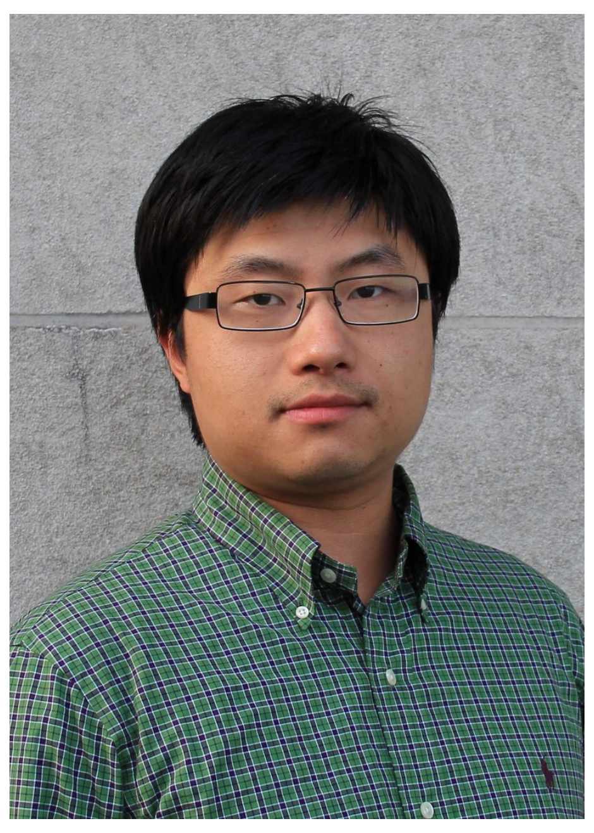
Blockchain-based Fork Prevention with Low-end Clients



Yuzhe Tang, Syracuse University

Introduction

Cloud storage services

- Serve data reads/writes

Storage consistency

- Does a data read return the latest write?

Host a public-key directory on Amazon S3.

Is Amazon trustworthy?

- Constant security incidents & data breaches
- Buggy software that can be exploited

Inconsistency means insecurity:

Proposal: Blockchain-enforced consistency check

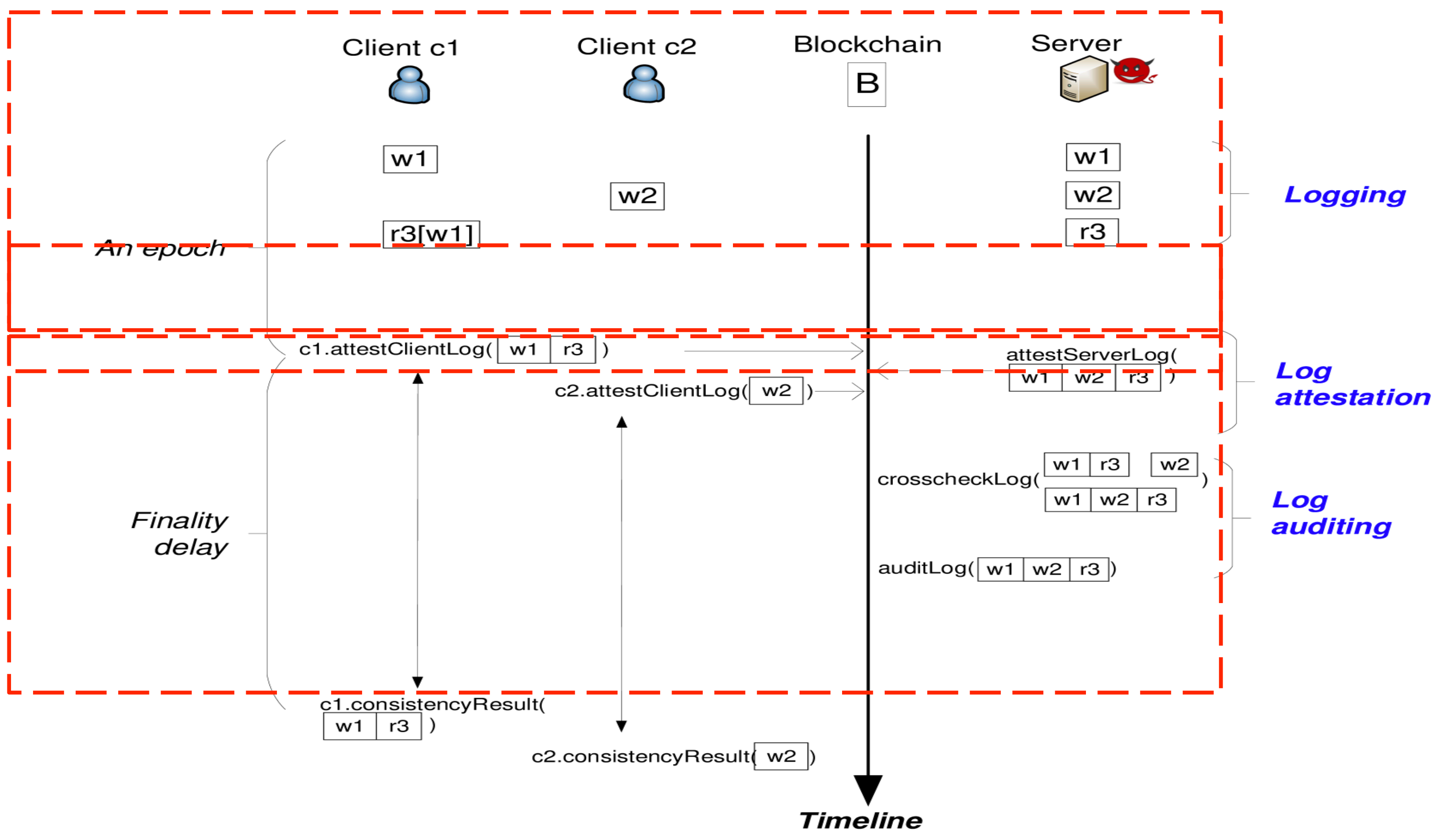
- Propose to use Blockchain for detecting and mitigating inconsistency of untrusted cloud service.

Ideally, one can use Blockchain as a TTP to:

- Collect the traces from clients and server,
- And cross-check the traces to detect the inconsistency.

We propose, ContractChecker, a security protocol running between Blockchain and clients/server

Security Protocol (ContractChecker)



Research Problem: Blockchain Systems are Exploitable?

- ContractChecker is secure under the (unrealistic) assumption that Blockchain is trusted and secure.
- But in the real world, Blockchain systems are exploitable.
- Thus, can we attack ContractChecker exploiting real Blockchain systems' "vulnerabilities"?

ContractChecker Attacks

ContractChecker attacks exploiting Blockchain system vulnerabilities

1. **Exploiting write availability**
2. **Exploiting Blockchain forks**
3. Exploiting smart contract races

Attack 1

Attacker's goal is to hide any inconsistency by malicious server.

- Client C1 sends w1 at time t1
- Client C2 sends w2 at time t2
- Client C1 sends r3 and receives w1 at time t3 (that is, r3[w1])
- An attack succeeds if the server can convince C1 that r3[w1] is consistent.

Idea: Exploiting Blockchain write unavailability to omit valid operations and to hide inconsistency.

Attack 2

Attacker's goal is to hide inconsistency.

Time t1:

- Client C1 sends w1
- Client C2 sends w2

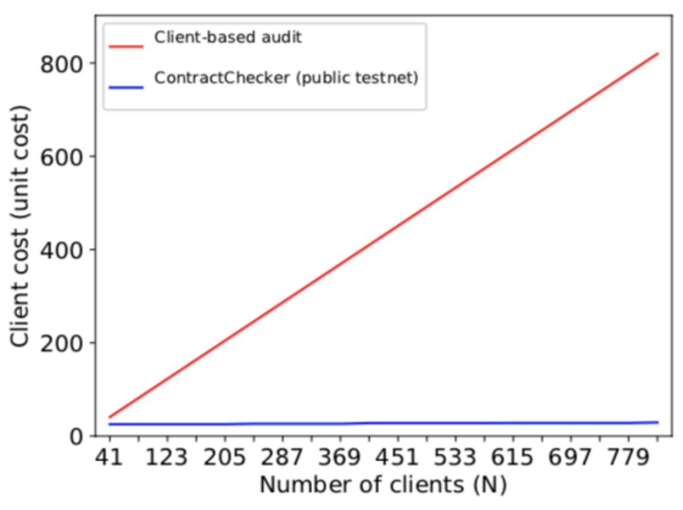
Time t2:

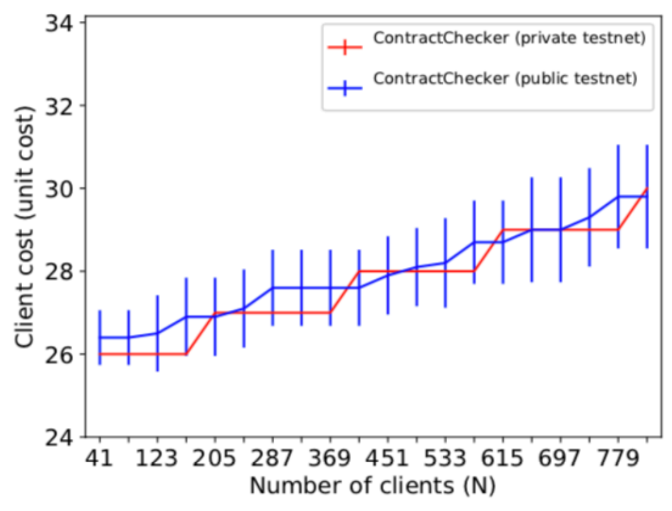
- Client C1 sends r3 receives w1
- Client C2 sends r4 receives w2

The attack succeeds when the server can convince C1 and C2 that their reads (r3 and r4) are consistent.

Evaluation (Cost)

- Measurements include number of clients, number of operations per epoch.
- ContractChecker saves client-side cost significantly.





The 4th NSF Secure and Trustworthy Cyberspace Principal Investigator Meeting (2019 SaTC PI Meeting)
October 28-29, 2019 I Alexandria, Virginia