Asian Deans' Forum 2026

THE RISING STARS

Women in Engineering Workshop

SEP 13 – 15 | HKUST, Hong Kong

# Xiao Sui


Postdoctoral Researcher, Tsinghua University


BFT Consensus · Blockchain Systems · Distributed Cryptography · AI-assisted Protocol Analysis

Academic Profile

# Secure and Efficient Distributed Computing

## From Byzantine Consensus to Real-World Systems

**I design consensus protocols that keep distributed systems trustworthy and efficient, even when nodes fail, networks fluctuate, or participants behave maliciously.**

**Research at a Glance**

A roadmap that connects protocol design, deployment lessons, and standardization.

**Consensus Protocols**
BFT · SMR · Atomic Broadcast

**Distributed Cryptography**
MPC · Threshold Crypto · Distributed Randomness

**Real-World Systems**
Blockchain · Deployment · Standardization

**Selected Research**

| SQ | Rondo | Dashing | BG | Pike | Marlin |
|---|---|---|---|---|---|
| S&P 2025 | NDSS 2025 | EuroSys 2024 | TIFS 2023 | TC 2025 | DSN 2022 |

Across partial synchrony, asynchrony, and distributed randomness

**Research → Systems**

1. **Protocol Design** — Provably secure & efficient BFT
2. **Real Deployment** — mBridge experience
3. **Standardization** — From guarantees to deployable requirements

**Selected Recognition**

- Financial Technology Development Award — First Prize, People's Bank of China
- Tsinghua Shuimu Scholar
- CCF Dissertation Recognition
- Special Funding, China Postdoctoral Science Foundation

**Academic Service · ACM CCS 2025–2026 Program Committee**

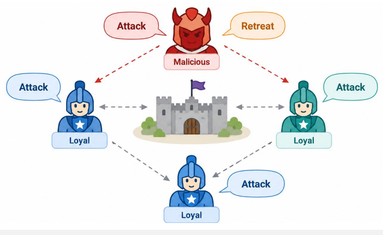
*Current focus: efficient Byzantine consensus as a foundation for broader secure distributed computing.*

## 1 Consensus: why it matters

*One primitive, many distributed systems*

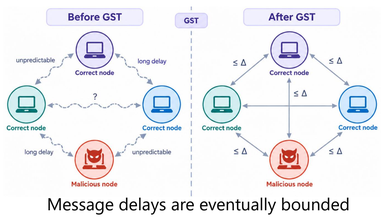
**Byzantine Generals**

*Lamport et al. (1982) introduced the Byzantine Generals Problem to model agreement under accidental or malicious faults.*



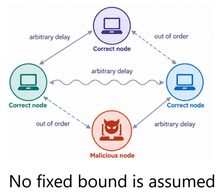

All loyal generals decide upon **the same plan of action**
A small number of traitors **cannot cause the loyal generals to adopt a bad plan**.

**Partially synchronous**




Message delays are eventually bounded

**Asynchronous**




No fixed bound is assumed

Partial synchrony and asynchrony better reflect real-world distributed systems.

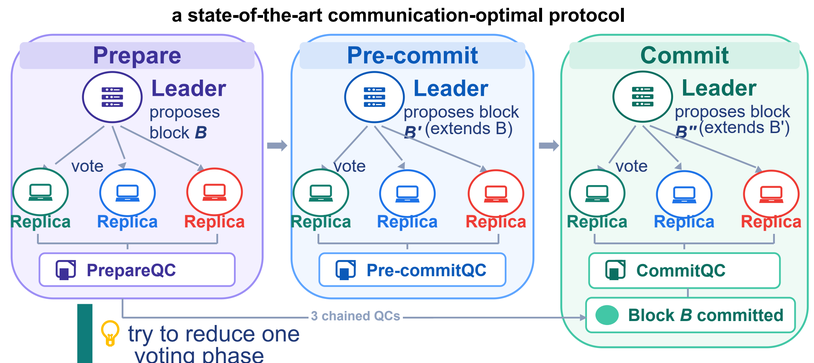
**Byzantine Consensus** — Agree on a common state/ order despite Byzantine failures → **Replicated Services** — State Machine Replication; **Blockchain Systems** — Transaction Ordering & Finality; **Distributed Cryptography** — MPC · DKG · Randomness

**Takeaway: BFT is the core technology of fault-tolerant distributed computing, and BFT consensus sits on the critical path of the entire distributed system.**

## 2 Efficient BFT under partial synchrony

*Can we improve performance without hidden costs?*

**a state-of-the-art communication-optimal protocol**




Marlin / Pike DSN'22 · TC'25

**optimal** two-phase linear BFT with practical view-change trade-offs
Inspired follow-up work on two-phase BFT in leading security and distributed-systems venues

weak certificates exploit heterogeneous networks and improve **robustness**
First to use weak certificates with f + 1 votes, reaching the **information-theoretic minimum**

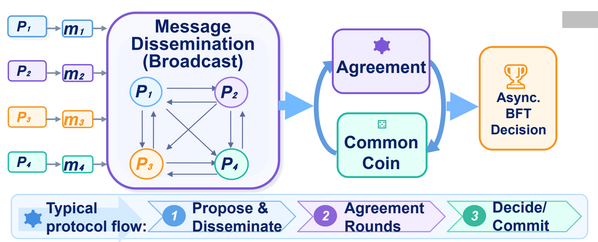
Dashing / Star EuroSys'24

**Takeaway: efficiency is multi-dimensional; no single mechanism optimizes every metric.**

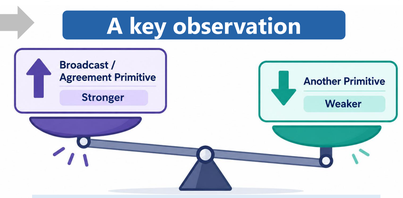
## 3 Efficient BFT under asynchrony

*Can we break through the bottleneck of theoretical complexity?*

Most asynchronous BFT protocols are built on broadcast primitives (e.g., RBC) and agreement primitives (e.g., MVBA, RABA, ABA).



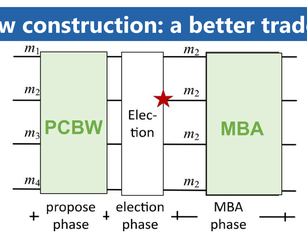

**A key observation**




When one primitive is strong, **the other can be weakened**

**A new construction: a better trade-off**

Agreement primitive: ABA/RABA → MBA → MVBA (stronger, cryptographically heavier)
Broadcast primitive: CBC → ? → RBC (stronger, message heavier)




| optimal | optimal |
|---|---|
| $O(n^2)$ messages | $O(1)$ expected time |
| none extra signatures | signature-free security flavor |

**post-quantum security**

**Takeaway: sometimes the right abstraction makes an old primitive useful again.**

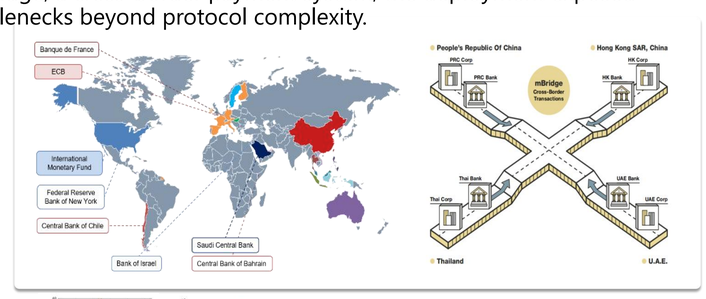
## 4 From protocols to real systems—mBridge

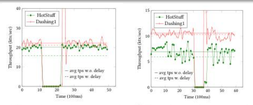
*A deployment lens from financial infrastructure*

Dashing1 was deployed in mBridge, a cross-border payment system; the deployment exposed bottlenecks beyond protocol complexity.

**Consensus design**
**Network**
**Execution**
**Storage & state**
**Operations**


The project is a collaborative effort of the *BIS Innovation Hub, four founding central banks and over 25 observing members*



(c) Throughput under one-second unexpected delay for $f = 1$. (d) Throughput under one-second unexpected delay for $f = 10$.

| bandwidth | HotStuff | Dashing1 | improvement |
|---|---|---|---|
| 10 Mbps | 575 | 942 | 63.8% |
| 20 Mbps | 1,134 | 1,616 | 42.5% |
| 50 Mbps | 1,668 | 2,483 | 48.8% |
| 200 Mbps | 1,896 | 3,034 | 60.02% |

peak throughput with four sites for Dashing1 and HotStuff

- Dashing1 achieves both higher efficiency and robustness
- Star achieves greater scalability under high transaction concurrency

Project mBridge: Connecting economies through CBDC, 2022
Project mBridge: experimenting with a multi-CBDC platform for cross-border payments, 2023

## 5 From guarantees to standards

*Turning proofs and deployment lessons into reusable requirements*

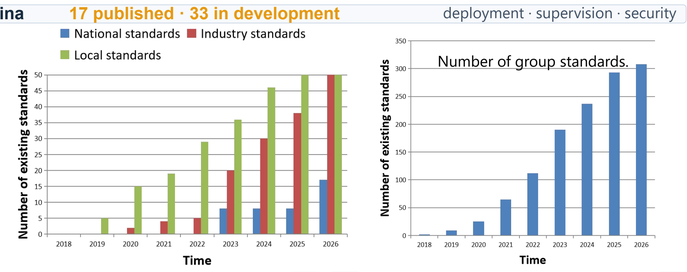
**General path**
Protocol → requirement

1. Terminology & architecture
2. System-level security
3. Consensus requirements
4. Safety · liveness · finality

Assumptions + properties + verification

**Standardization landscape (from our survey)**

| | | |
|---|---|---|
| ISO/TC 307 | 14 published · 25 in development | interoperability · identity · tokenization/Web3 |
| IEEE SA | 25 published · 71 in development | consensus framework · testing |
| ITU-T | 49 published · 39 in development | applications · security · performance evaluation |
| China | 17 published · 33 in development | deployment · supervision · security |




Coverage: 14 / 20 economic sectors (selected); reviewed group-standard records (n=309).

**Theory** — properties → criteria
- Assumptions
- Properties
- Tests

testable requirements

**Protocol** — algorithm → specification
- Roles, states, logs, message formats.
- Certificates, thresholds, validation rules.

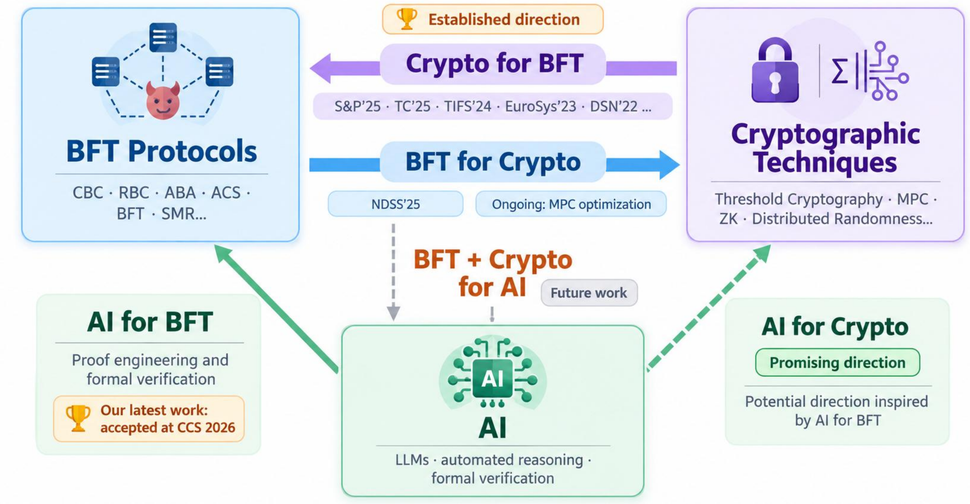
reusable specification

**Deployment** — practice → guidance
- WAN instability, slow links, recovery paths.
- Version consistency, timer/clock gaps,

deployment guidance

## 6 Next questions

*BFT protocols, cryptography, and AI form a two-way research agenda*




**Secure and Efficient Distributed Computing**

Organizers

The Hong Kong University of Science and Technology · UNSW Sydney · Seoul National University · National University of Singapore · NTU · UTokyo · Tsinghua University