Hermes

A practical SPARTA-LL Implementation

Surendra Jammishetti

Mar 16, 2025

CSE 108C

Problem Defenition

- Current anonymous messaging systems aren't resilient to traffic analysis attacks.
- SPARTA lays a framework for a fast, traffic analysis resistant solution.
- Common pitfalls, such as user validation, that can starve user messages.
- Lacks details for multi-device communication.

Threat Model and Security Guarantees

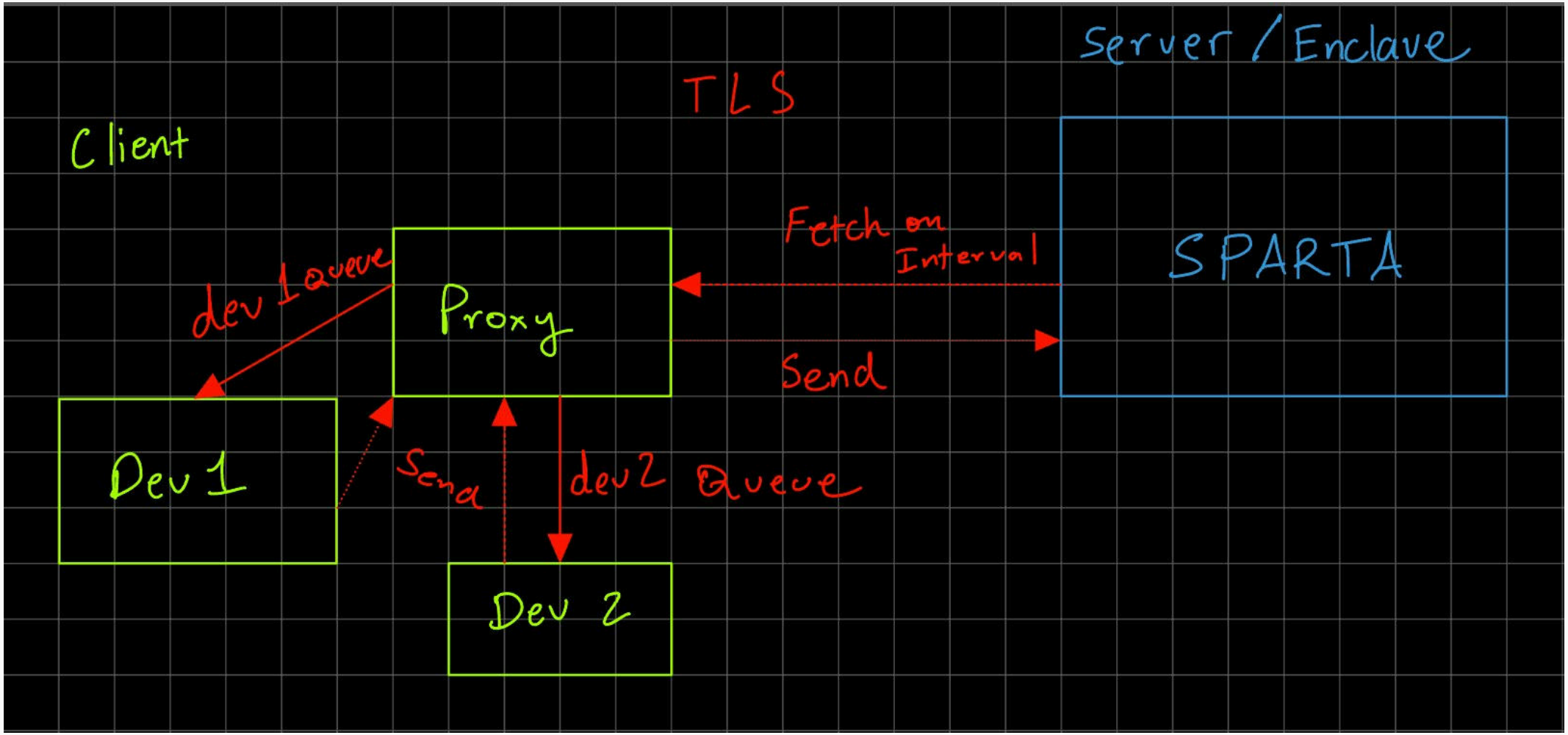
Adversary

- Inheriting SPARTA's threat model of a global active attacker who can
  - control / modify all network links
  - participate in the protocl
  - observe traffic for an arbitrary amount of time
  - can breach everything on the server excluding the enclave code

Differential Privacy

- Guarantee that adversary cannot corrolate that one user is messaging another.
  - The base SPARTA-LL construction already acheives this.

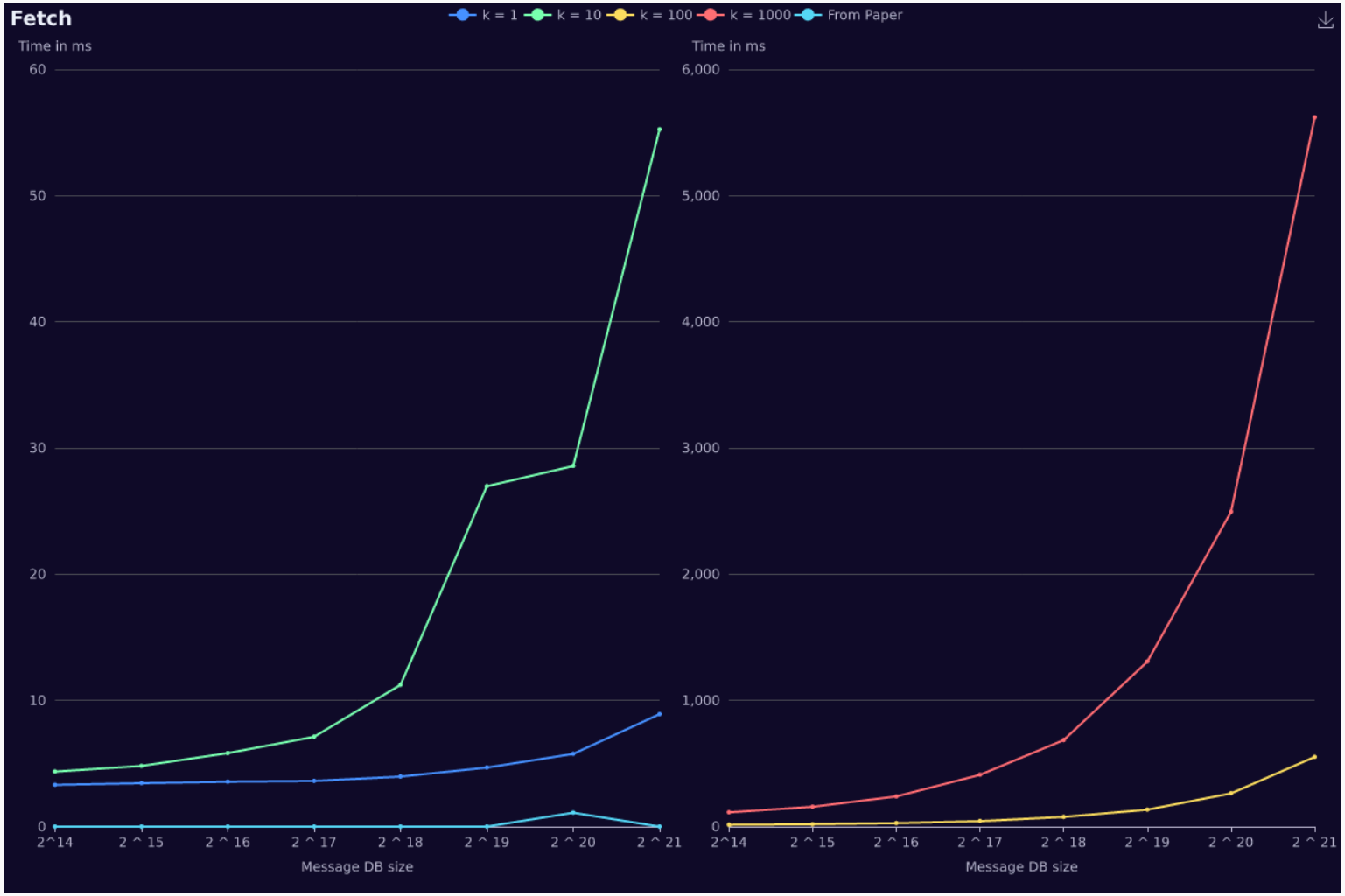
My Approach


Server / Enclave
TLS
Client
Fetch on Interval
SPARTA
dev 1 queue
Proxy
Send
Dev 1
Send
dev 2 Queue
Dev 2

Implementation Details

- No enclave (lack of hardware)
- 2.3k lines of rust
- GRPC as messaging protocol
- Facebook ORAM Implementation
- O(N log(N)) Implementation for UserStore OMAP
- ed25519 Signatures to determine Fetch authenticity
- Proxy has queue per device to hold older messages

Results


Fetch
k = 1
k = 10
k = 100
k = 1000
From Paper
Time in ms
60
50
40
30
20
10
0
2^14
2 ^ 15
2 ^ 16
2 ^ 17
2 ^ 18
2 ^ 19
2 ^ 20
2 ^ 21
Message DB size
Time in ms
6,000
5,000
4,000
3,000
2,000
1,000
0
2^14
2 ^ 15
2 ^ 16
2 ^ 17
2 ^ 18
2 ^ 19
2 ^ 20
2 ^ 21
Message DB size