC INTERGRATION

1. Users send an Expression for MPC to count. Ex: a+c
2. Change the infix expression to postfix and identify which assets need to receive from others.

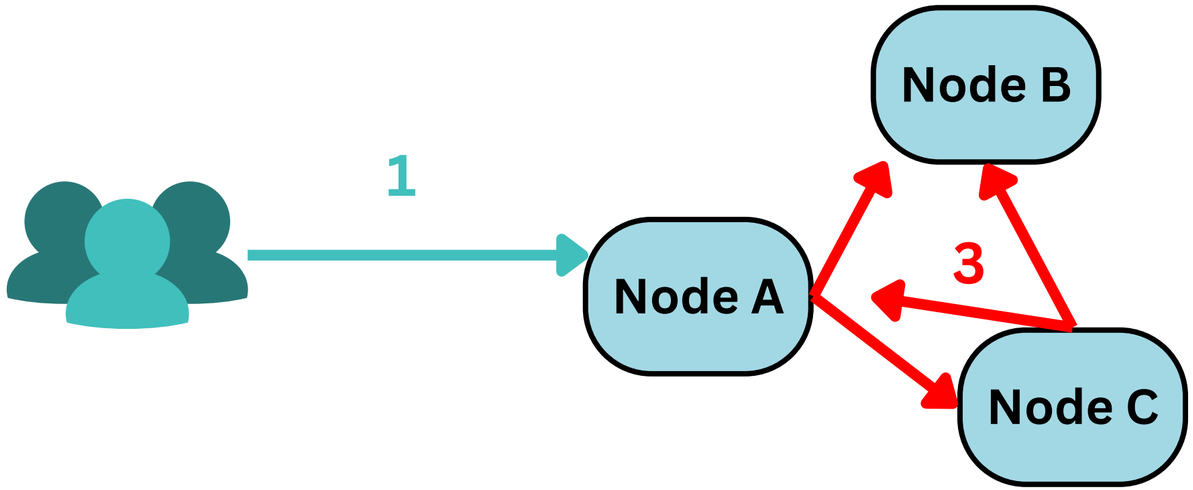


2
a+c --> a c +

Chen Chang Lew

MPC INTERGRATION

1. Users send an Expression for MPC to count. Ex: a+c
2. Change the infix expression to postfix and identify which assets need to receive from others.
3. SSS the assets that are needed in this round. *

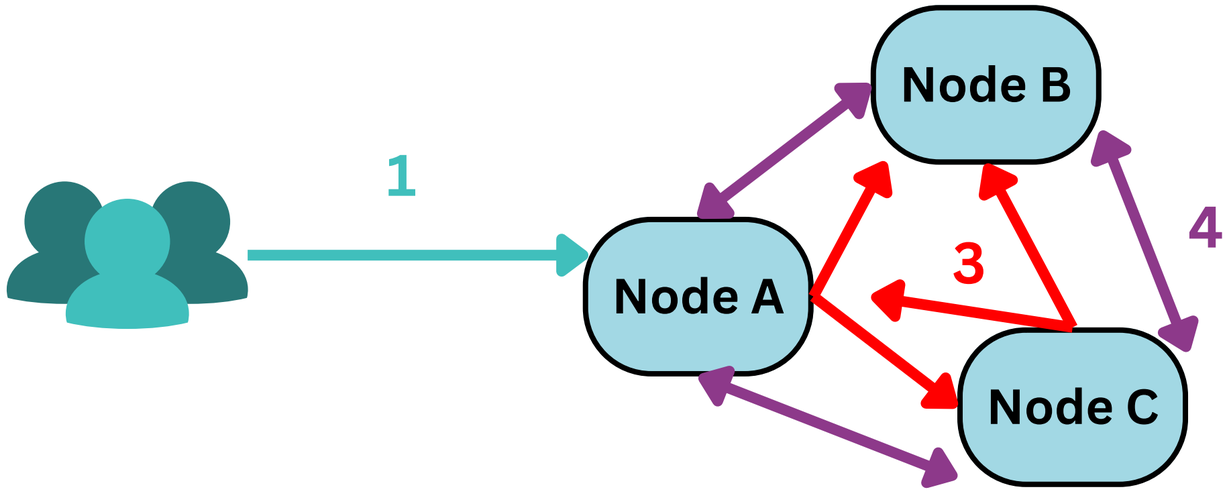


2
a+c --> a c +

Chen Chang Lew

MPC INTERGRATION

1. Users send an Expression for MPC to count. Ex: a+c
2. Change the infix expression to postfix and identify which assets need to receive from others.
3. SSS the assets that are needed in this round. *
4. Boardcast Interpolation Msg to reconstruct the required value. *

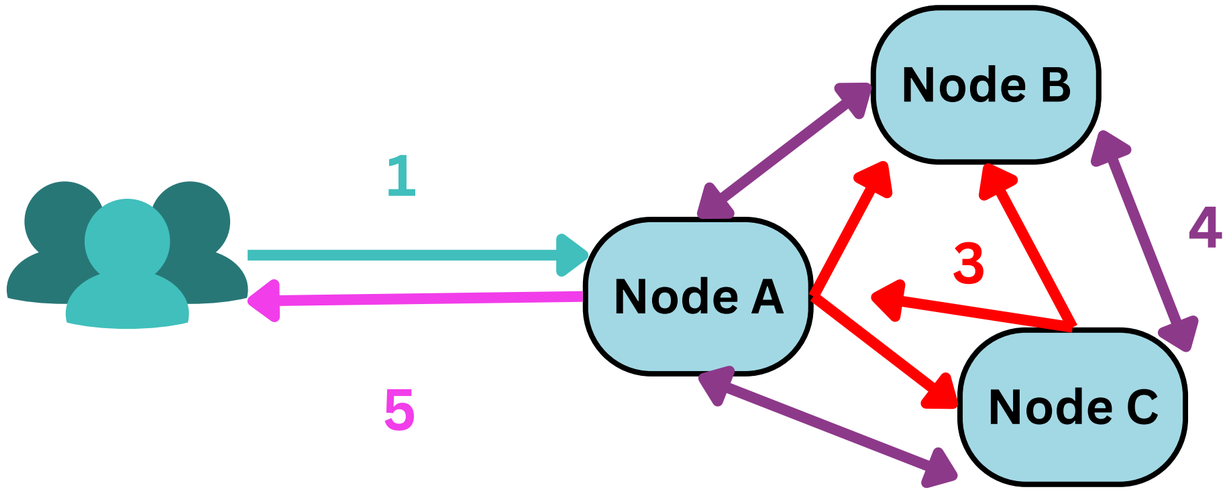


2
a+c --> a c +

Chen Chang Lew

MPC INTERGRATION

1. Users send an Expression for MPC to count. Ex: a+c
2. Change the infix expression to postfix and identify which assets need to receive from others.
3. SSS the assets that are needed in this round. *
4. Boardcast Interpolation Msg to reconstruct the required value. *
5. Reconstruct the final value using Lagrange Interpolation and return result.



2
a+c --> a c +

Chen Chang Lew

PERMISSIONED BLOCKCHAIN

Overview

- A decentralized ledger. An Ethereum-like state machine
- In-Chain configuration specifies participants and other policies
- Support execution of specific types of transactions

Yuening Yang

PERMISSIONED BLOCKCHAIN

Overview

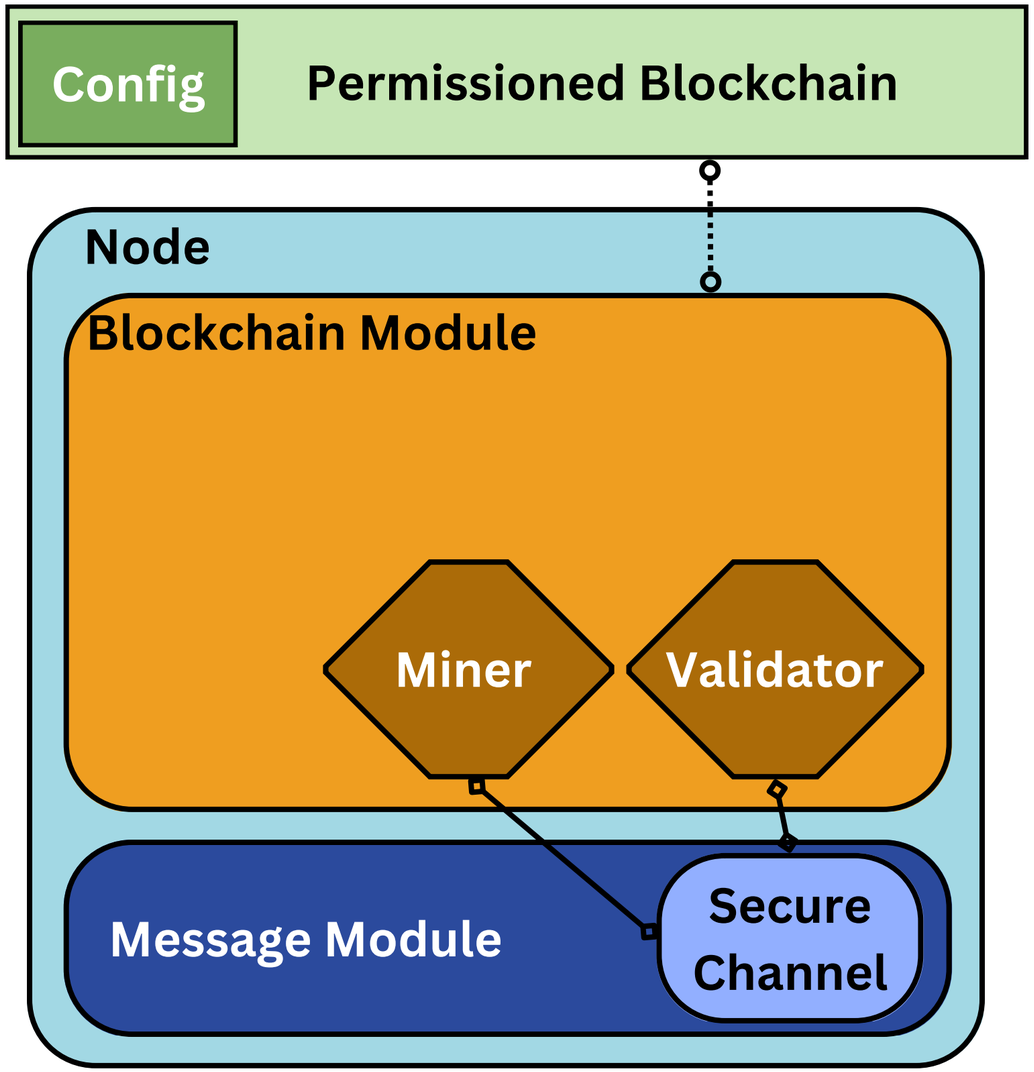
- A decentralized ledger. An Ethereum-like state machine
- In-Chain configuration specifies participants and other policies
- Support execution of specific types of transactions

Purpose

- Create an incentive for parties to provide their data in MPC
- Filter adversaries for MPC
- Facilitate consensus on network assets and MPC participants as well as distribution of encryption keys

Yuening Yang

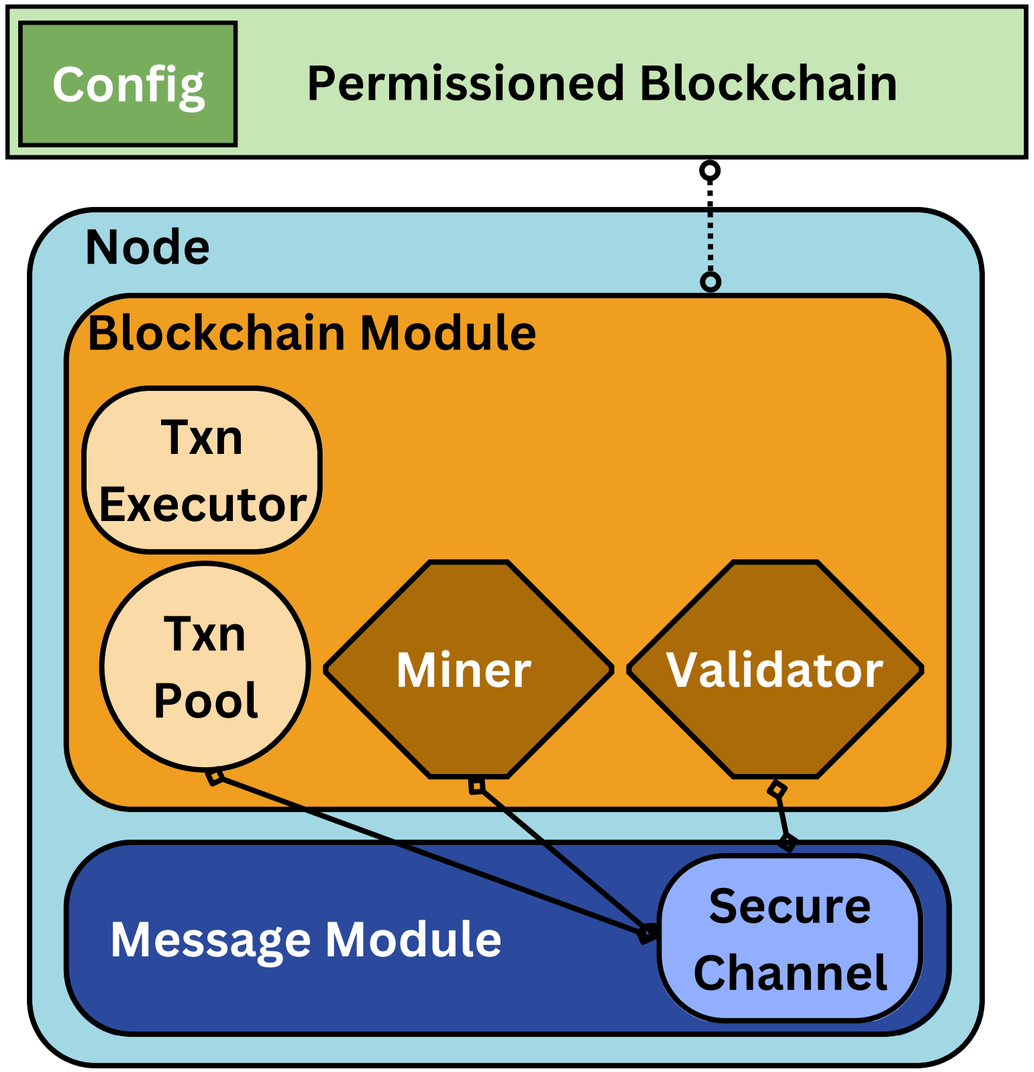
PERMISSIONED BLOCKCHAIN



- **Miner**: a daemon for mining and broadcasting new blocks
- **Validator**: a daemon for validating blocks and appending them to the blockchain

Yuening Yang

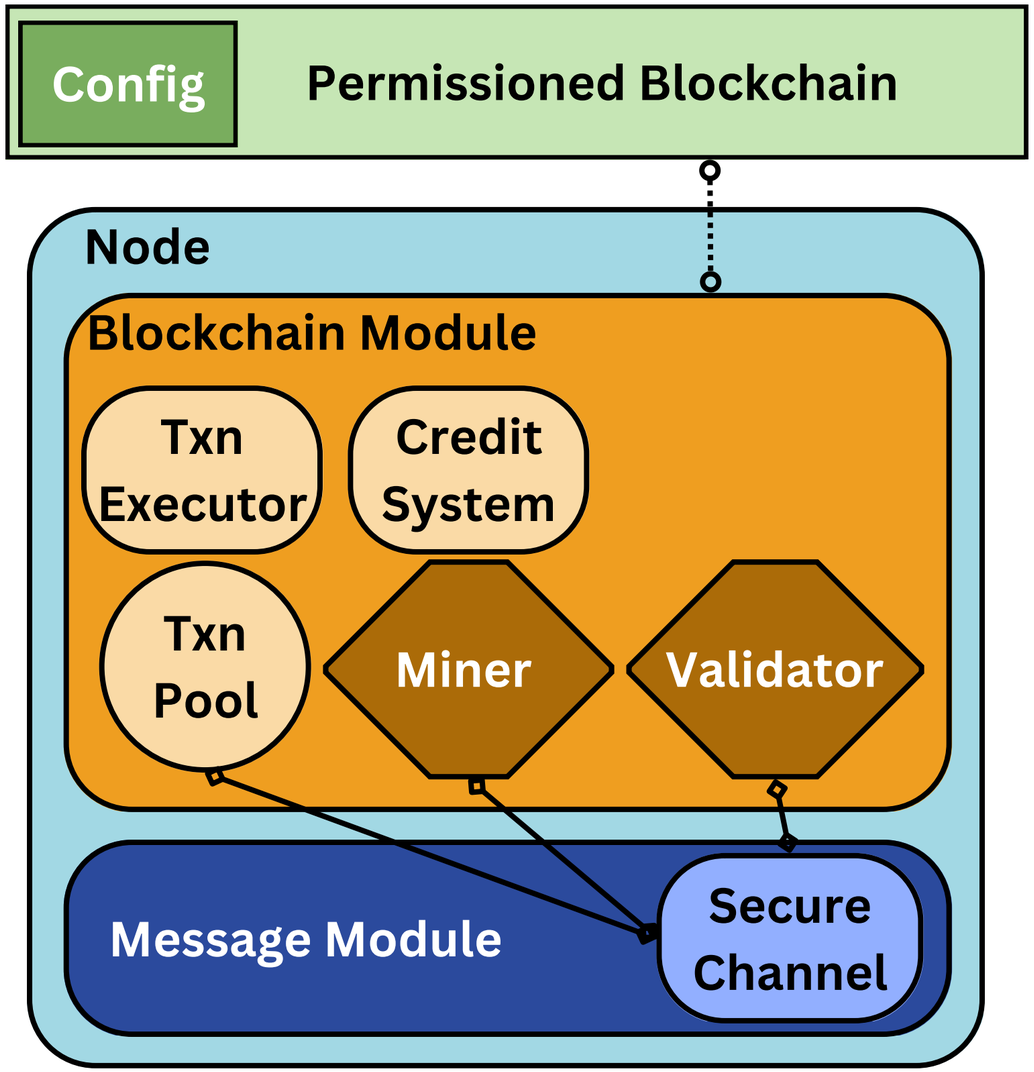
PERMISSIONED BLOCKCHAIN



- **Miner**: a daemon for mining and broadcasting new blocks
- **Validator**: a daemon for validating blocks and appending them to the blockchain
- **Txn Pool**: cache unprocessed txns between Message Module and miner
- **Txn Executor**: validate and execute txn. Only support dedicated txn types

Yuening Yang

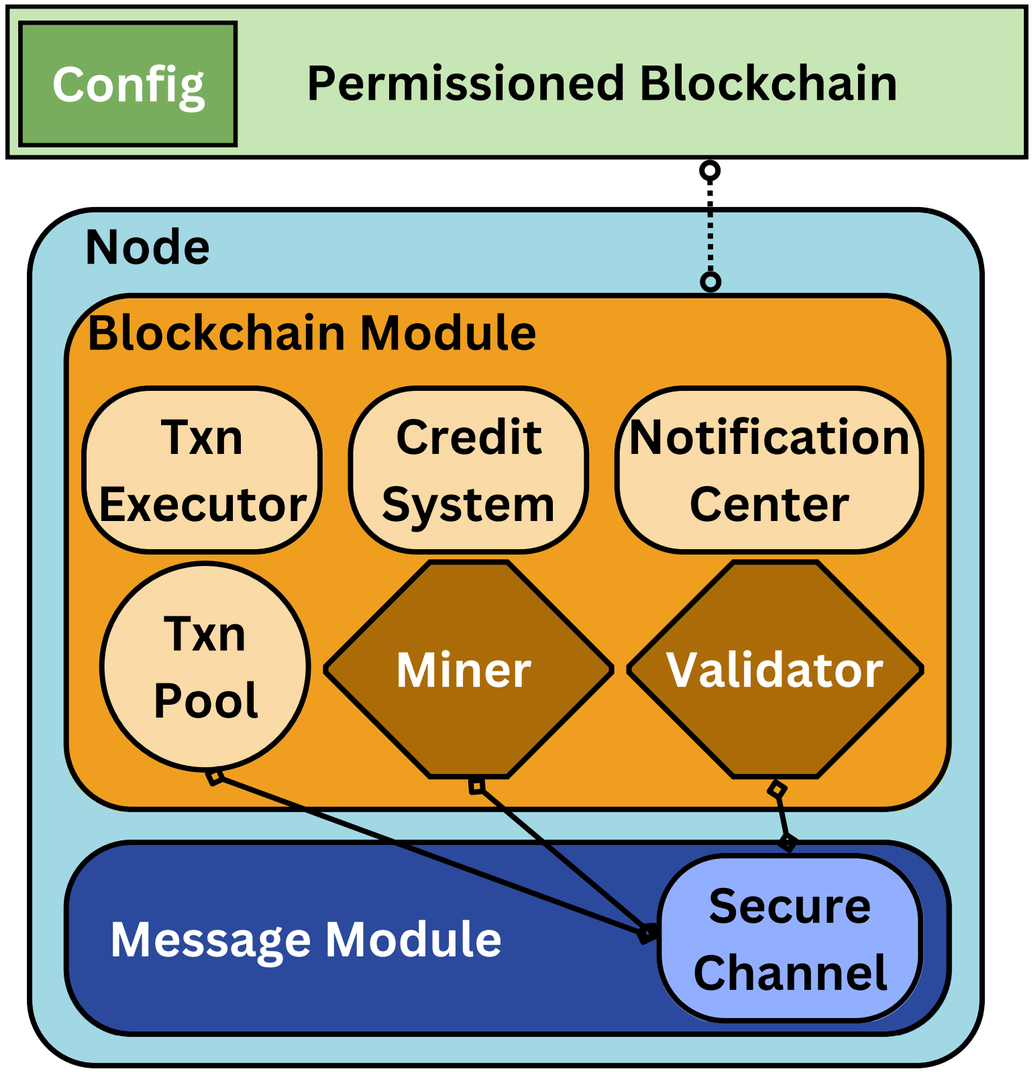
PERMISSIONED BLOCKCHAIN



- **Miner**: a daemon for mining and broadcasting new blocks
- **Validator**: a daemon for validating blocks and appending them to the blockchain
- **Txn Pool**: cache unprocessed txns between Message Module and miner
- **Txn Executor**: validate and execute txn. Only support dedicated txn types
- **Credit System**: select the next miner based on balances and coinage

Yuening Yang

PERMISSIONED BLOCKCHAIN

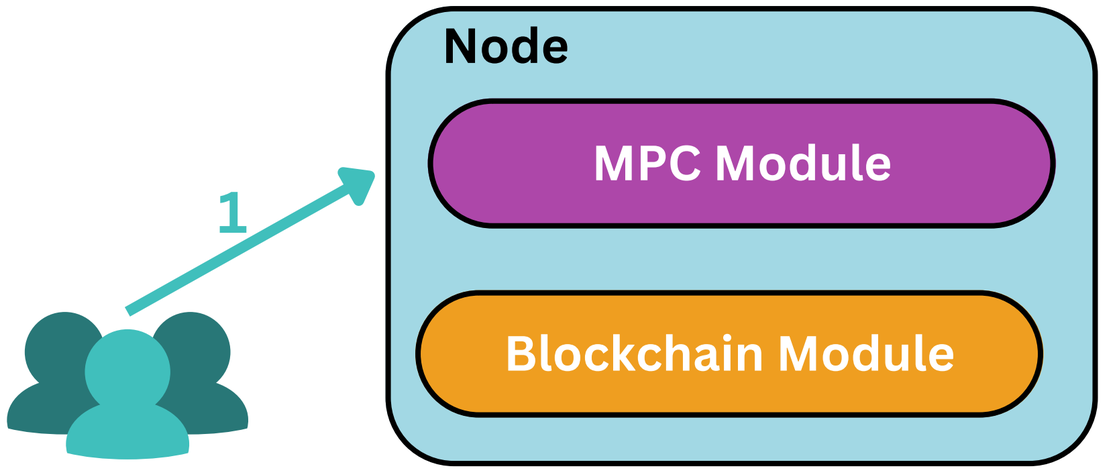


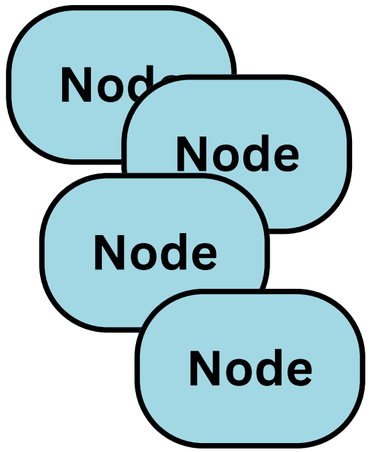
- **Miner**: a daemon for mining and broadcasting new blocks
- **Validator**: a daemon for validating blocks and appending them to the blockchain
- **Txn Pool**: cache unprocessed txns between Message Module and miner
- **Txn Executor**: validate and execute txn. Only support dedicated txn types
- **Credit System**: select the next miner based on balances and coinage
- **Notification Center**: notifies other modules of new transaction commits

Yuening Yang

INTEGRATION: MPC+BC

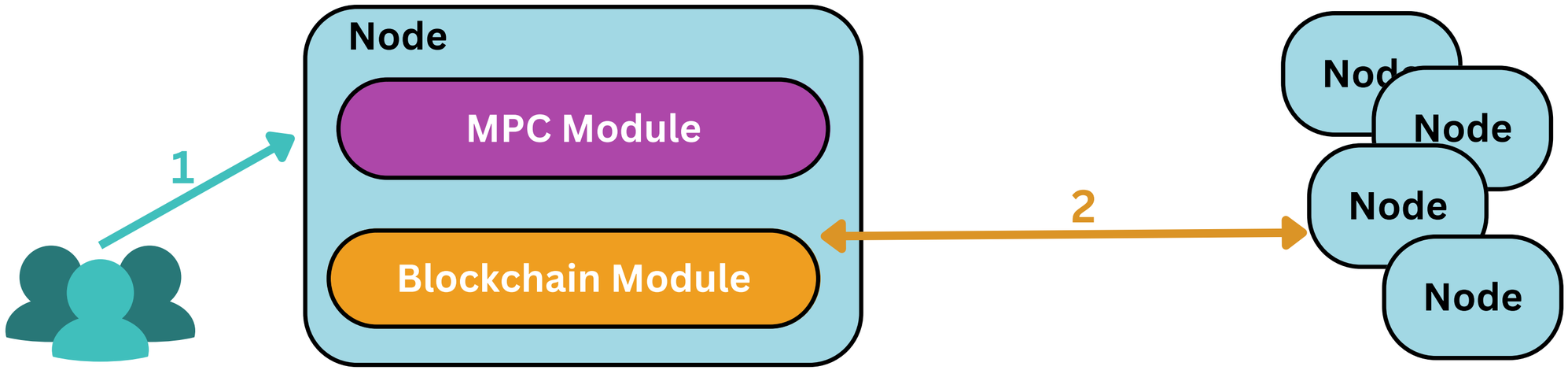
1. User sends MPC request with a budget to be paid





INTEGRATION: MPC+BC

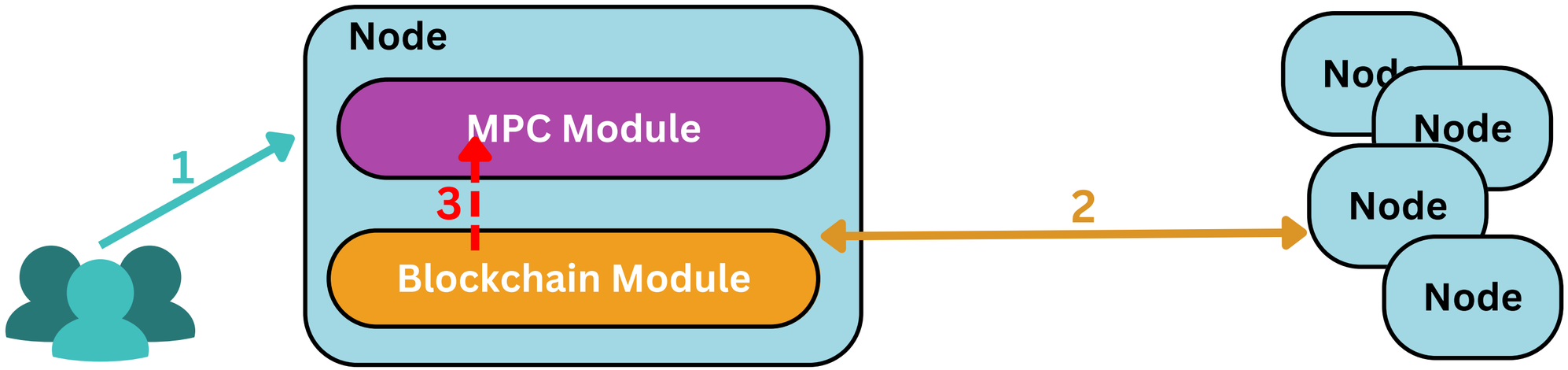
1. User sends MPC request with a budget to be paid
2. Blockchain Module includes the PreMPC txn into the ledger



Yuening Yang

INTEGRATION: MPC+BC

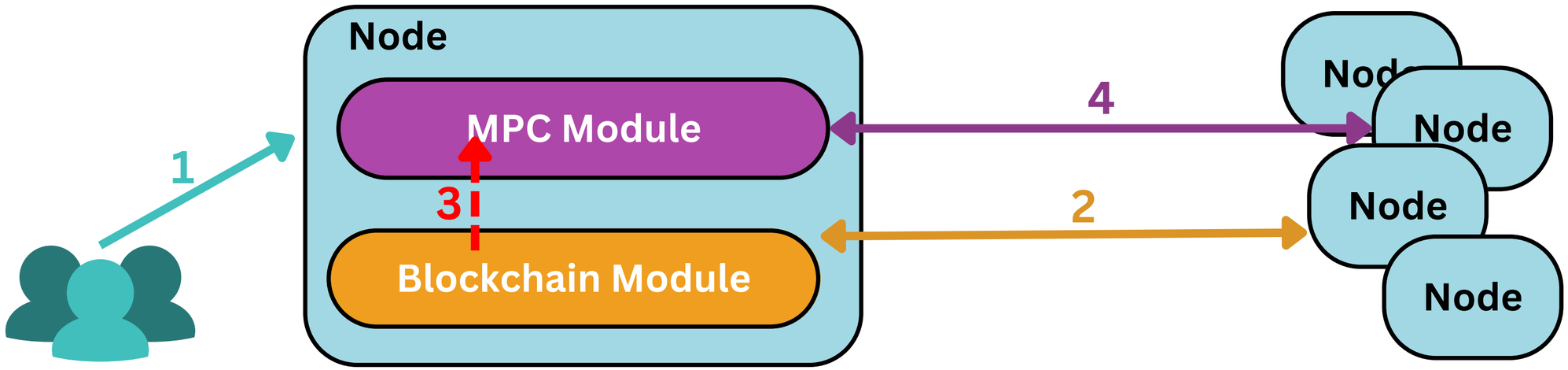
1. User sends MPC request with a budget to be paid
2. Blockchain Module includes the PreMPC txn into the ledger
3. Blockchain Module notifies MPC Module of PreMPC txn commit



Yuening Yang

INTEGRATION: MPC+BC

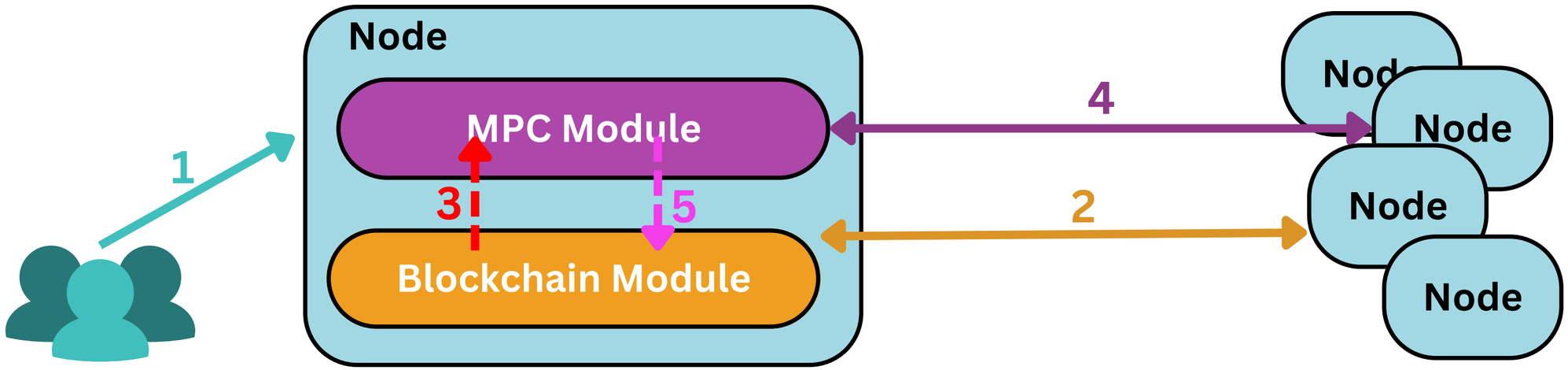
1. User sends MPC request with a budget to be paid
2. Blockchain Module includes the PreMPC txn into the ledger
3. Blockchain Module notifies MPC Module of PreMPC txn commit
4. MPC Module starts MPC



Yuening Yang

INTEGRATION: MPC+BC

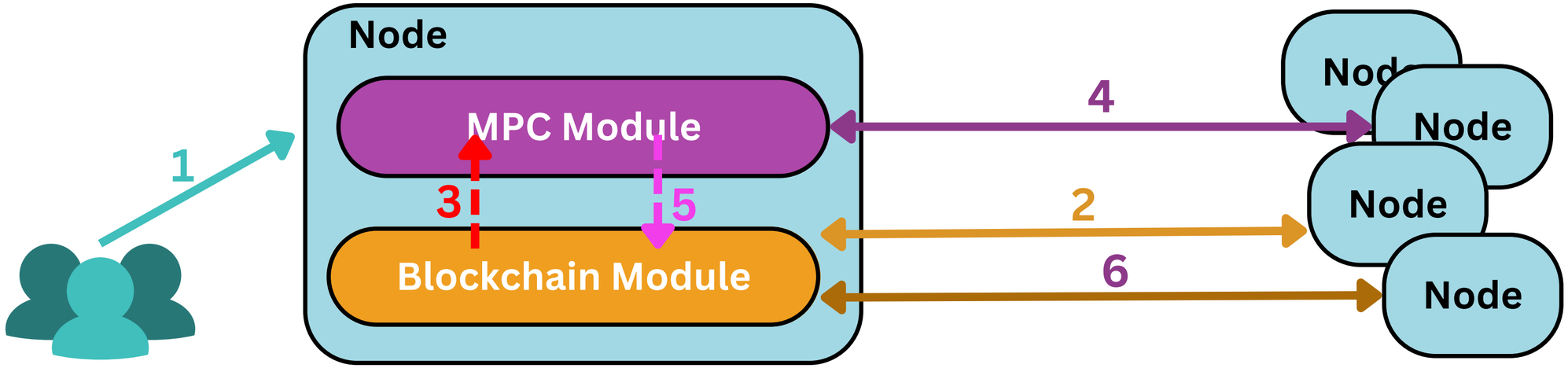
1. User sends MPC request with a budget to be paid
2. Blockchain Module includes the PreMPC txn into the ledger
3. Blockchain Module notifies MPC Module of PreMPC txn commit
4. MPC Module starts MPC
5. MPC Module notifies Blockchain Module who sends a PostMPC txn



Yuening Yang

INTEGRATION: MPC+BC

1. User sends MPC request with a budget to be paid
2. Blockchain Module includes the PreMPC txn into the ledger
3. Blockchain Module notifies MPC Module of PreMPC txn commit
4. MPC Module starts MPC
5. MPC Module notifies Blockchain Module who sends a PostMPC txn
6. Locked budget is released to participants (#PostMPC Txn > 1/2)



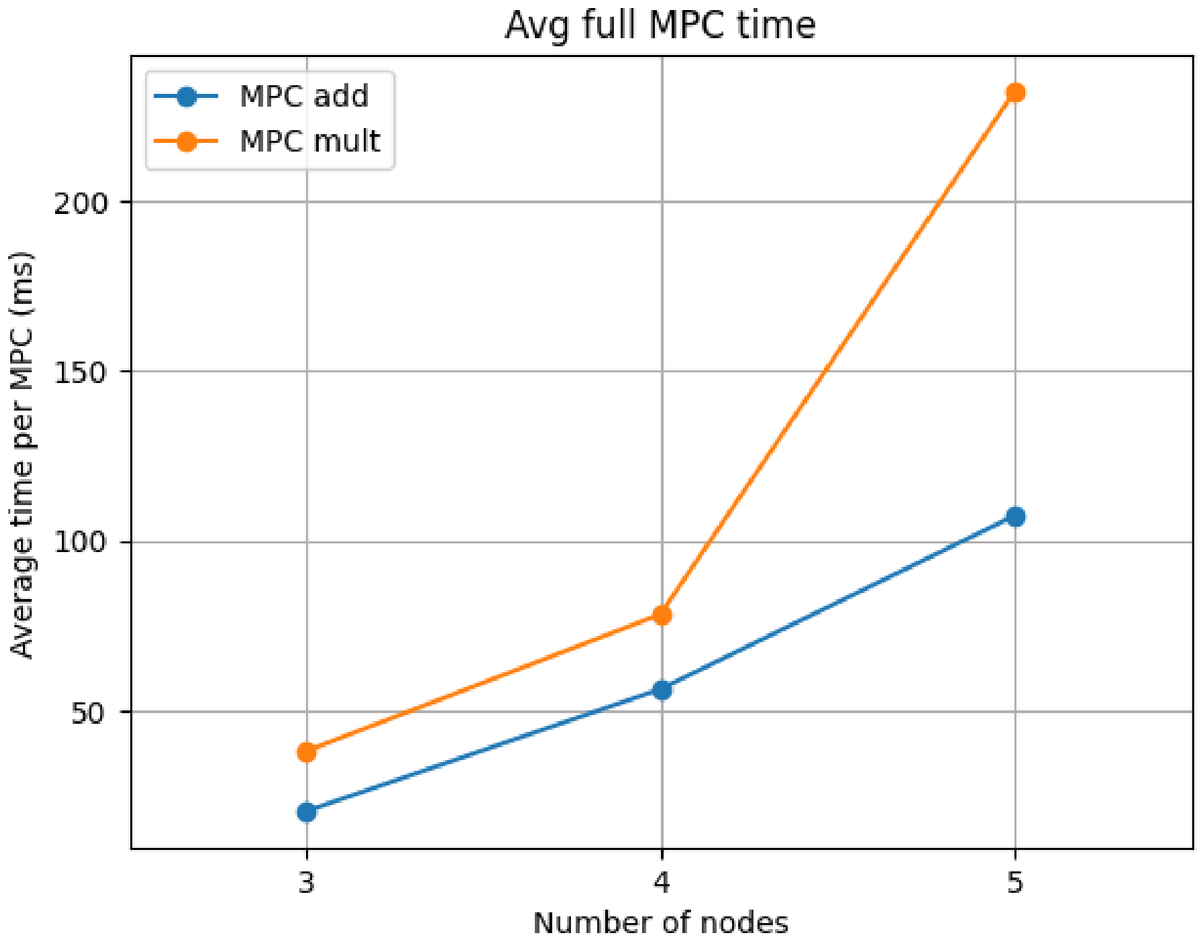
Yuening Yang

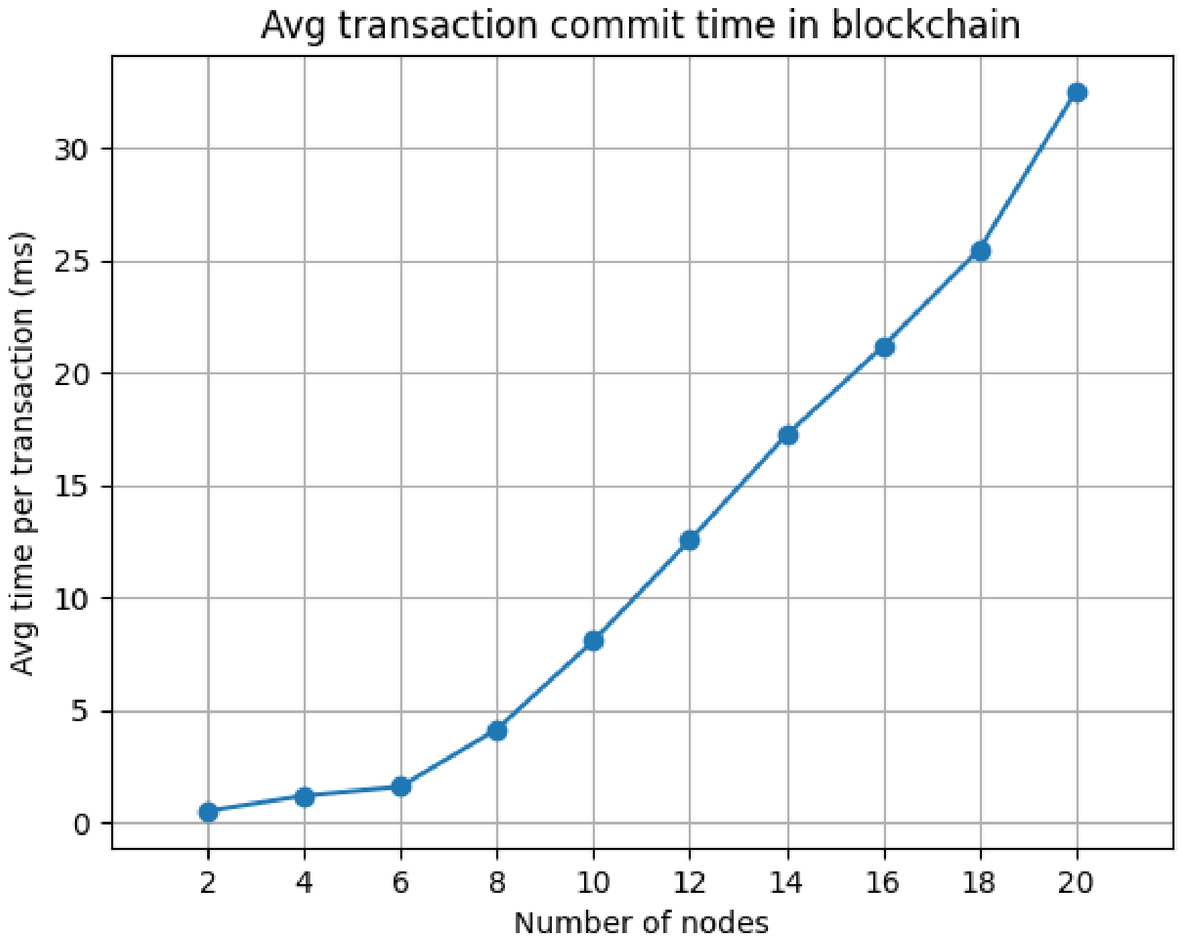
3

EVALUATION

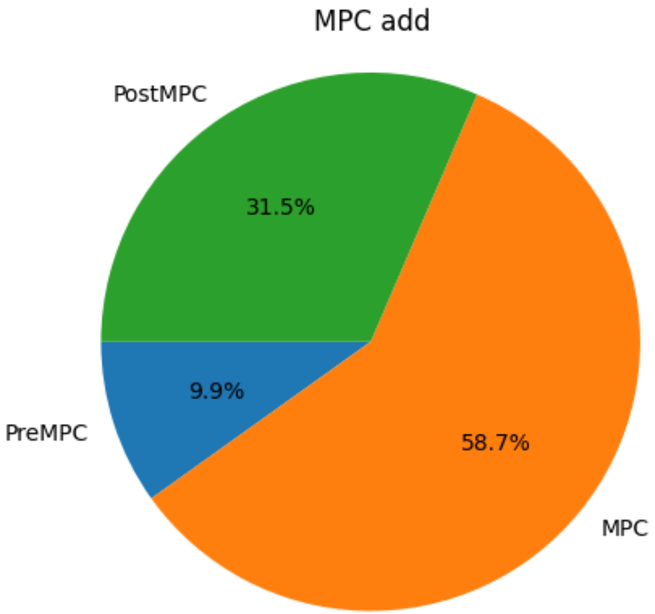
MPC PEERSTER

PERFORMANCE

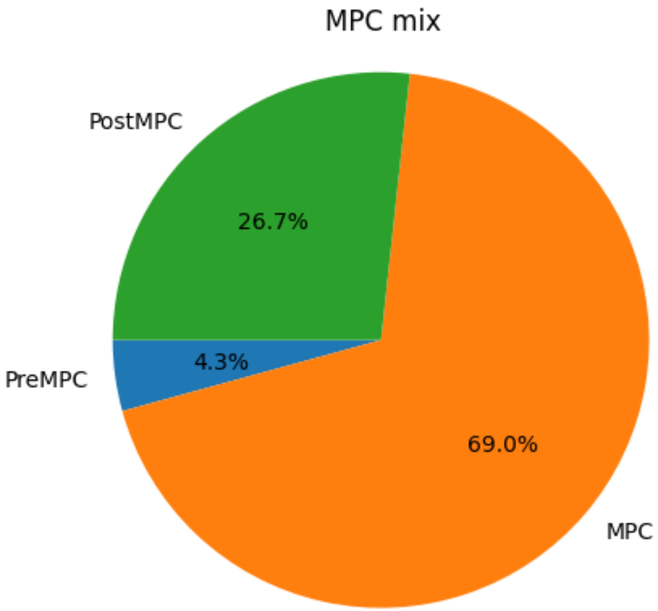




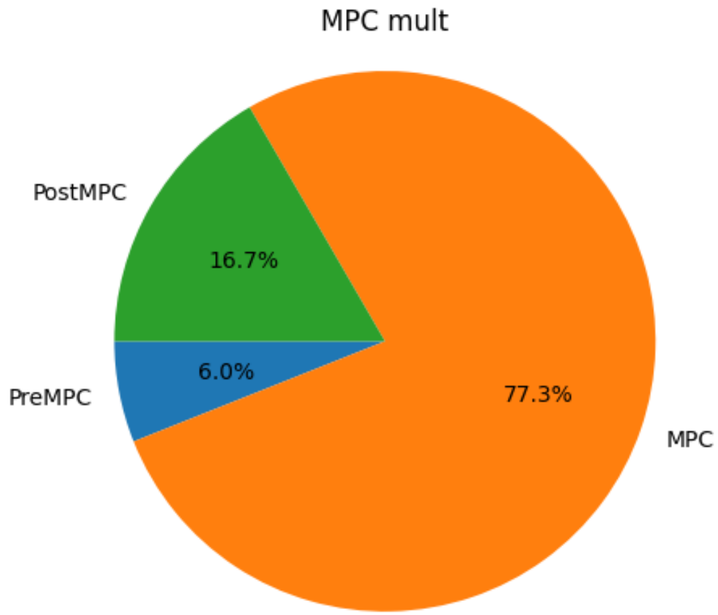
PERFORMANCE



a+b+c
(42ms)



(a+b)*c
(60ms)



a*b*c
(107ms)

SECURITY

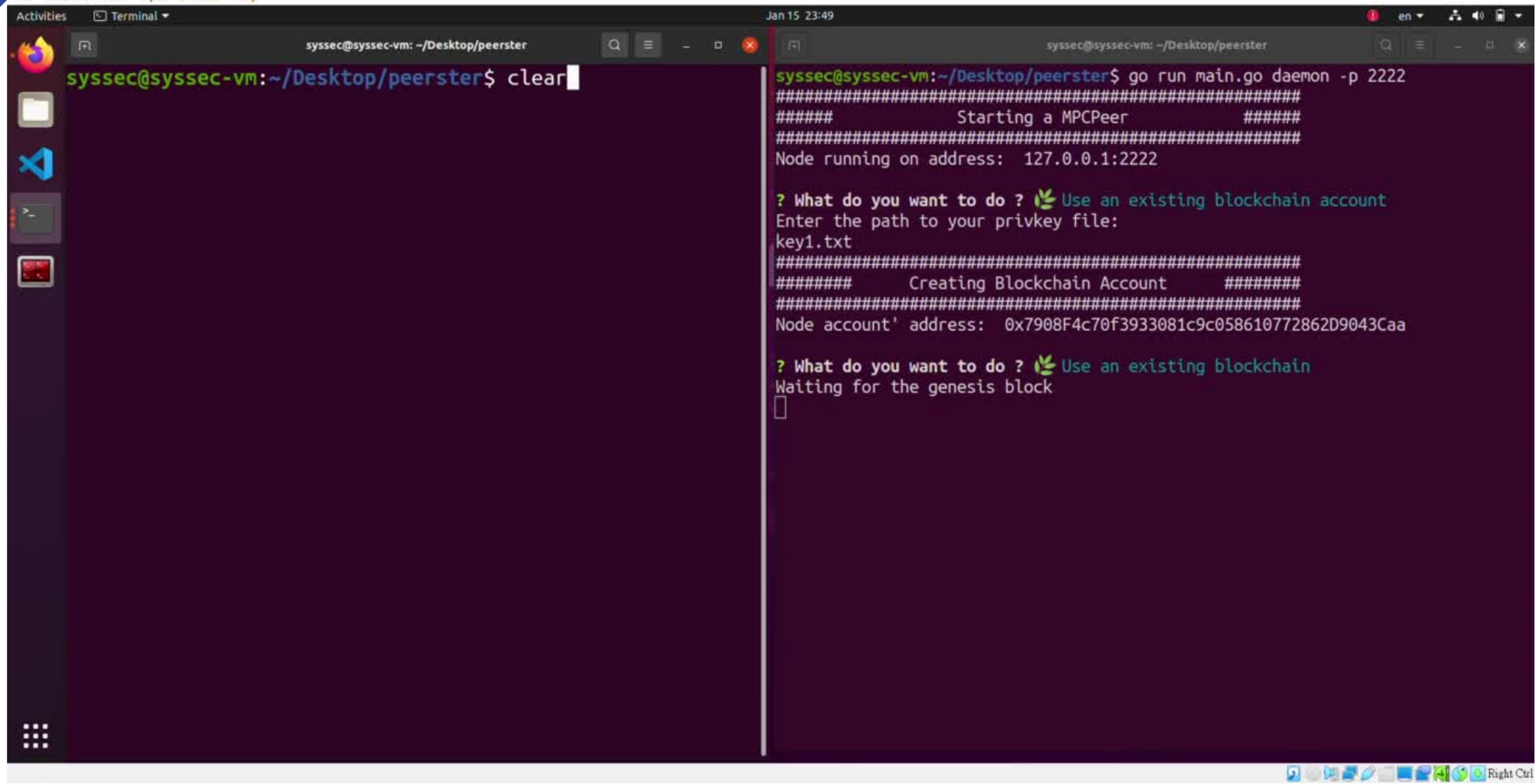
	MPC	Blockchain

Things Provided	• Secure channels
	• Synchronous communication
	• Passive Adversary
	• Information-theoretic level security & privacy against <1/2 corruptions
		• Access control & Authentication
		• Accountability
		• Correctness against Active Adversaries who can
		  ◦ produce fake transactions
		  ◦ replay authenticated transactions
		  ◦ produce faulty blocks

Future Improv-ments	• Asynchronous communicaiton • Privacy against Active Adversary (crytographic security: Commitment)	• Asynchronous consensus algorithm • Protection against DoS attack • Real blockchain privacy

DEMO

UI available at: https://chenlew.wixsite.com/mpcpeerster
***Must be connected to the localhost:7122 node**



THANK YOU !

Code available at: https://github.com/lilyyangyn/peerster

Code Contribution			
Chen Chang Lew	MPC Integration Flow, Benchmark UI	Total	
		+2833	-857
Junzhen Lou	MPC Module and Protocols, Finite Field Arithmetic, Benchmark	Total	
		+2126	-297
Yuening Yang	Secure channel, Blockchain, Integration MPC on Blockchain, Benchmark CLI tool	Total	
		+15010	-4774