

Secure Message Transport (SMT)

Michio Honda

University of Edinburgh

3 July 2026, Coseners



THE UNIVERSITY of EDINBURGH
informatics

With acknowledge to Tianyi Gao, Xinshu Ma, Eugenio Luo, Steven Chien, Suhas Narreddy, Moray Li

SMT overview

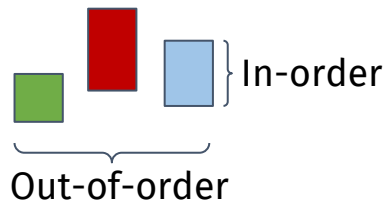
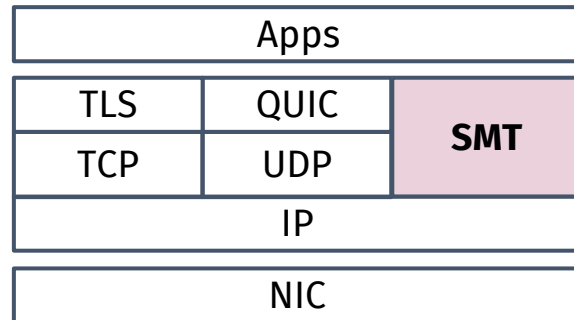
Designing Transport-Level Encryption for Datacenter Networks

Tianyi Gao, Xinshu Ma, Suhas Narreddy, Eugenio Luo, Steven W. D. Chien, and Michio Honda
University of Edinburgh

Abstract—Cloud applications need network data encryption to isolate from other tenants and protect their data from potential eavesdroppers in the network infrastructure. This paper presents SMT, a protocol design for emerging datacenter

properties of datacenter transports while supporting the same threat model as TLS/TCP—to protect the endpoints from data breach, packet injection, and replay attacks. SMT uses per-message TLS record sequence number space in the authenticated session, while guaranteeing message uniqueness

- Transport from Homa
 - Message-based abstraction
 - Out-of-order messages
 - Each message is reliable
- Security from TLS
 - app-to-app confidentiality, integrity, non-replayability..

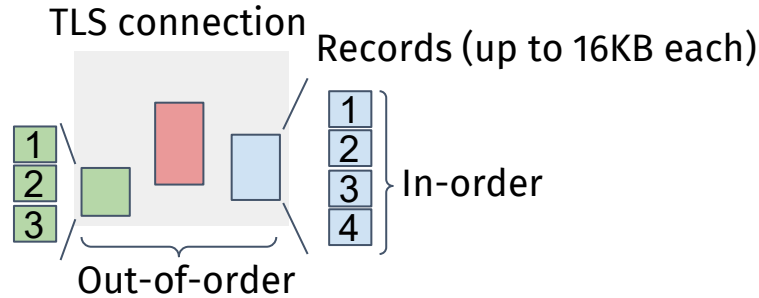


TLS incompatibility with Homa

- Alternatives
 - DTLS: No TCP segmentation offload
 - PSP/IPsec: No TLS-style trust/guarantees
- TLS needs in-order delivery
 - Records must be processed and decrypted in-order

SMT approach: Per-message record sequence number

- Every message has own record seqno space



- Higher bits of record seqno encode a unique message ID
 - Avoid nonce duplicate between seqno spaces across the connection

TLS offload & TSO

**TLS security
guarantees**

Homa parallelism

Implementation

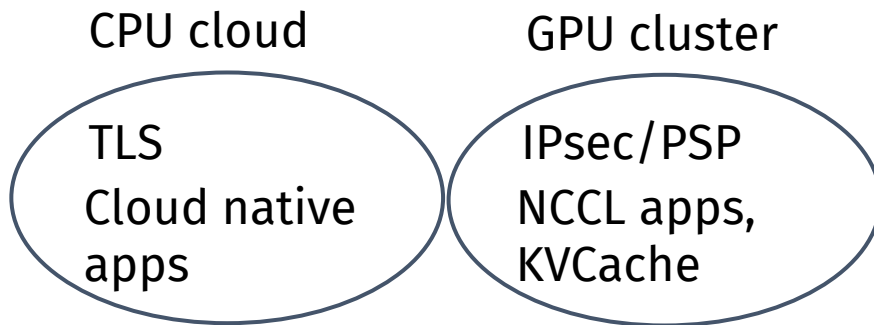
- Linux
 - Patch to Homa/Linux and mlx5 driver
- FreeBSD
 - Minimum (no congestion control or hw offload)
 - Presented at BSDCan last month

Is a new transport hopeful?

- The Internet is ossified already
 - Cloud networks are also not middlebox-free
- Offload compatibility can be preserved
- Homa protocol number (IANA 146) goes through AWS, GCP, Digital Ocean and Oracle cloud internal network

Next steps: GPU networking

- Extend CPU-networking trust semantics to GPU networking
 - They are currently isolated
- TLS is unfit
 - GPU apps send parallel chunks
- SMT for parallelism
 - Encryption in GPUs or NICs



Summary

- SMT: Message-based, encrypted transport for datacenters
 - Linux kernel implementation as Homa/Linux extension
 - 10-35% latency improvement over TLS/TCP
 - 16-41% throughput improvement
 - Under active development, Linux and FreeBSD
 - AWS, GCP, Digital Ocean, Oracle clouds pass new transports!

<https://github.com/uoenoplab/smt>