TNIC

A Trusted NIC Architecture

Dimitra Giantsidi, Julian Pritzi, Felix Gust,
Antonios Katsarakis, Atsushi Koshiba, Pramod Bhatotia

ASPLOS 2025

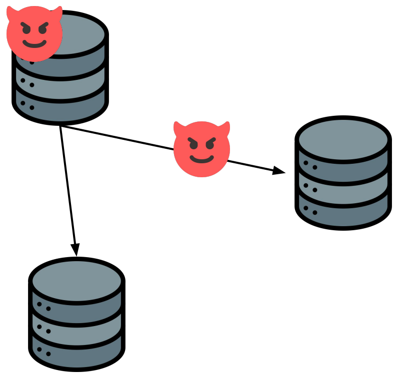
Distributed systems in the cloud

- Distributed systems are the cloud computing foundations
  - scalability
  - performance
- However, distributed systems are prone to failures!
  - machines can fail
- How to make distributed systems fault tolerant?

Crash Fault Tolerance (CFT) makes systems fault tolerant

Crash Fault Tolerance (CFT)

- CFT model handles benign failures
  - requires $2f+1$ nodes to handle f failures
- However, insufficient in the untrusted cloud
  - e.g., untrusted nodes, malicious attackers
  - arbitrary (**Byzantine**) failures go undetected

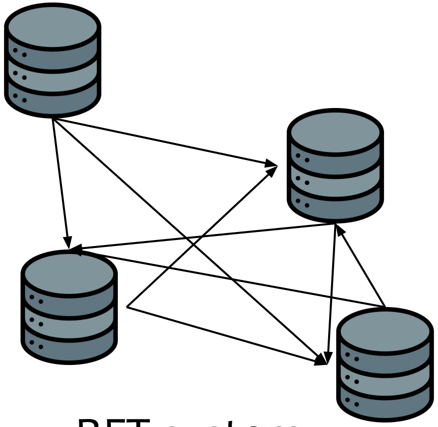


CFT system
($2f+1$ machine nodes)

CFT systems are **not well-suited** for the untrusted cloud infrastructure

Byzantine Fault Tolerance (BFT)

- BFT model handles arbitrary failures
  - requires ***3f+1*** nodes to handle *f* failures
- However, BFT is costly
  - limited scalability (***f*** more nodes than CFT)
  - complexity and high-latency



BFT system
(***3f+1*** machine nodes)

BFT's low scalability impedes its adoption in the untrusted cloud

Trusted computing for BFT systems

- Foundational building block for trustworthy systems
  - CPU-based ***Trusted Execution Environments (TEEs)***
- TEEs can ensure a node to follow the protocol faithfully
- Therefore, TEEs can improve scalability in BFT systems
  - requires **$2f+1$ nodes,** the same as CFT systems

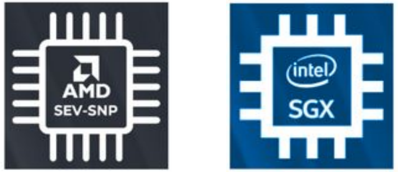
arm TRUSTZONE

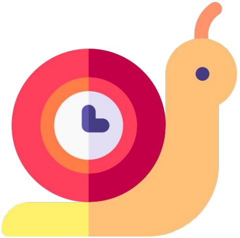
Keystone

AMD SEV-SNP

Trusted computing can make BFT systems practical, ***but…***

Limitations of CPU-based TEEs





#1: Heterogeneity

E.g., AMD-SEV's confidentiality vs. Intel SGX's integrity

#2: Large TCB

E.g., 2M LoC on AMD-SEV and Intel TDX TCBs

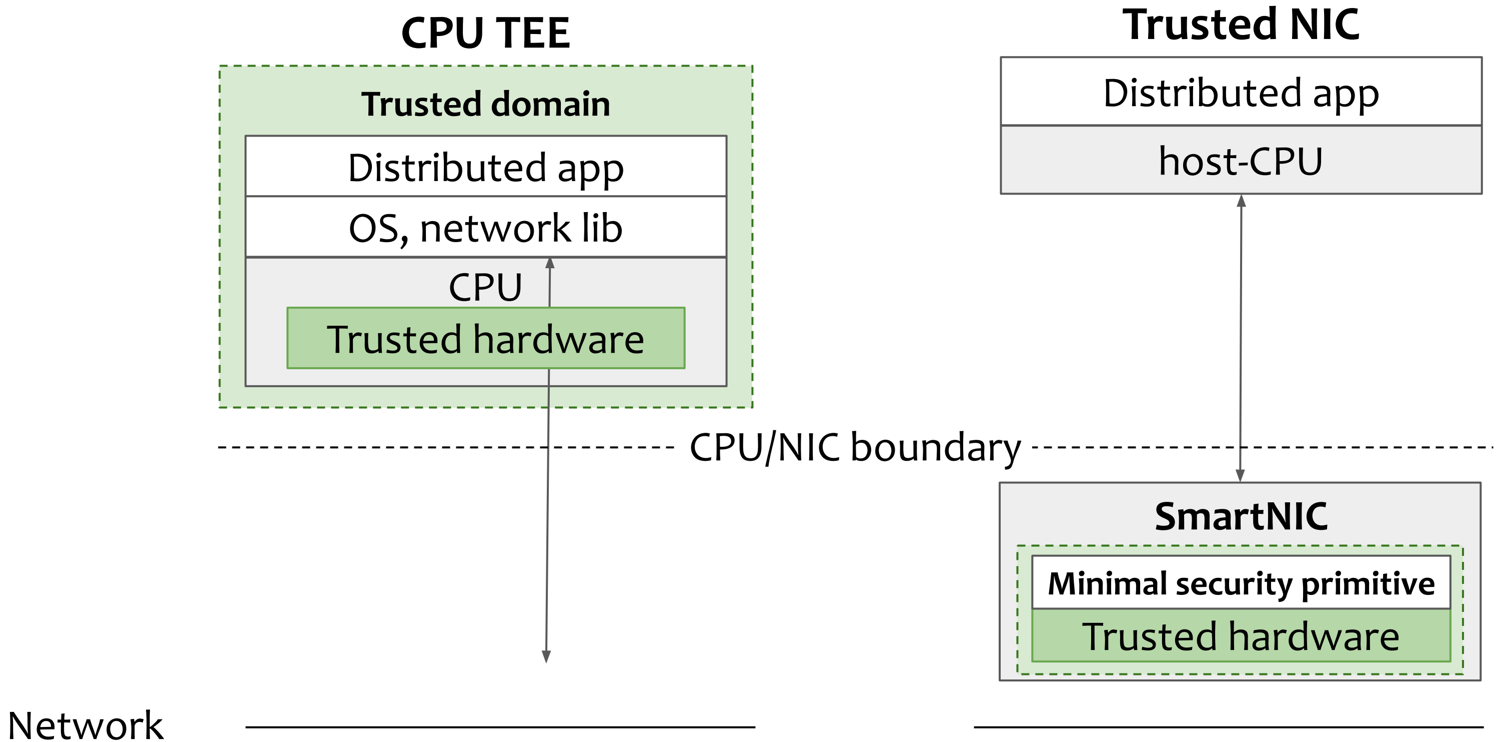
#3: Low performance

E.g., syscalls, virtualization overheads, world switches

Research question

How do we design **trustworthy distributed systems for Byzantine cloud environments** while overcoming the limitations of CPU-based TEEs?

Key insight: Moving trusted computing into a NIC



CPU-dependent, large TCB, slow

CPU-agnostic, small TCB, fast

Our proposal

TNIC: A Trusted NIC Architecture
A hardware-network substrate for building
high-performance, trustworthy distributed systems

Properties:

- Uniform interface
  - host CPU-agnostic
- Minimalism
  - small TCB with verified security properties
- Performance
  - hardware-offloading of security processing

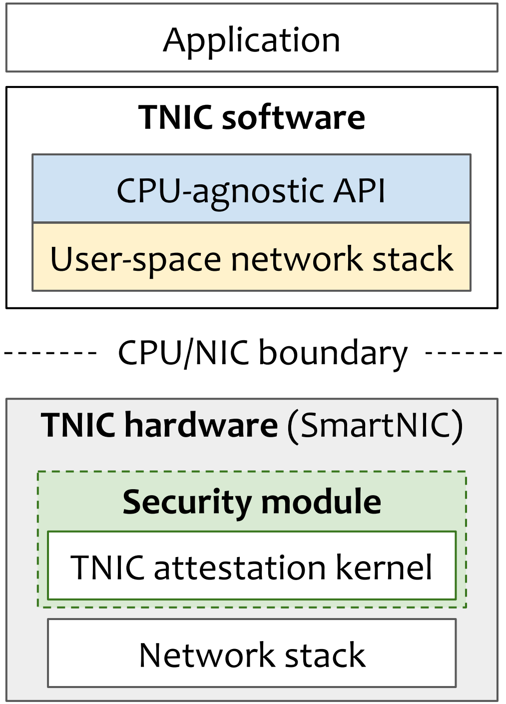
Outline

- ~~Motivation~~
- Overview
- Evaluation

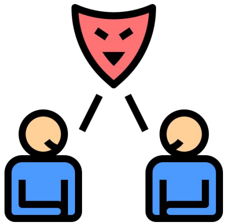
TNIC overview

- TNIC software
  - CPU-agnostic API
  - user-space networking

- TNIC hardware
  - guarantees two security properties for BFT:
    #1 Non-equivocation
    #2 Transferable authentication

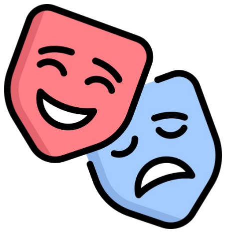


Key ingredients for trustworthy distributed systems



#1: Non-equivocation

Do not make conflicting statements
to different nodes



#2: Transferable authentication

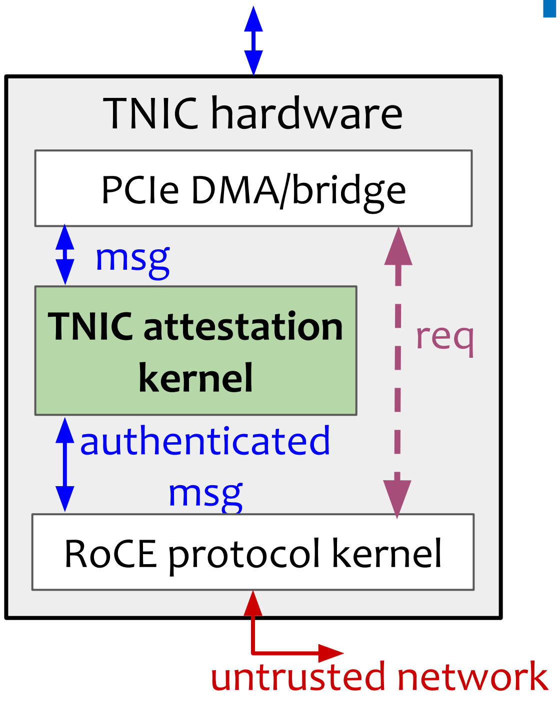
Be capable of verifying
the original sender of the message

Allow systems to operate with ***2f+1*** nodes in Byzantine environments[1]

[1]On the (limited) power of non-equivocation, PODC'12.

TNIC hardware

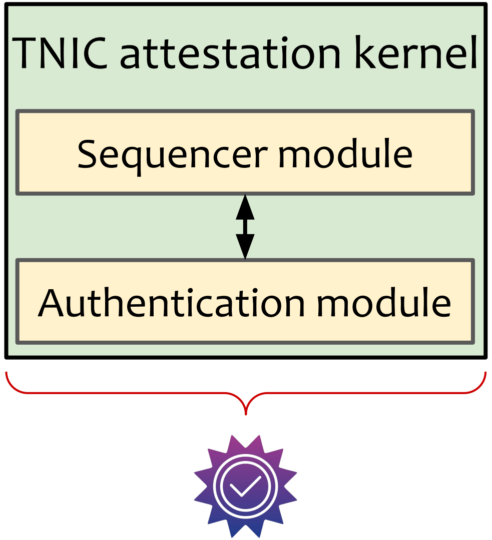
- TNIC attestation kernel
  - non-equivocation
  - transferable authentication
- RoCE protocol kernel
  - RDMA operations
- Separate data and control path



TNIC attestation kernel authenticates (and verifies) RDMA-driven messages

TNIC attestation kernel

- Attest and verify operations
  - generates and verifies authenticated messages
- Authentication module
  - guarantees transferable authentication
  - computes cryptographic MAC
- Sequencer module
  - guarantees non-equivocation
  - assigns monotonically increased numbers to messages (and verifies them)

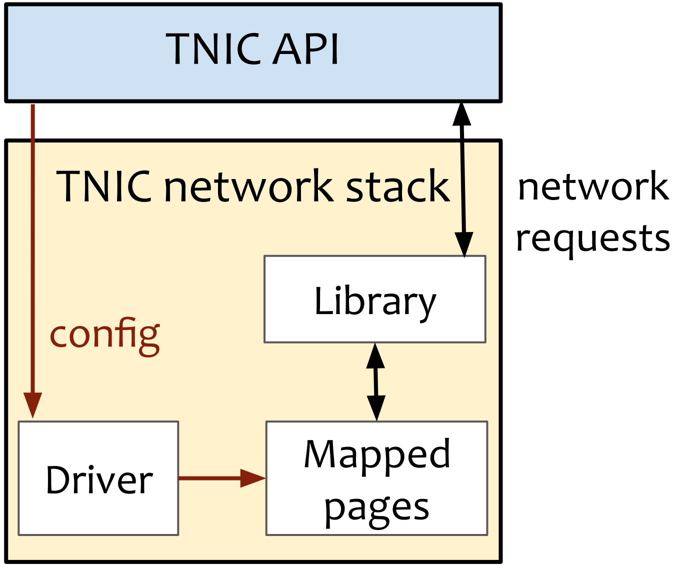


Formally verified

TNIC attestation kernel is minimal and formally verified

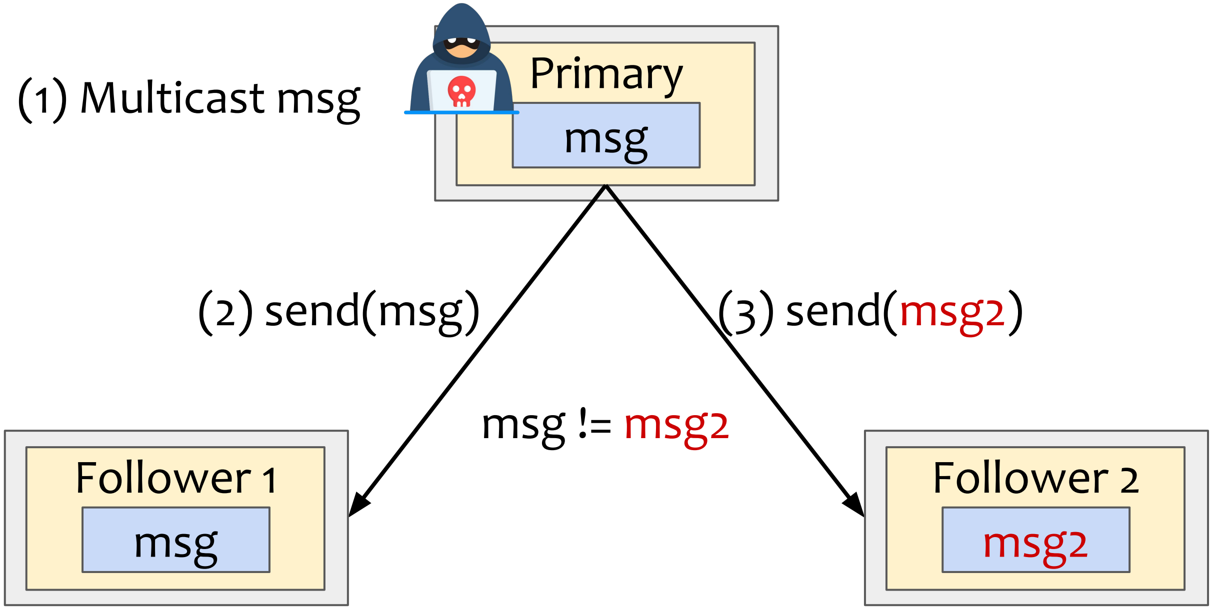
TNIC network stack and API

- TNIC network stack
  - driver enables user-space device access
  - library for RDMA support
- TNIC API
  - trusted message format
  - peer-to-peer trusted operations
  - group communication primitives



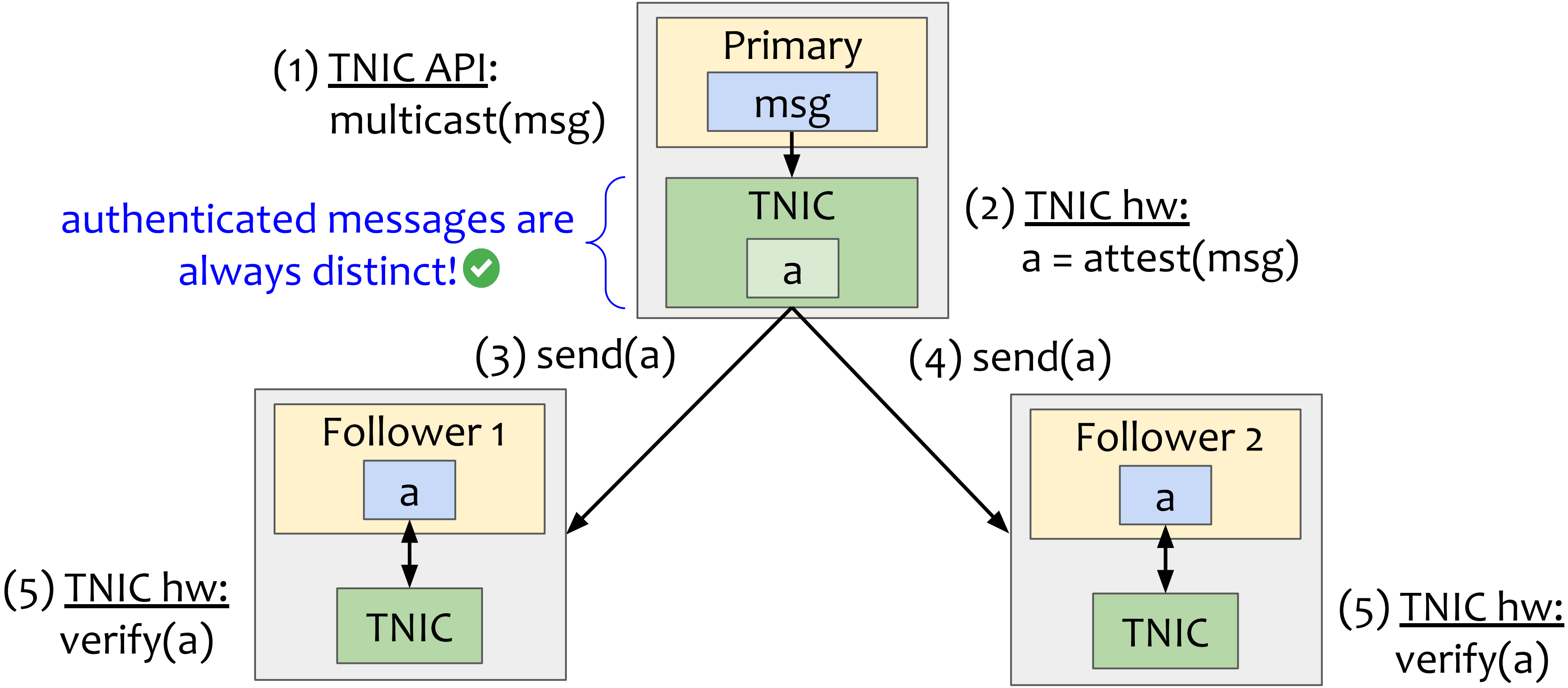
TNIC implements user-space **trusted** networking

Multicast under equivocation attack



Untrusted distributed system

TNIC in action: equivocation-free multicast



Trustworthy distributed system

Outline

- ~~Motivation~~
- ~~Overview~~
- Evaluation

Evaluation

Questions:

- What is the performance of TNIC?
- What is the performance for the trusted systems?

Experimental setup:

- HW evaluation on 2 Alveo U280 FPGA NICs
- Distributed systems evaluation on 3x Intel i9-9900K @3.60GHz

Evaluation

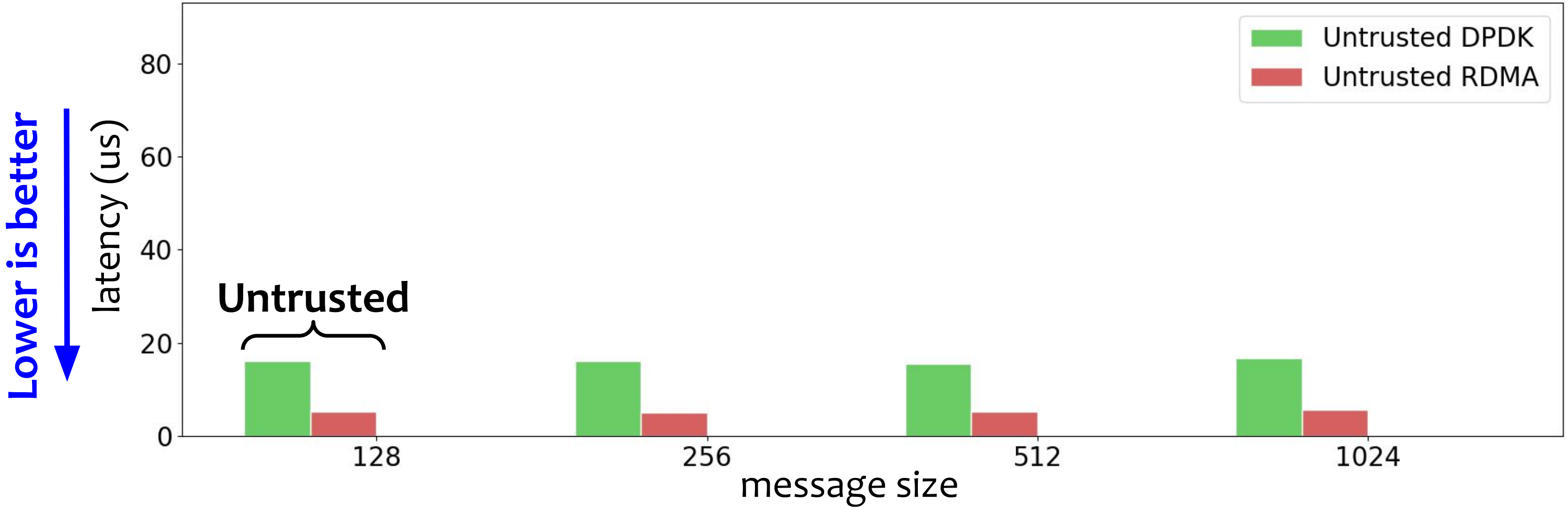
Questions:

- What is the performance of TNIC?
- What is the performance for the trusted systems?

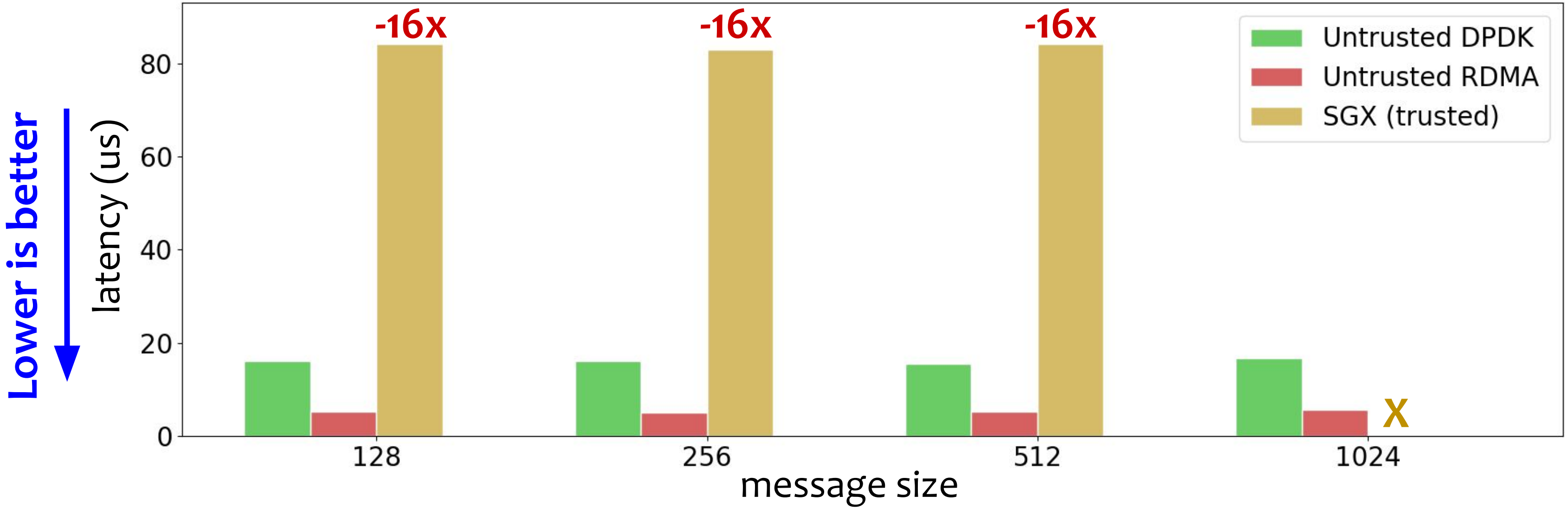
Experimental setup:

- HW evaluation on 2 Alveo U280 FPGA NICs
- Distributed systems evaluation on 3x Intel i9-9900K @3.60GHz

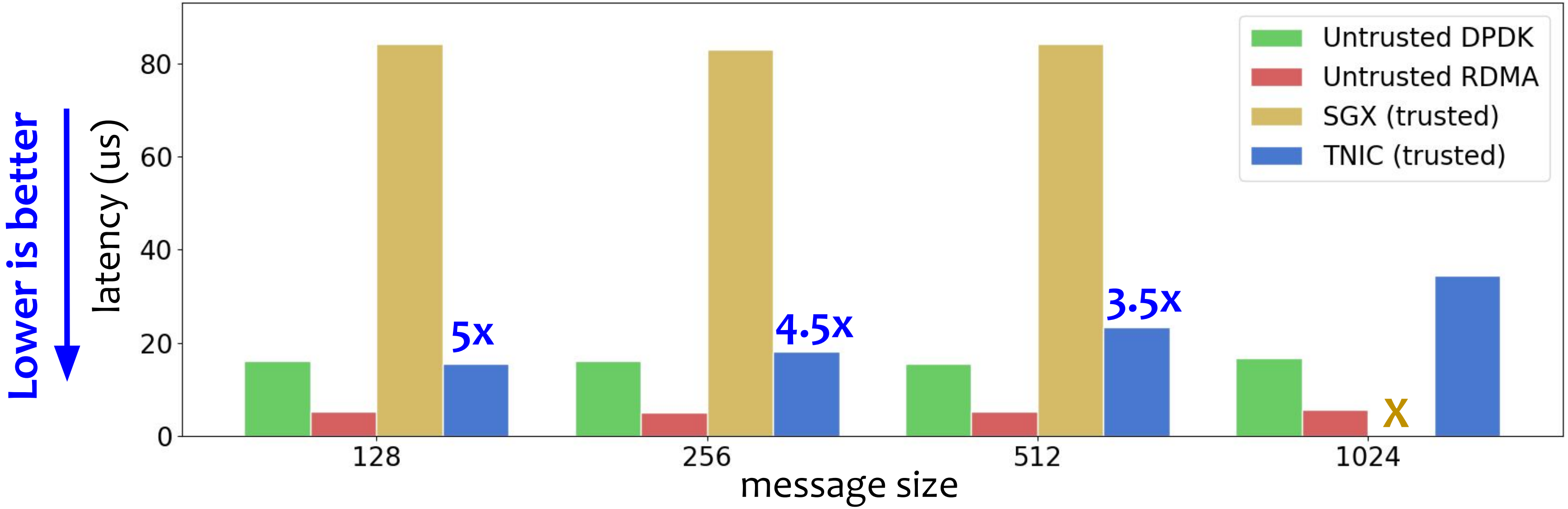
Q1: TNIC performance



Q1: TNIC performance



Q1: TNIC performance



TNIC is **up to 5x faster** w.r.t. a TEE-based network stack

Evaluation

Questions:

- What is the performance of TNIC?
- What is the performance for the trusted systems?

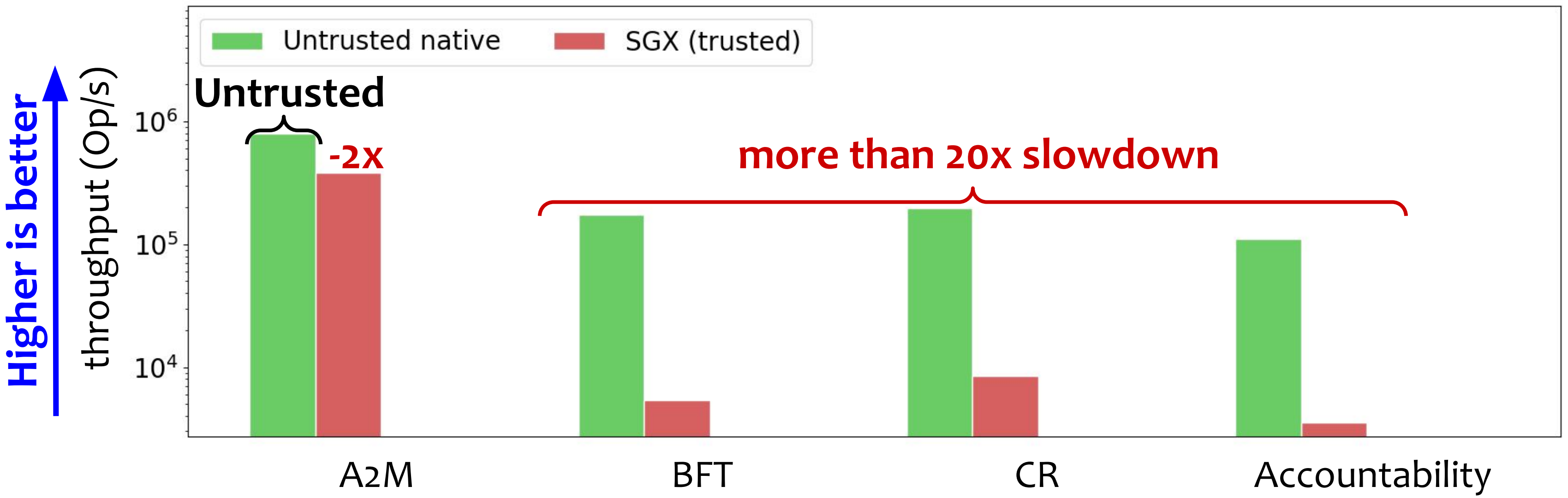
Experimental setup:

- HW evaluation on 2 Alveo U280 FPGA NICs
- Distributed systems evaluation on 3x Intel i9-9900K @3.60GHz

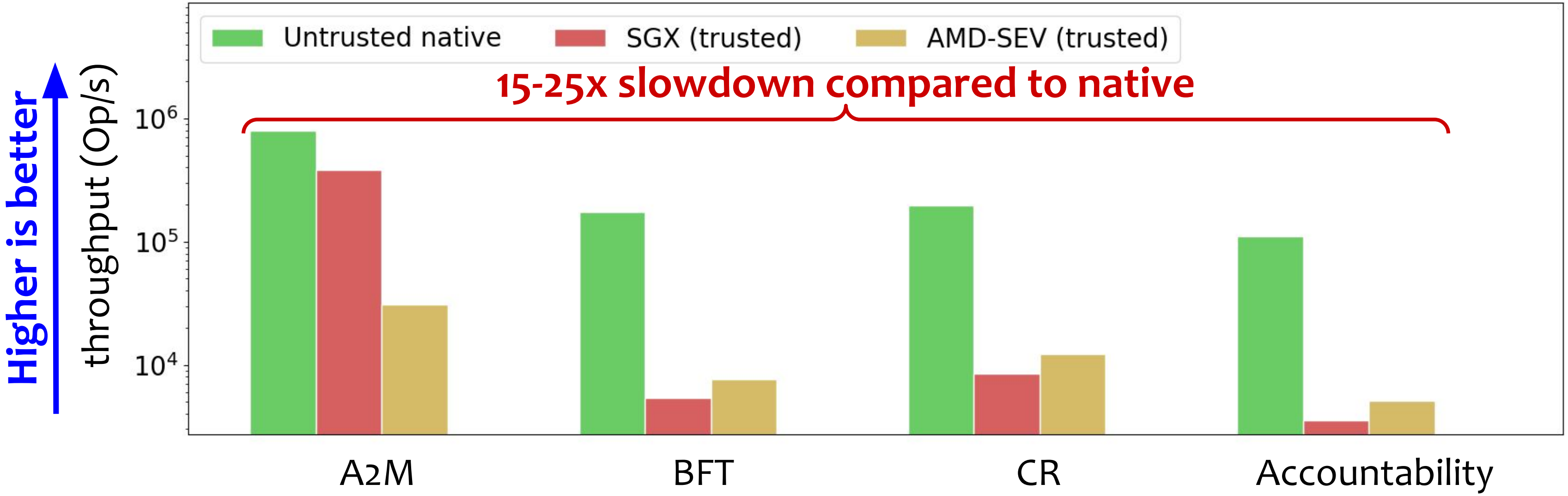
TNIC application on distributed systems

- Attested-Append-Only-Memory (A2M) **[SOSP'07]**
  - append-only log in the untrusted memory
- BFT **[OSDI'99]**
  - broadcast-based protocol with a unique leader
- Chain Replication (CR) **[OSDI'04]**
  - nodes organized as a chain
- PeerReview accountability protocol **[SOSP'07]**
  - failure detection

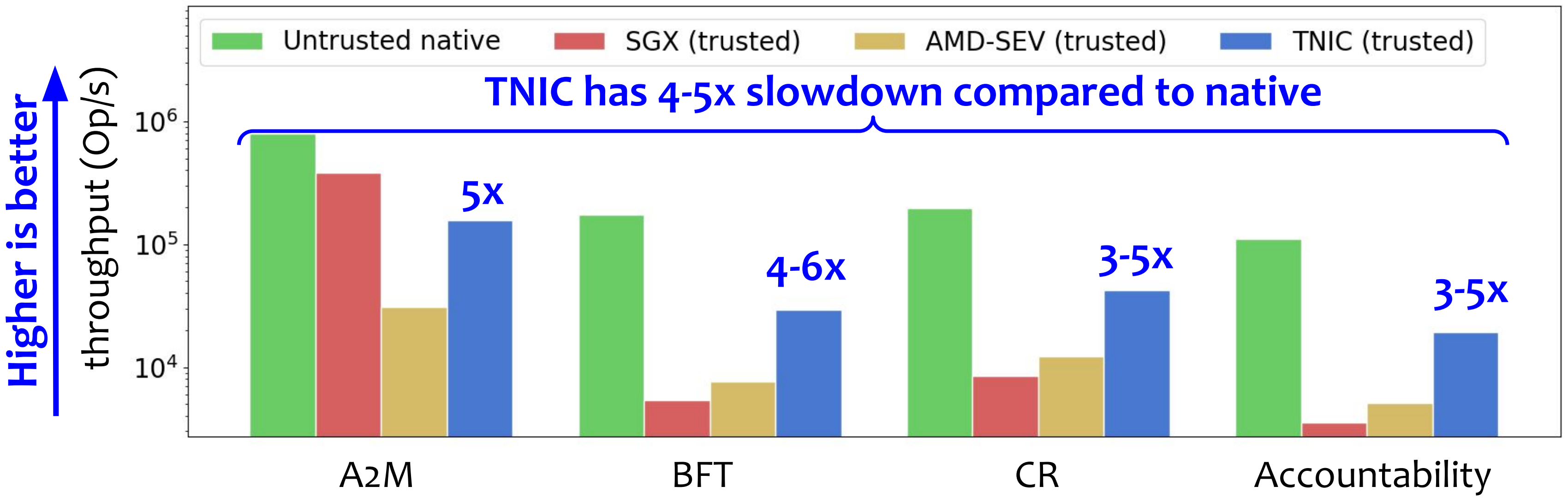
Q2: Performance of trusted systems



Q2: Performance of trusted systems



Q2: Performance of trusted systems



TNIC offers at least **3x better throughput** w.r.t. to TEE-based trusted systems

Summary

CPU-based TEEs for efficient trustworthy distributed systems are not a good fit!

- heterogeneity in security properties, programmability and performance
- large TCBs with vulnerabilities that go undetected
- performance overheads

TNIC: A trusted NIC architecture

- **CPU-agnostic** network APIs
- **minimal and verified security properties**
- hardware-offloaded **high-performance** networking

Paper **Code**