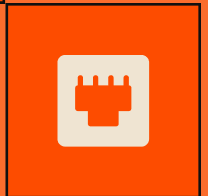
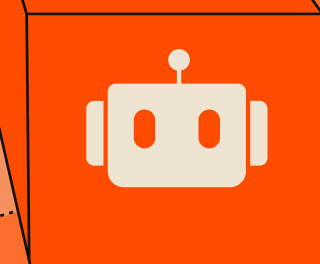
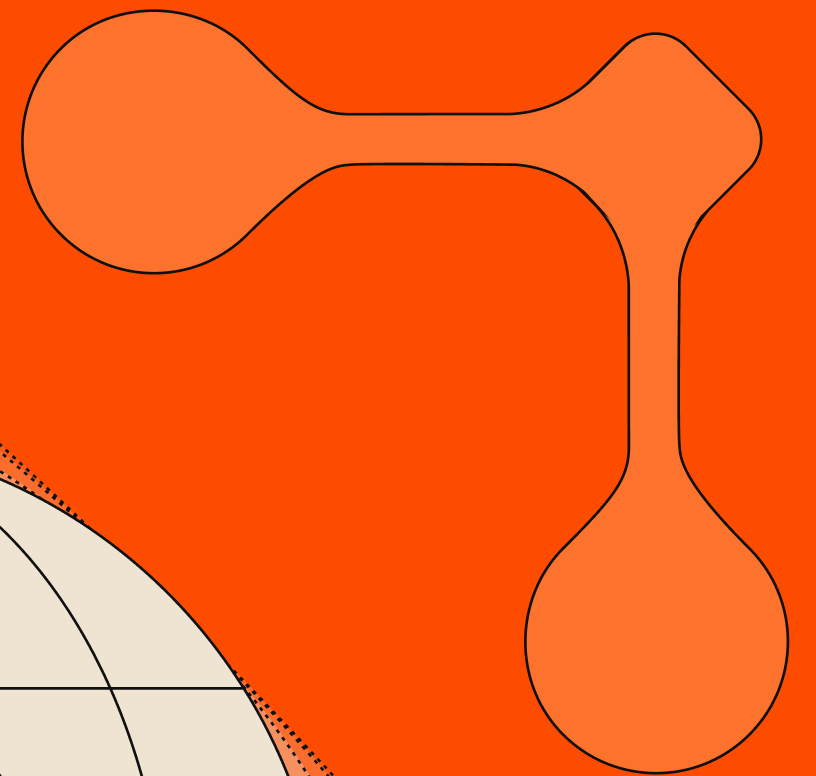
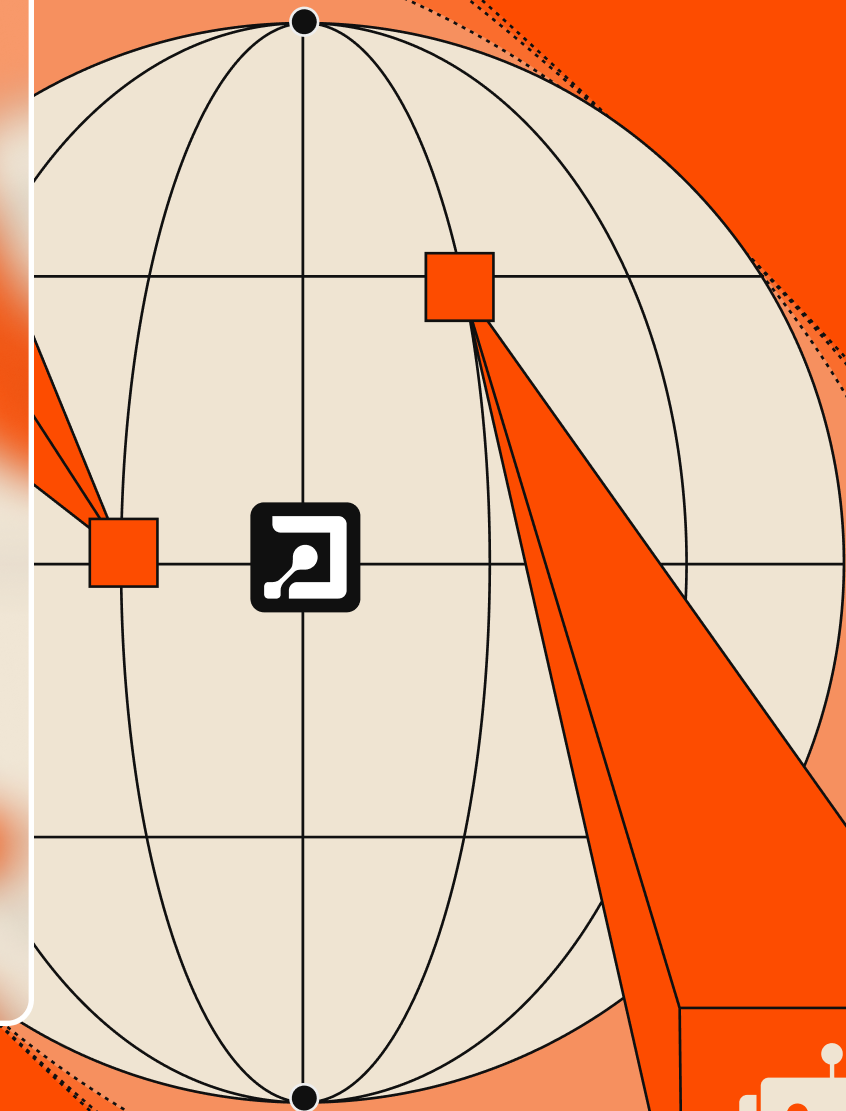
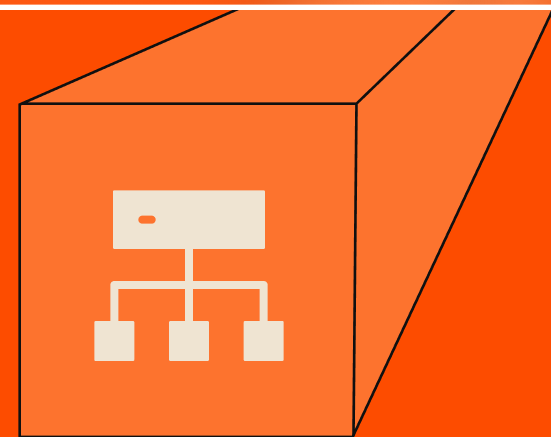


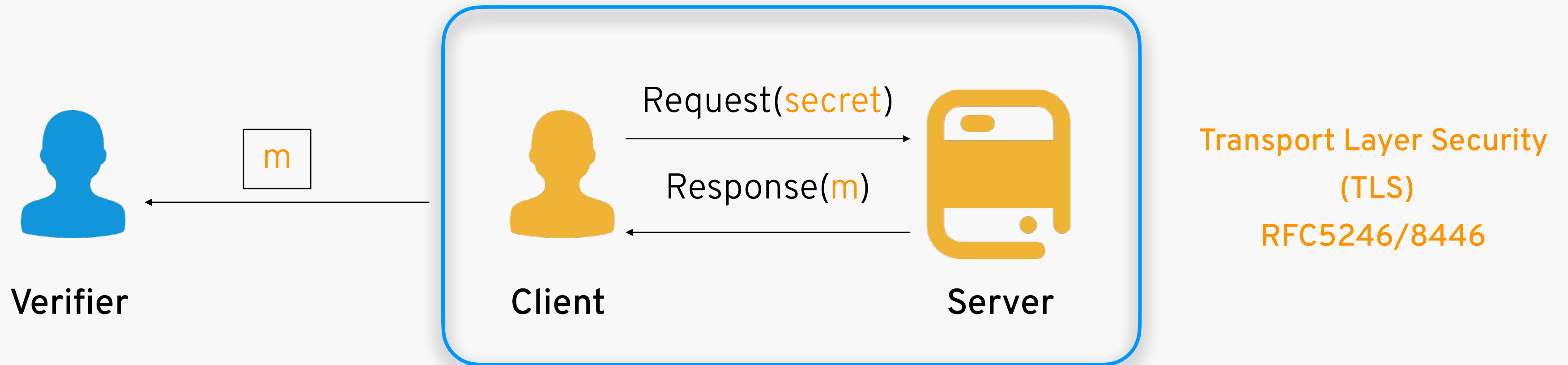
Deploying TLS Oracles Using Interactive ZK

Xiang Xie@Primus Labs

RWC 2025



Background



Client convinces Verifier that (sealed) m is retrieved from the Server,
 without any modifications on the Server side

Example: prove that I have enough ETH in Coinbase

Related Work

- This problem was introduced by Dan from TLS Notary about 10 years ago


Author

Topic: [tlsnotary - cryptographic proof of fiat transfer for p2p exchanges](#) (Read 42896 times)

[dansmith](#) (OP)
Full Member


Activity: 202
Merit: 100



[tlsnotary - cryptographic proof of fiat transfer for p2p exchanges](#)
April 11, 2013, 10:32:11 AM

#1

EDIT: 3 September 2014
The software now is near feature-complete, but in order to test its compatibility with the large number of banks out there, we need testers! Please, if you want to help freeing Bitcoin from the harassment of the banks, come and talk to us.

E-mail: tlsnotarygroup-at-gmail.com

Freenode IRC: [#tlsnotary-chat](#)

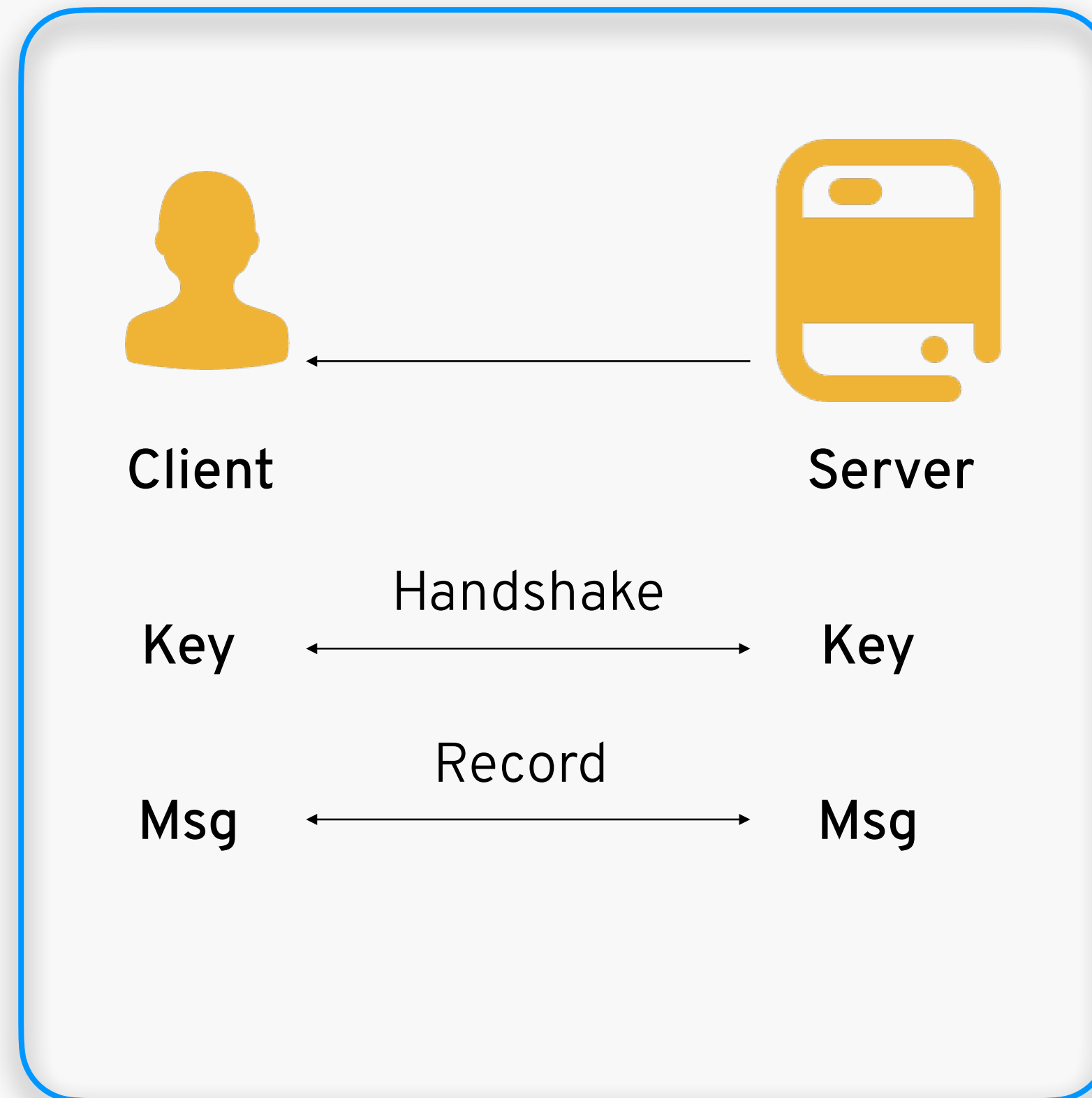
Code: <https://github.com/tlsnotary/tlsnotary>

Town Crier[ZCCJ+'16], DECO[ZMMG+'20], Janus[LEFS'23],
DiStefano[CDHP+'23], Garble-then-Prove[XYYWY' 24],
ORIGO[ELWG+'24], [LJSK'24] and many others

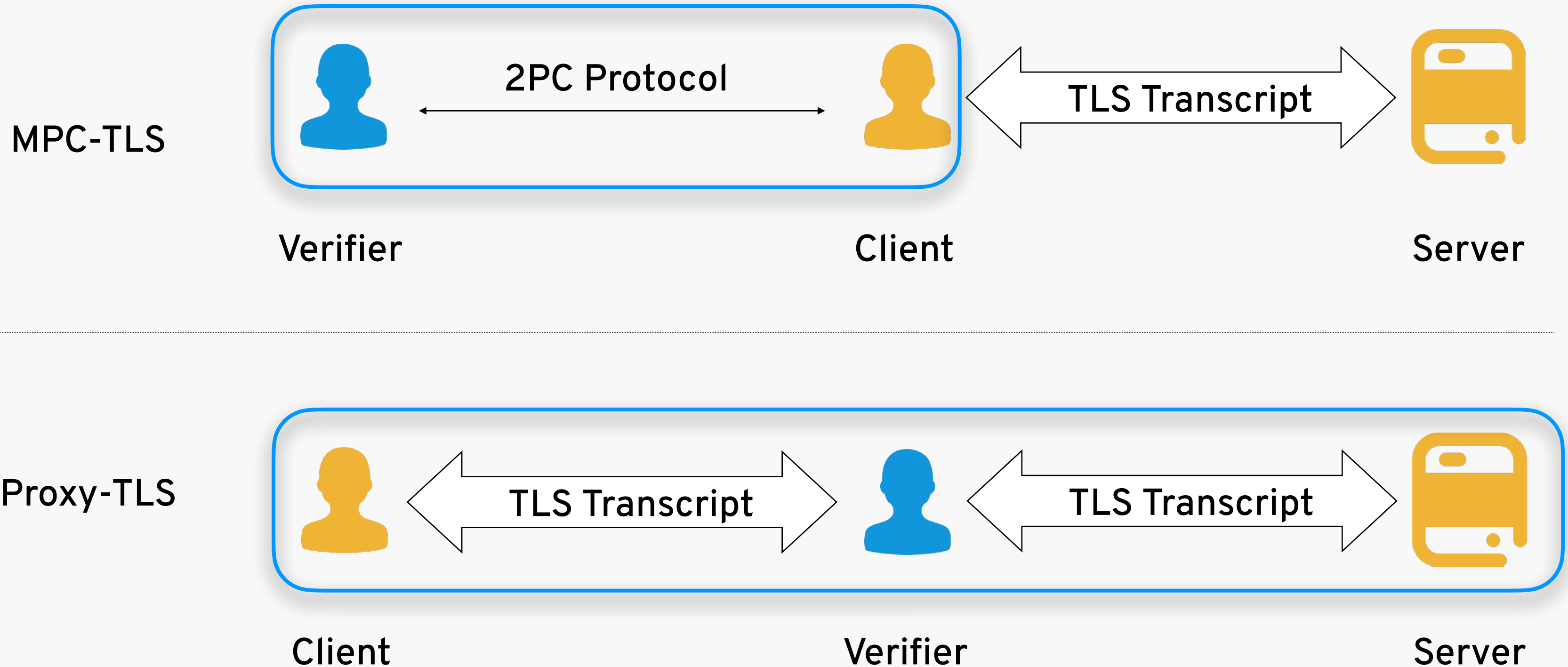
Transport Layer Security (TLS)

- Handshake Phase
 1. Certification
 2. Key Exchange
 3. Key Expansion (KDF)

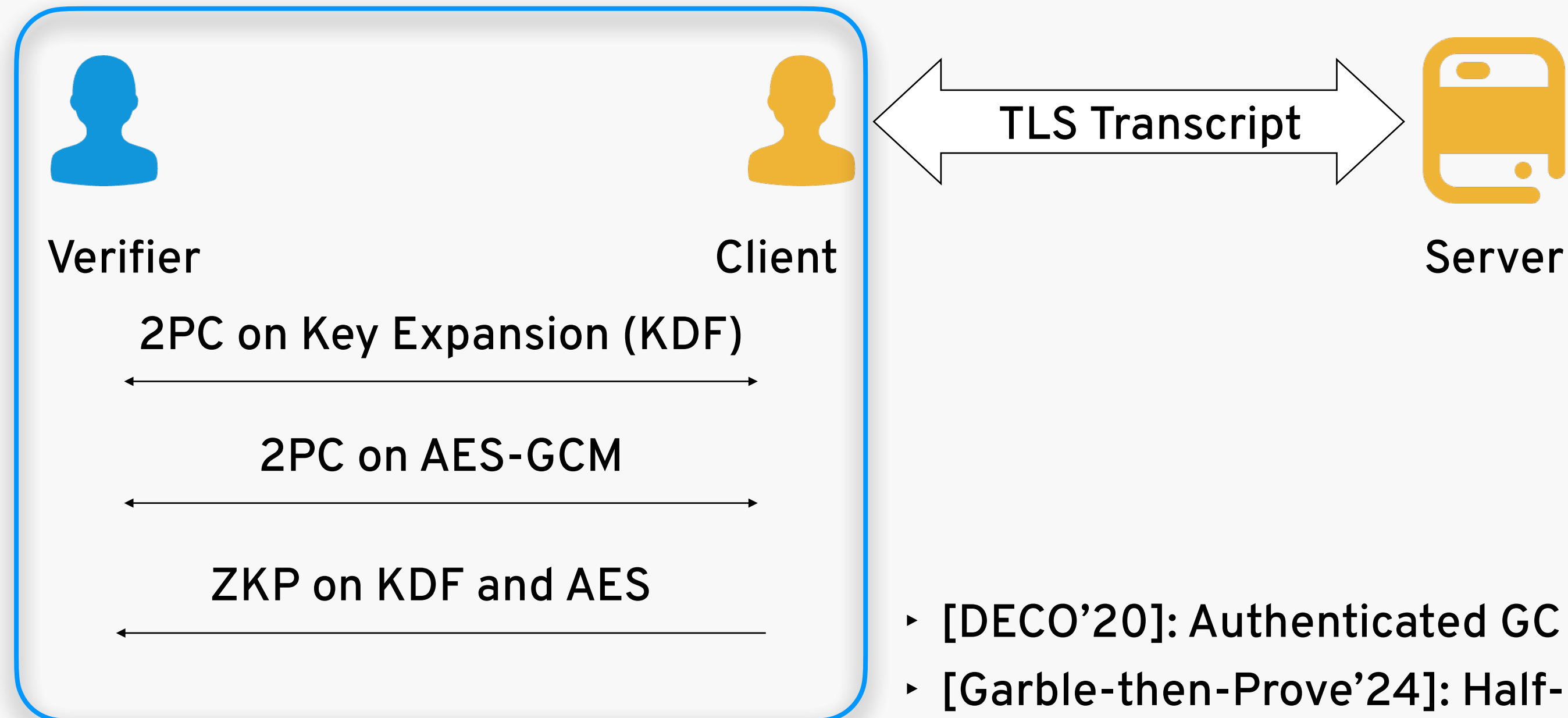
- Record Phase
 1. AEAD
 2. AES-GCM



Two Cryptographic Modes

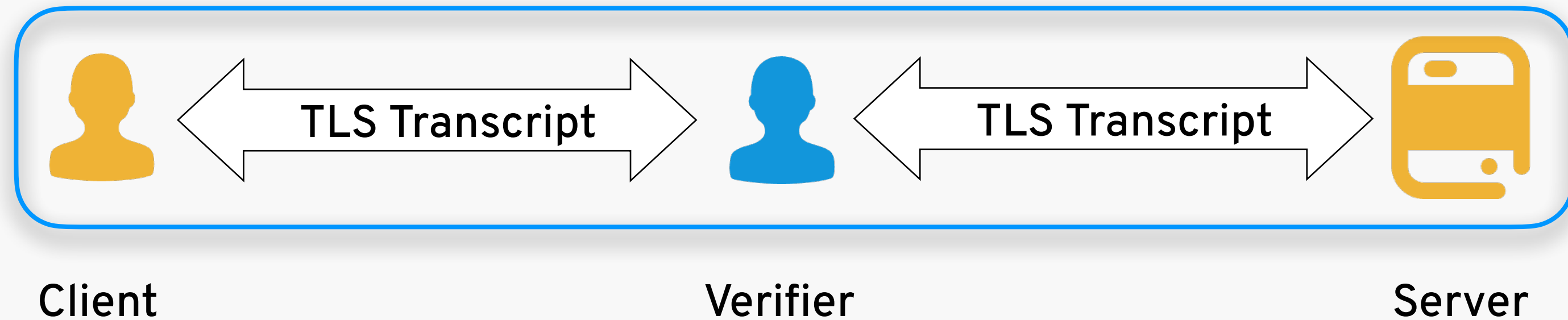


MPC-TLS



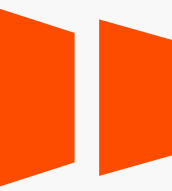
- [DECO'20]: Authenticated GC + zkSNARK
- [Garble-then-Prove'24]: Half-gate GC + QuickSilver

Proxy-TLS



Prove in zk client knows the
secret related to the ciphertexts

- [Garble-then-Prove'24]: Prove KDF + AES with QuickSilver
- [LJSK'24]: Prove AES (restricted to public padding) with zkSNARK
- [Origo'24]: Prove KDF + AES/ChaCha20 with zkSNARK



Comparisons of two modes

Mode	Trust Assumptions	Efficiency
MPC-TLS	Trust the TLS Server No collusion between client and verifier	Run 2PC protocol for KDF and AES Run zkp for KDF and AES
Proxy-TLS	Trust the TLS Server No collusion between client and verifier Verifier ensures the connection to TLS Server	Run zkp for KDF and/or AES

Why Interactive ZKP

1. Interactivity is acceptable

- The protocol does not need to be fully non-interactive.

2. KDF&AES inefficiency

- These functions are Boolean circuits, unsuitable for zkSNARKs.

3. Client-side constraints

- ZKPs usually run in resource-limited environments like browsers and mobile devices.

4. Performance bottlenecks

- Current zkSNARKs have high proving time and memory usage, posing a major challenge.

Interactive ZK — QuickSilver [YSWW'21]

- 1.9 USD → **one trillion** AND gates !
- 2.5 USD → **one trillion** MULT gates over a 61-bit field !

Instance Information			Boolean Circuits		Arithmetic Circuits	
Type	Price cents/hour	CPU	Speed gates/sec	Cost gates/cent	Speed gates/sec	Cost gates/cent
c6g.medium	1.9	ARM	5.3 M	10.0 B	2.2 M	4.1 B
c5.large	4.7	Intel	5.9 M	4.5 B	2.9 M	2.2 B
c5a.large	4.2	AMD	7.3 M	6.3 B	3.0 M	2.6 B

- 2 vCPU and 1GB Memory.
- Boolean circuits - network bandwidth 20Mbps.
- Arithmetic circuits - network bandwidth 500Mbps.

Performance

▸ Global scale experiments (MPC-TLS)

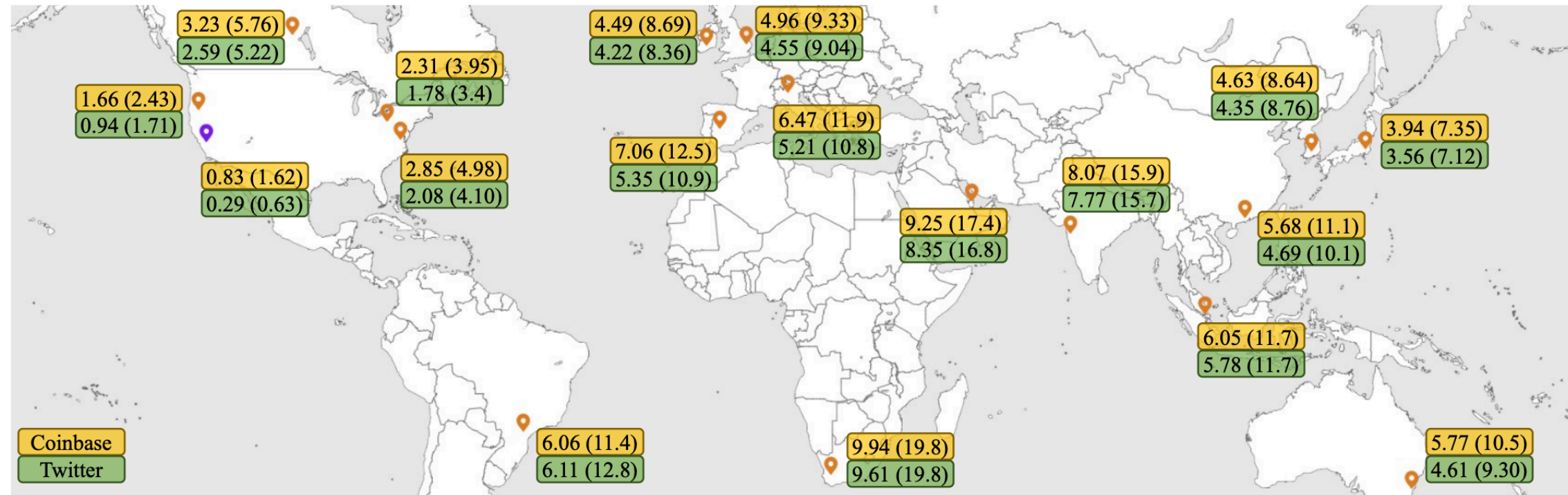


Figure 7: **Online and total performance of accessing Coinbase and Twitter servers with globally distributed provers.** All numbers are reported in seconds in the form of “online time (total time)”. The verifier is fixed at California. The server is hosted by Coinbase/Twitter, which may have mirrors in various locations.

- Prove large content from LLM (Proxy-TLS)
 - Proving a 200KB image generated from ChatGPT within 2mins with browser extension

Benchmark

- A zkTLS benchmark framework: <https://github.com/primus-labs/zktls-bench>
- Benchmark existing open source zkTLS implementations across various platforms (including X86/Arm/WASM) and network conditions.
- The garble-then-prove scheme is up to **an order of magnitude** faster than other MPC-TLS solutions.
- The QuickSilver based Proxy-TLS is up to **30x** and **145x** faster than alternatives.
- TLSNotary, an open source library by PSE, is transitioning to QuickSilver.

Products

All the SDKs support both MPC-TLS and Proxy-TLS



Developer Hub

- <https://dev.primuslabs.xyz/>
- A tool for developers to prove any data from any website



Web SDK

- Chrome extension
- JavaScript SDKs



Mobile SDK

- App clip/Instant app
- iOS/Android SDKs



Backend SDK

- SDKs for x86/Arm
- JavaScript SDKs

Deployment

360K

Successful
Attestations

160K

On-chain
Submissions

- Ethereum
- Linea
- BNB Chain
- Arbitrum
- Scroll
- Monad

540K

Chrome
Extension
Users

4.8K

Developer
Hub Users

Challenges

1. The verifier should be globally distributed and positioned close to the client to enhance user experience.
2. To mitigate the risk of collusion between the client and verifier, the best practice is to deploy the verifier within a TEE.

Thank you!

 github.com/primus-labs

 [@primus_labs](https://twitter.com/primus_labs)

 medium.com/@primuslabs

 primuslabs.xyz

