EGES: Efficient and DoS-Resistant Consensus for Permissioned Blockchains

Xusheng Chen, Shixiong Zhao, Ji Qi, Cheng Wang, Haoze Song,
Jianyu Jiang, Tsz On Li, T.-H. Hubert, Fengwei Zhang,
Xiapu Luo, Sen Wang, Gong Zhang, Heming Cui

The University of Hong Kong

SUSTech

The Hong Kong Polytechnic University

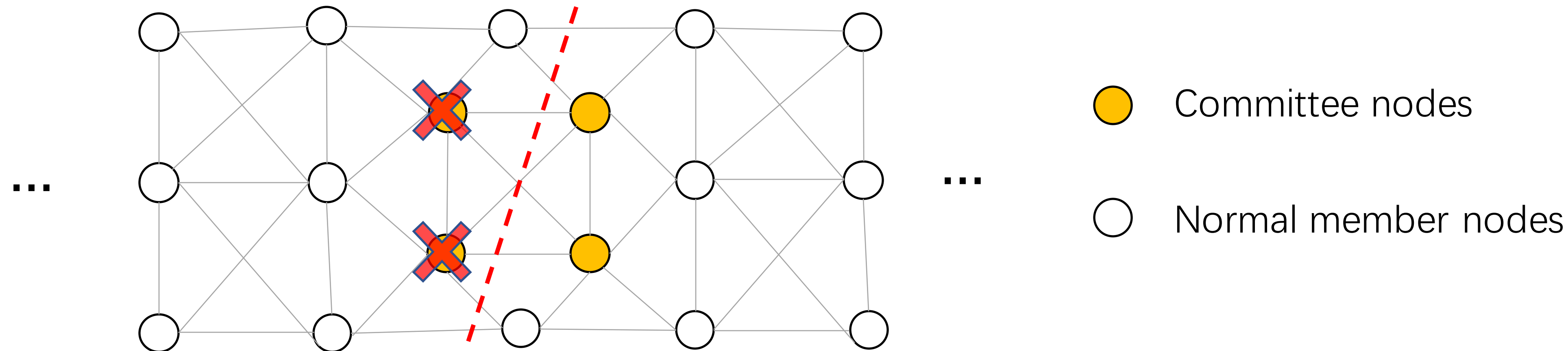
Huawei

Background: blockchain systems

- A distributed ledger of totally ordered transactions, forming a hash chain.
- Nodes run a **distributed consensus protocol** to agree on transactions.
- Two types of blockchains:
  - Permissionless (open membership): any node can join and leave at any time.
    - E.g., BitCoin, Ethereum, Algorand [SOSP '17]
    - Requires cryptocurrency to incentivize nodes to follow the protocol.
  - Permissioned: nodes should register before joining the system.
    - E.g., HyperLedger projects, Libra, Quorum.
    - Decoupled from cryptocurrency, can deploy diverse data-sharing applications.
    - Can explore existing consensus protocols for high performance.
- We focus on **permissioned blockchains** for their generality and high performance.

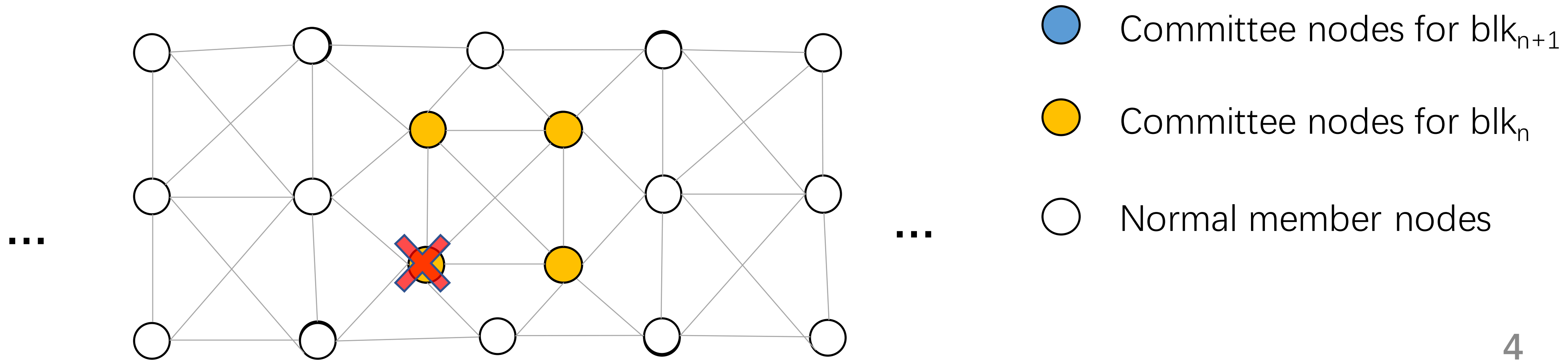
Problems of existing permissioned blockchains

- Susceptible for **targeted DoS and network partitioning attacks** if deployed on the Internet or among edge computing IoT devices.
  - We discuss these two attacks together and call them altogether **targeted DoS attacks.**
- Existing permissioned blockchians rely on a **static and explicit committee** to achieve consensus.
  - E.g., BFT-SMaRT [DSN '18], Tendermint [OPODIS '18], HotStuff [PODC '19], MinBFT [TOC '11], SBFT [DSN '19], Honey Badger BFT [CCS '16].
  - This committee is usually very small (e.g., 10 in BFT-SMaRt) for efficiency.
  - Attacking only a portion of these committee nodes turn down the whole system.
  - Scalable BFT protocols (e.g., SBFT) cannot ensure performance with a few nodes attacked.
- Algorand [SOSP '17] uses dynamic committee and is resistant to targeted DoS attacks, but it is designed for permissionless blockchains, not suitable for permissioned blockchains.



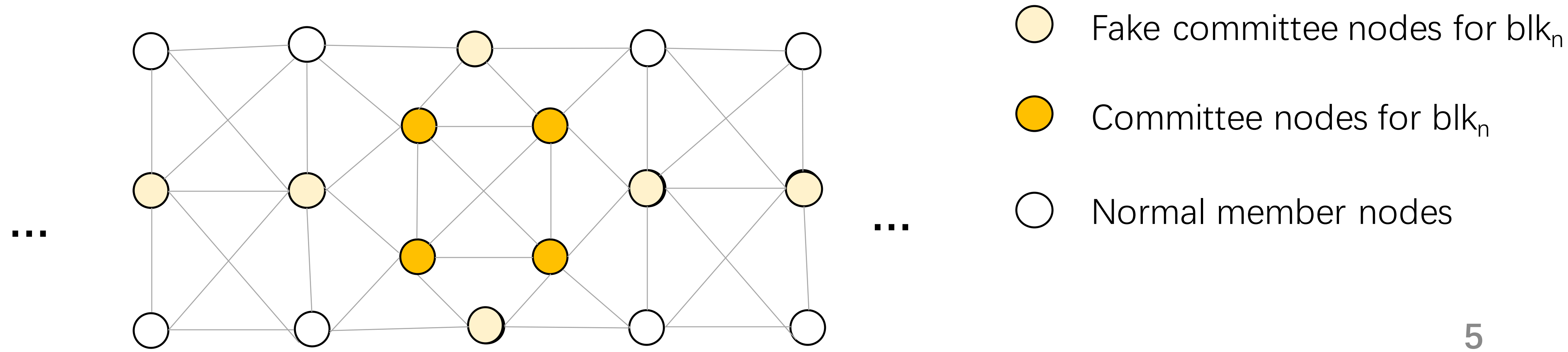
The high-level idea of EGES

- Existing permissioned blockchains are susceptible to targeted DoS attacks because they rely on a **static and explicit committee** to achieve consensus.
- We propose a new abstraction called **dynamic and stealth committee** for permissioned blockchain.
- **Dynamic**: Select a **random** group of committees for each block.
  - Move the attack target constantly to avoid being aimed at.
  - Concept borrowed from permissionless blockchains, but they use it for fairness.



The high-level idea of EGES

- Existing permissioned blockchains are susceptible to targeted DoS attacks because they rely on a **static and explicit committee** to achieve consensus.
- We propose a new abstraction called **dynamic and stealth committee** for permissioned blockchain.
- **Dynamic**: Select a random group of committees for each block.
- **Stealth**: Hide the committee's membership before and during consensus.
    - Use fake committee nodes to cover the new ones.

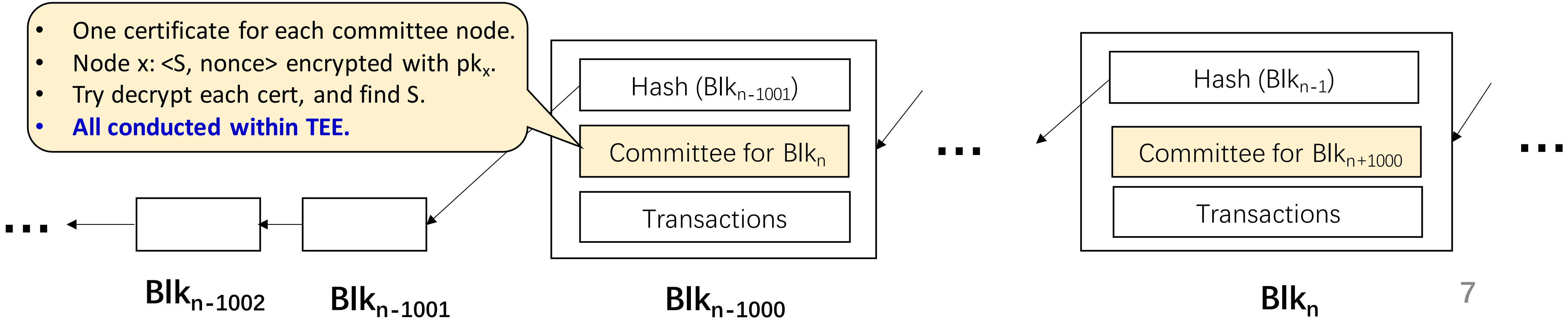


EGES leverages TEEs to realize the new abstraction

- Trusted Executing Environments (TEE) is a hardware feature of modern CPUs.
- TEEs are widely used in blockchain systems
  - REM [Usenix Security '17], PoET [Intel], Ekiden [Euro S&P '18], Hybster [EuroSys '17].
  - But none of them tackles the DoS-resistant problem.
- TEE provides an isolated trusted space called the **enclave.**
- **Integrity**: code running in an enclave cannot be tampered with (can be killed) from outside (even by the kernel/administrator)
  - EGES leverages this feature to govern the behavior of randomly selected nodes.
  - Otherwise, it's almost impossible to ensure any committees meet BFT's honesty requirement.
- **Confidentially**: data within enclave is not visible to outside world.
  - EGES leverages this feature to make the committee's members identities stealth.

EGES's detailed protocol – committee formation

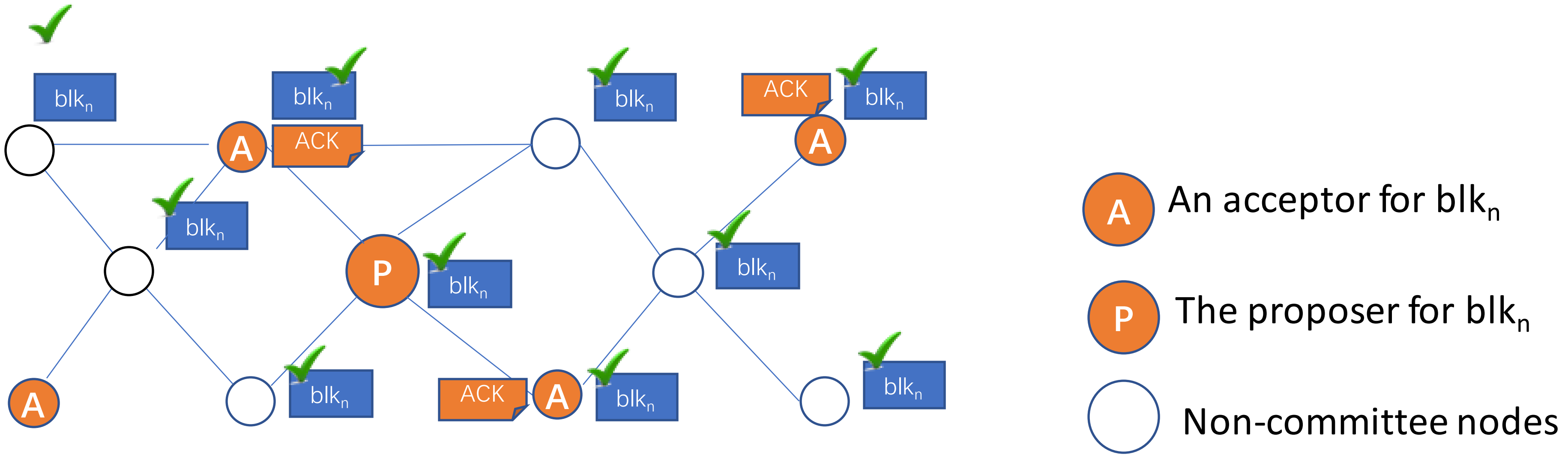
- EGES has one stealth committee for each block number.
  - Randomly and uniformly selected from all member nodes in the system.
  - One **proposer**: generates the unique block proposal and tries to confirm the block.
  - A group of **acceptors**: send ACKs on receiving the proposal.
  - **Fake acceptors** and **arbiters**: help to cover the real committee nodes.
- Committee nodes' identities are encrypted on the blockchain k (e.g., 1000) blocks ago.
  - Only a committee member node's enclave knows whether it is a committee member for $block_n$.



EGES's detailed protocol – confirming a block

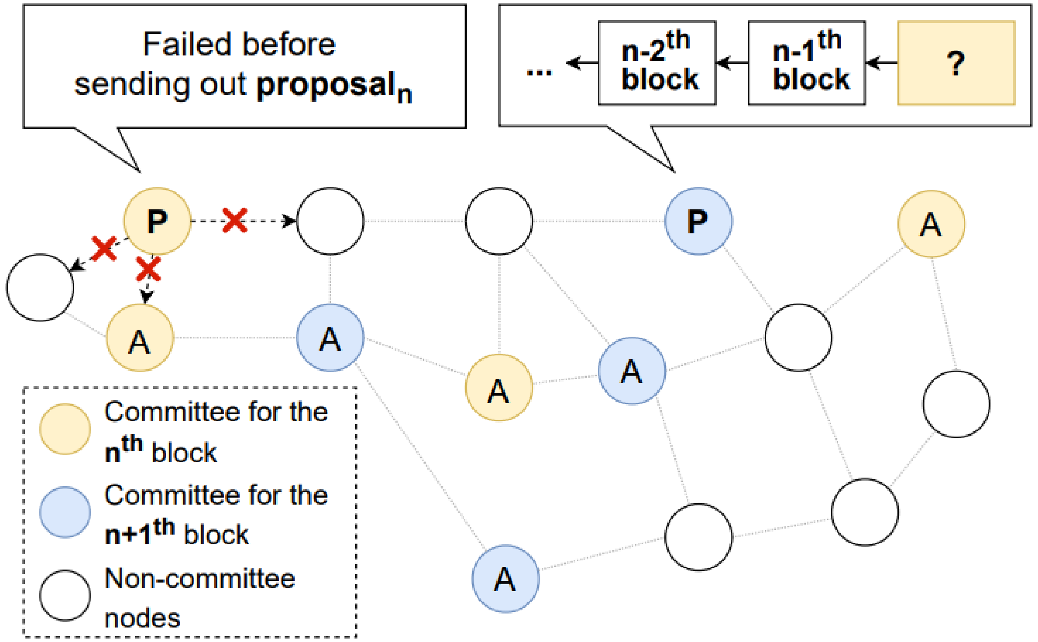
- Normal case (the proposer can reach most acceptors)
  - We omit fake acceptors in this talk (please read our paper).

1. The proposer broadcasts its unique block proposal via the P2P network.
2. An acceptor replies an ACK after receiving the proposal.
3. The proposer broadcasts a confirm message after receiving quorum ACKs.

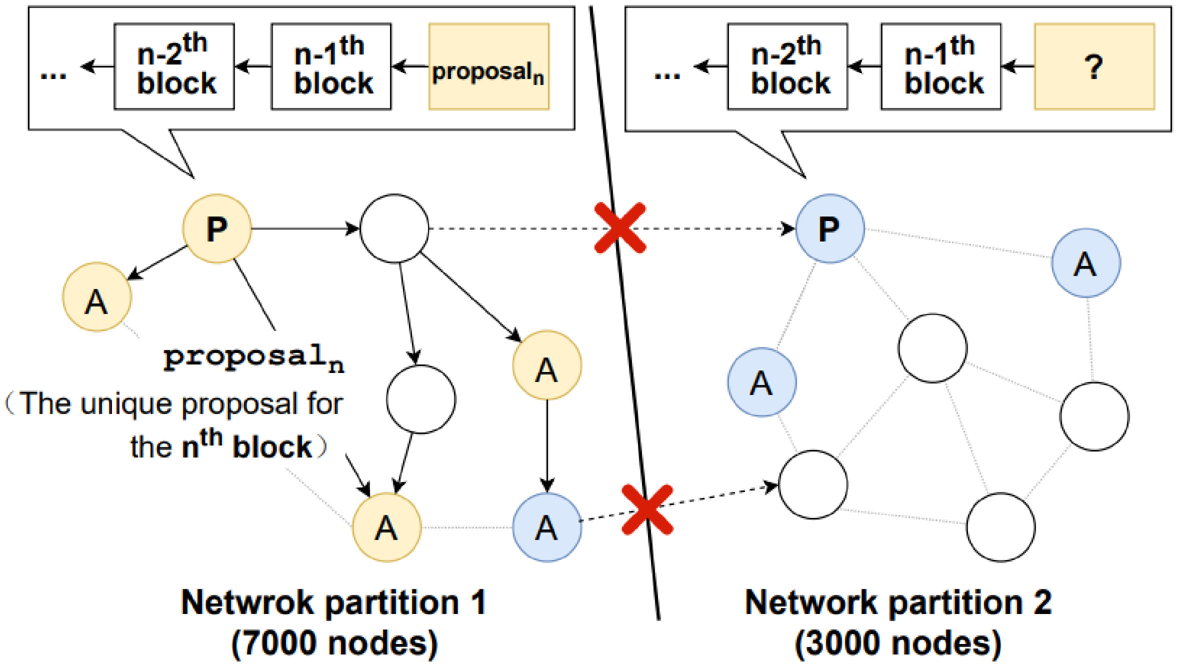


Key challenge: ensuring consistency on failure cases

- (Failure case) what if a node cannot receive the next block (say $block_n$) in time?
- Any single node cannot distinguish two cases:
  - **Case A**: The consensus on $block_n$ is never achieved.
  - **Case B:** The consensus $block_n$ is achieved, but this node does not receive the notification.



Case A: the $block_n$ is never proposed.



Case B: the $block_n$ is confirmed

EGES's new protocol based on probability theory

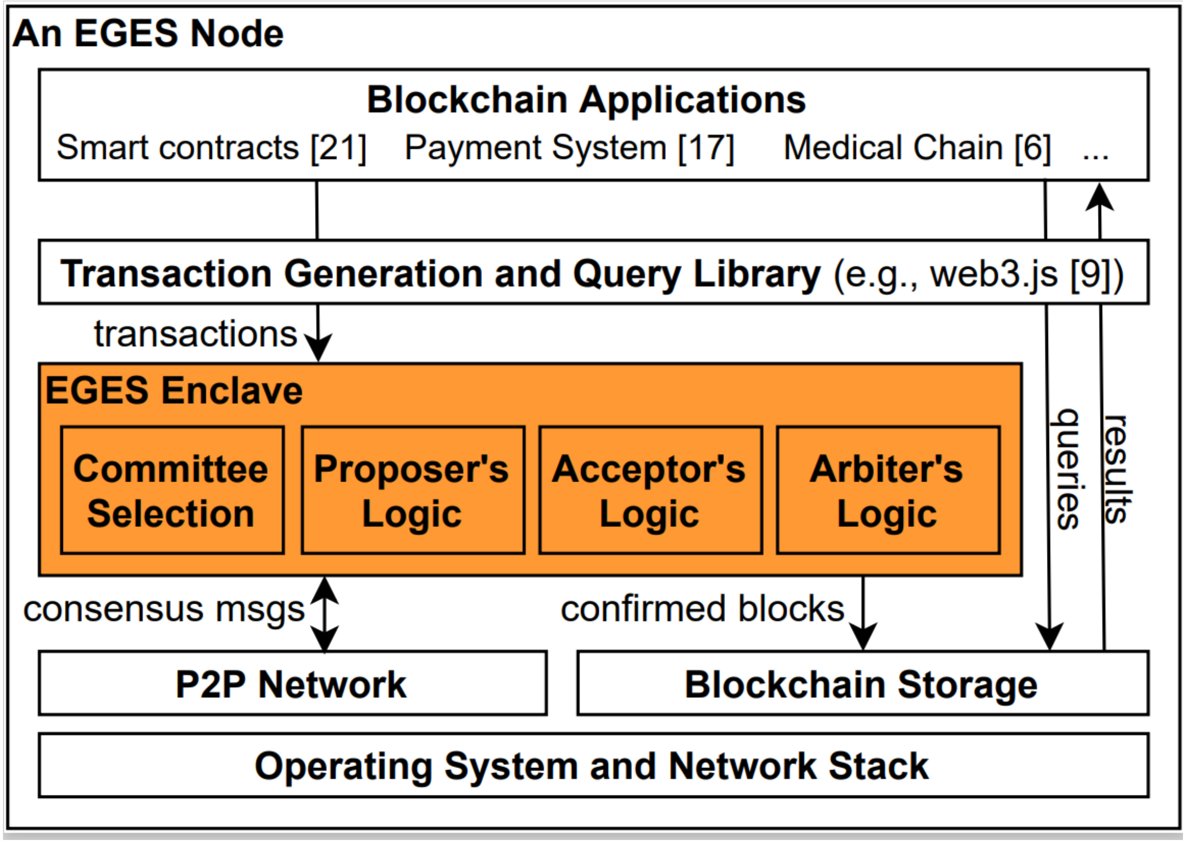
- Each block can only have two choices: a unique proposal or a default empty block.
- Model the **randomly selected acceptors** as a random **sampling of the dissemination** of the block proposal in the P2P network.
- Assume **P%** of a total of **M** nodes received the unique proposal, quorum ration is **q**.
  - **X:** the number of a total of n_A acceptors that have received the proposal.
  - X follows hypergeometric distribution: $X \sim H(M, n_A, P\% * M)$.
- Confirming the unique proposal: $Prob(X > q * n_A)$. Prob non-trivial requires **P% to be large enough.**
- Confirming empty block: ask another group of nodes whether received the proposal, repeat **D** times.
  - **Y:** the number of acceptors not received the proposal : $Y \sim H(M, n_A, (1-P\%) * M)$.
  - Probability $(Prob(Y > q * n_A))^n$ being non-trivial requires **P% to be small enough.**
- For instance, when q = 59%, n_A = 100, D = 4, M = 10k, EGES ensures the probability of inconsistency < 10^{-9}.

Summary of EGES

- Resistance to DoS or partition attacks targeting committee nodes.
  - **Before** a committee is on duty: use TEE to designate them in a random and unpredictable way.
  - **During** a committee is on duty: use fake committee nodes to cover their identities.
  - **After** a committee is on duty: rotate to a different group of committee to avoid the attacker target this committee.
- Efficiency:
  - The normal case is very lightweight.
  - The failure case's repetitive sampling runs in parallel with subsequent blocks' consensus.
- Please refer to our paper for detailed analysis and proof.

Implementation

- Based on the codebase of Ethereum.
  - Reuse its P2P layer;
  - Rewrite its membership and consensus layer.
- More details in the paper:
  - Supporting dynamic membership
  - Enclave interactions.
  - Handling forking attacks.
  - Handling timeout attacks.



Evaluation

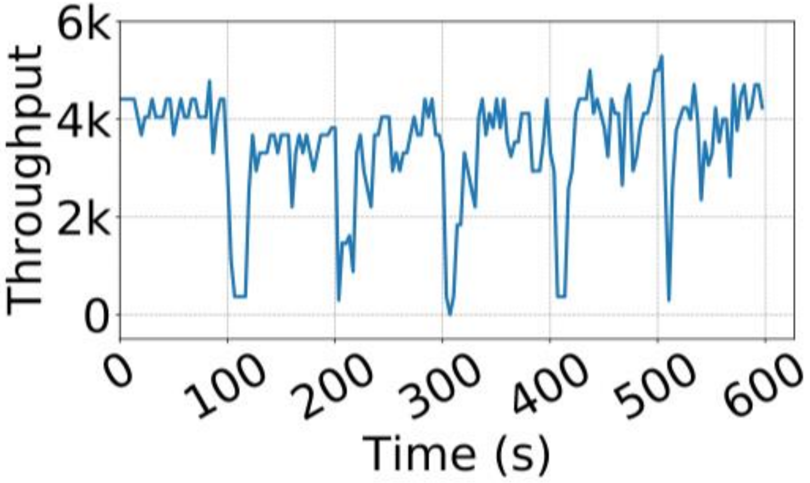
- Evaluation settings:
  - Cluster: 30 machines, Intel E3-1280 V6 CPU with SGX support.
  - AWS: up to 100 c5.18xlarge VMs 72 cores with SGX simulation mode.
  - Each node runs in a docker container, up to 10,000 nodes.
  - Network RTT between each pair of nodes: 150 ~ 300ms.
- Evaluation questions:
  - Is EGES efficient?
  - What is EGES's performance under DoS attacks?
  - Is EGES scalable?
  - How sensitive is EGES to its parameters?
  - How do EGES performance and fault tolerance compare with notable BFT protocols?

Throughput and latency

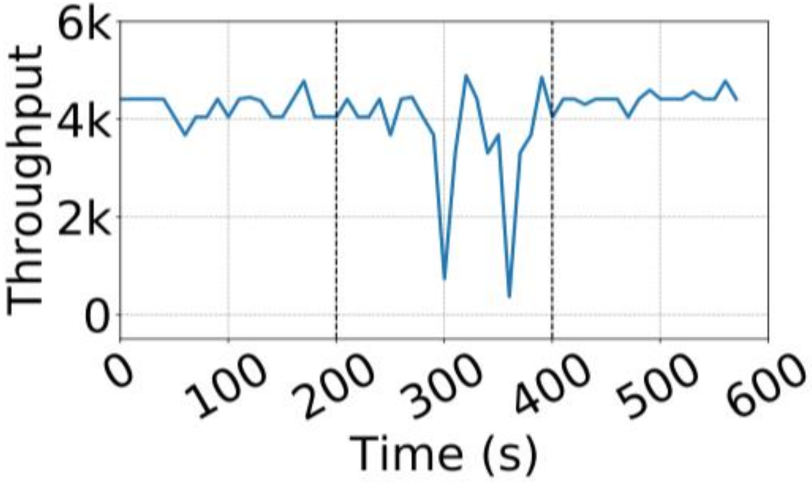
Protocol	DoS and partition resistance	With SGX?	Number of nodes	Tput (txn/s)	Confirm latency (s)
EGES	high	Yes	300	3226	0.91
			10K	2654	1.13
Algorand	high	No	10K	~727	~22
PoET	medium*	Yes	100	149	45.2
Ethereum	medium*	No	100	178	82.3
SBFT	low	No	62	1523	1.13
MinBFT	low	Yes	64	2478	0.80
BFT-SMaRt	low	No	10	4512	0.67
Tendermint	low	No	64	2462	1.31
HotStuff	low	No	64	2686	2.63
HoneyBadger	low	No	32	1078	9.39

- More efficient than existing DoS-resistant protocols for permissionless blockchains.
- Comparable efficiency to existing blockchain protocols that are not DoS resistant.

EGES's performance under targeted DoS attacks



(a) Targted DoS attacks



(b) 80%-20% partition

- Attacked 10% of all nodes every 100s.
- Targeting the real/fake committee nodes

- Created a partition at 200s
- Network reconnected at 400s.
- Throughput measured at the large partition.

Conclusion

- EGES is the first consensus protocol for permissioned blockchain that:
  - Resistant to DoS and network partition attacks targeting committee nodes.
  - Achieves comparable performance as protocols running on static committees.
- We propose:
  - A new dynamic stealth committee abstraction to protect committee nodes.
  - A new consensus protocol to ensure safety on dynamic committee.
- EGES is suitable to large scale permissioned blockchains running on the Internet and IoT devices (e.g., edge computing).
- EGES is open source on **github.com/hku-systems/eges.**