Ticket to Hide: Practical, Private Proofs of Provenance for TLS

Ryan Little[1], **Daniel Roche**[2], **Mayank Varia**[1]

[1]**Boston University**
[2]**US Naval Academy**

Problem Statement and Goals

How can you convince someone you obtained some data from a trusted TLS web server, without revealing more information than necessary?

To address this, we introduce **Ticket to Hide:** a cryptographic protocol designed around TLS 1.3. Our protocol

- Allows clients to convince third parties they received data from a trusted server and prove properties about the data, *without revealing it*.
- Hides the identity of the server among a set of *N* publicly known servers.
- Works immediately with real-world web servers: *servers only need to run regular TLS 1.3!*
- Has rigorously proven security and privacy guarantees, even against network adversaries.

Approach

We use two key cryptographic building blocks:

- **Secure Multi-Party Computation (MPC):** Allows parties to jointly evaluate a function on their private inputs, without revealing their inputs.
- **Zero-Knowledge Proofs (ZKPs):** Allows a party to convince another that some statement is true, without revealing any additional information.

Protocol Idea:

1. Let the server run regular TLS 1.3, while the client and verifier compute client-side TLS operations in MPC, each learning a secret share of the TLS encryption key.
2. Client and verifier jointly encrypt messages in MPC.
3. Client convinces the verifier with a ZKP that they obtained encrypted data from the server satisfying arbitrary properties.

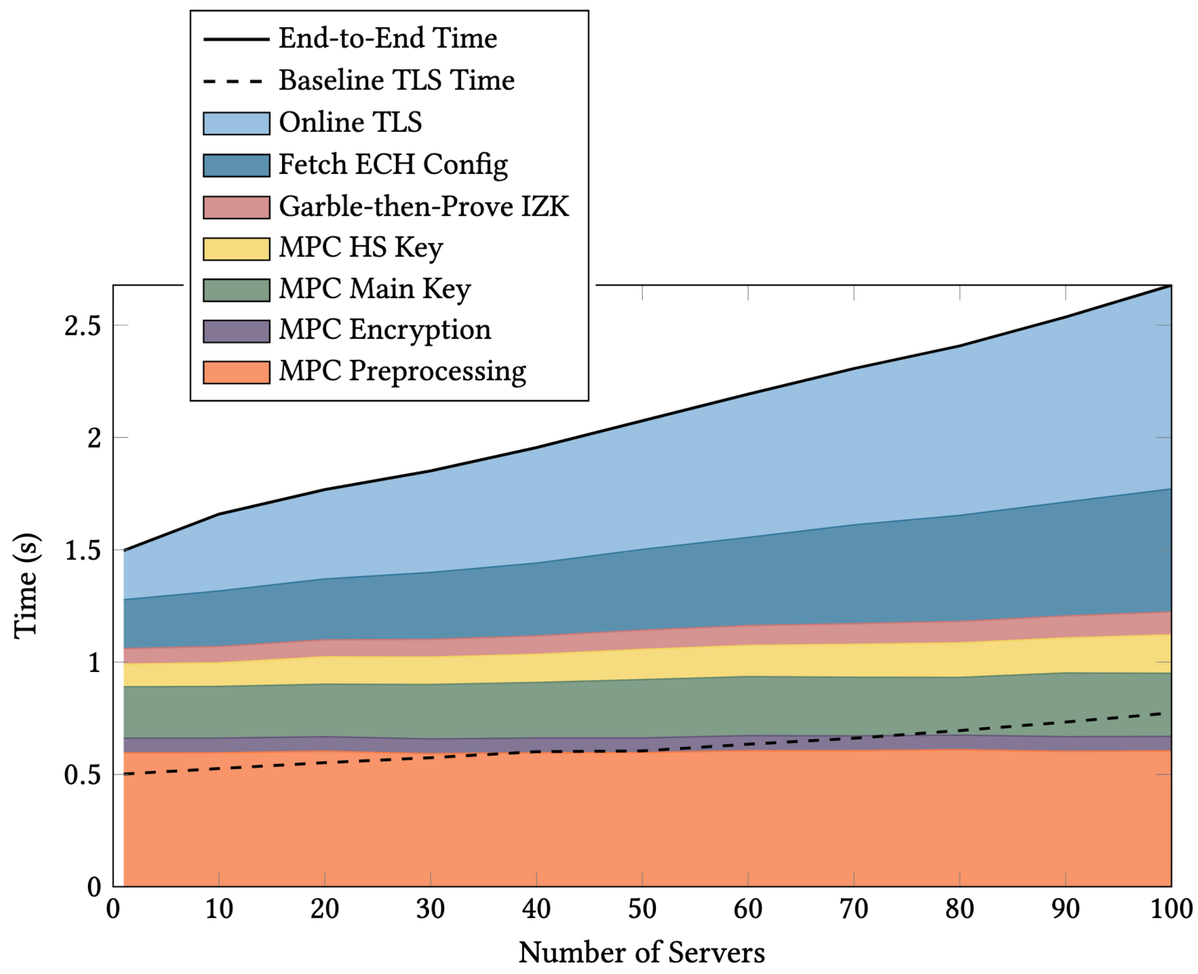
Results

Main Contributions:

- Runs 4-15x faster than prior solutions for TLS 1.3
- Identifies and fixes security issues in prior schemes
- Compatible with post-quantum key exchange algorithms

Comparison with Prior Approaches:

Scheme	Multi-Server?	Network Adversary?	TLS Version
DECO [1]	✗	✓	1.2
k-PECO [2]	✓	✗	1.2
Garble-then-Prove [3]	✗	✓	1.2
Janus [4]	✗	✓	1.3
DiStefano [5]	✓	✗	1.3
Ticket to Hide	✓	✓	1.3



End-to-end run time in the WAN setting

References:

[1] Fan Zhang, Deepak Maram, Harjasleen Malvai, Steven Goldfeder, and Ari Juels. *DECO: Liberating web data using decentralized oracles for TLS*. ACM CCS 2020.

[2] Manuel B. Santos. *PECO: methods to enhance the privacy of DECO protocol.* Cryptology ePrint Archive, 2022.

[3] Xiang Xie, Kang Yang, Xiao Wang, and Yu Yu. *Lightweight authentication of web data via garble-then-prove*. USENIX Security 2024.

[4] Jan Lauinger, Jens Ernstberger, Andreas Finkenzeller, and Sebastian Steinhorst. *Janus: Fast privacy-preserving data provenance for TLS*. PETS 2025.

[5] Sofía Celi, Alex Davidson, Hamed Haddadi, Gonçalo Pestana, and Joe Rowell. *DiStefano: Decentralized infrastructure for sharing trusted encrypted facts and nothing more*. NDSS 2025.

Ryan Little
Boston University

BOSTON UNIVERSITY

Ticket to Hide: Practical, Private Proofs of Provenance for TLS

Ryan Little[1], **Daniel Roche**[2], **Mayank Varia**[1]

[1]**Boston University**
[2]**US Naval Academy**

Abstract

When using Transport Layer Security (TLS), web users can connect to a server and trust that they are sending and receiving data with the intended web server. This guarantee, however, is not transferable: there is no immediate way for a client to convince an external party that a transcript or message originated from a particular server. Beginning with the DECO protocol of Zhang et al. [1], there has been a line of work on "TLS oracles"–cryptographic protocols that allow a client to commit to, prove provenance, and disclose arbitrary properties of TLS application data to a verifier party. TLS oracles only require the server to run standard TLS, making them compatible with existing real-world web servers.

In this work we introduce Ticket to Hide, a new TLS oracle protocol for TLS 1.3. We operate in the multi-server setting, previously explored in the DiStefano protocol by Celi et al. [2], in which the client additionally wishes to hide the identity of the server they are communicating with among a set of N publicly known servers. We leverage new features of TLS 1.3 in surprising ways to yield performance and security benefits, resulting in a protocol that is both faster and more private than previous work. Additionally, we are the first TLS oracle protocol to be compatible with post-quantum secure TLS key agreement and certificates. Our implementation, which builds on top of the Garble-then-Prove framework of Xie et al. [3], scales to N=100 servers in less than 3 seconds of end-to-end time in a WAN setting–only 3.5× the latency of a normal TLS 1.3 interaction.

References:

[1] Fan Zhang, Deepak Maram, Harjasleen Malvai, Steven Goldfeder, and Ari Juels. *DECO: Liberating web data using decentralized oracles for TLS*. ACM CCS 2020.

[2] Xiang Xie, Kang Yang, Xiao Wang, and Yu Yu. *Lightweight authentication of web data via garble-then-prove*. USENIX Security 2024.

[3] Sofía Celi, Alex Davidson, Hamed Haddadi, Gonçalo Pestana, and Joe Rowell. *DiStefano: Decentralized infrastructure for sharing trusted encrypted facts and nothing more*. NDSS 2025.

Ryan Little
Boston University

BOSTON UNIVERSITY