Precursor: A Fast, Client-Centric and Trusted Key-Value Store using Intel SGX and RDMA

Middleware 2021

Ines Messadi, Shiva Neumann, Nico Weichbrodt, Lennart Almstedt, Mohammad Mahhouk, and Rüdiger Kapitza

TU Braunschweig, Germany
Key-value Stores in the Cloud

- Key-value stores are core of large-scale services
  → Low latency & high request rate are key

- When outsourced to the cloud
  - User data is exposed to malicious attacks
  → Concerns about privacy & integrity
Key-value Stores in the Cloud

- Key-value stores are core of large-scale services
  → **Low latency & high request rate** are key

- When outsourced to the cloud
  - User data is exposed to malicious attacks
  → Concerns about **privacy & integrity**
Key-value Stores in the Cloud

- Key-value stores are core of large-scale services → **Low latency & high request rate** are key

- When outsourced to the cloud
  - User data is exposed to malicious attacks
  → Concerns about **privacy & integrity**

Improvements with trusted execution environments such as **Intel Software Guard Extensions (Intel SGX)**
Research vs. Industry

- **Industry**
  - Redis, Memcached...
  - → Lack of basic security guarantees, e.g. plaintext key-value items

- **Research**
  - Concerto [Arasu et al., SIGMOD’17], ShieldStore [Kim et al., Eurosys’19]
  - → Secure but intensive computations

How to reduce the overhead of intensive computations?
Intel SGX Model

- Extension of the x86 instruction set
- Applications have secure compartments → **Enclave**
- Code & data reside in **Enclave Page Cache (EPC)**
- Confidentiality and integrity protected
- Restriction of **systems calls and I/O operations**
Intel SGX Model

- Extension of the x86 instruction set
- Applications have secure compartments → **Enclave**
- Code & data reside in **Enclave Page Cache (EPC)**
- Confidentiality and integrity protected
- Restriction of **systems calls and I/O operations**

- SGX-based key-value stores
  - Library OS solutions: **GRAPHENE-SGX** [Tsai et al., ATC’17], ..
  - Tailored solutions: **SHIELDSTORE** [Kim et al., Eurosys’19], **SPEICHER** [Bailleu et al., FAST’19]
Intel SGX Architectural Limitations

1. Limited EPC memory
   - Overhead up to $\times 1000$ (SCONE \cite{arnautov_osdi_2016})
   → Cannot protect the full state using the EPC memory

2. System call restriction & enclave transitions
   → Performance loss

3. DMA directly into the enclave are not allowed
   → Large copy overhead
Intel SGX Architectural Limitations

1. Limited EPC memory
   - Overhead up to $\times 1000$ (SCONE [Arnaudov et al., OSDI’16])
   → Cannot protect the full state using the EPC memory

2. System call restriction & enclave transitions
   → Performance loss

3. DMA directly into the enclave are not allowed
   → Large copy overhead

- Data copy and encryption inside the enclave for each operation
- Extensive server-side computation → CPU bottlenecks
Data Center Technology: RDMA

- Often employed in data centers
- Zero-copy & kernel bypassing communication
- Applications register memory with RDMA NIC

→ 1-3 $\mu$s latency and 10-200 Gb/sec bandwidth

---

What do we gain from combining both technologies?
How to combine them efficiently?
The Cost of Cryptographic Operations

- Comparison
  - A *server-encryption* approach
  - RDMA bandwidth

- Experimental setup
  - Intel Xeon E3-2176G
    - (6 cores, 12 hyperthreading)
  - 40 Gbit/s link
  - One-side RDMA WRITE using Perftest

→ 36% less throughput
The Cost of Cryptographic Operations

- Comparison
  - A server-encryption approach
  - RDMA bandwidth
- Experimental setup
  - Intel Xeon E3-2176G
    (6 cores, 12 hyperthreading)
  - 40 Gbit/s link
  - One-side RDMA WRITE using Perftest
  - 36% less throughput

Our approach: Client-side encryption to alleviate CPU bottlenecks
**Precursor Approach**

- Reduces server-side cryptographic load
  - Scalability: offloading cryptographic operations to the client-side
**Precursor Approach**

- Reduces server-side cryptographic load
  - Scalability: offloading cryptographic operations to the client-side
- Mitigates SGX constraints
  - Copy overhead: payload data never enters the enclave
**Precursor Approach**

- Reduces server-side cryptographic load
  - Scalability: offloading cryptographic operations to the client-side

- Mitigates SGX constraints
  - Copy overhead: payload data never enters the enclave

- Integrity preserved using one time **per-operation key**
  - Security: Forward secrecy and rollback attacks detection
Reduces server-side cryptographic load

Scalability: offloading cryptographic operations to the client-side

Mitigates SGX constraints

Copy overhead: payload data never enters the enclave

Integrity preserved using one time per-operation key

Security: Forward secrecy and rollback attacks detection

Use of data center network technology

Performance: High bandwidth and low latency
Experimental Setup

- **Questions**
  - How does PRECURSOR compare to existing SGX-based key value stores?
  - What is the impact of offloading on the performance?

- **Workload:** Yahoo! Cloud Serving Benchmark (YCSB) [Cooper et al., SoCC’10]
- **Server**
  - Intel Xeon E-2176G CPU (3.70 GHz, 6 cores, 12 hyper-threads)
- **Client:** 6 × machines
- **Link:** 40 Gbps RoCE NIC
- **Comparison:**
  - Shieldstore [Kim et al., Eurosys’19]
  - PRECURSOR variant using server-encryption
Evaluation

**Precursor** scales with the number of increasing clients

**Precursor** has 5.9-8.5 × higher throughput than **ShieldStore**

**Precursor** has 29%-40% higher throughput than *server-encryption* scheme

Average of 25 μs latency
**Precursor Take-Home Message**

**Precursor:** A Fast and Secure Key-Value Store

- **Properties**
  - Intel SGX to protect security-sensitive data
  - RDMA to achieve high-performance with low-latency
  - Client-side computation

- **Lessons learned**
  - Optimizing for leveraging RDMA improves the performance
  - Optimizing for CPU utilization is key for key-value stores

→ Paper: more results and technical details
Payload encryption and transport encryption separately

RDMA one-sided write in pre-allocated buffer in the server memory

Security metadata in the enclave while payload remains untrusted

The enclave stores the hash table with the security metadata and the pointers to the respective payload data
One-time keys for the payload is robust and preserves forward secrecy
MAC verification ensures integrity and rollback attacks detection
No re-encryptions once a client is excluded from accessing the service
**Question:** what is the impact of varying value sizes?

→ *server encryption* decreases the throughput with an average of 49% for a read-only and 27% for a update-mostly workload.
**Evaluation: Tail Latency**

![CDF graph showing tail latency comparison between ShieldStore, Precursor, and Precursor with EPC paging.](image)

**Question:** how does the tail latency perform?

→ **PRECURSOR** has lower `GET()` tail latencies

→ Latency steady until 95% at 8 µs

→ EPC impact is apparent from 95%
Evaluation: Latency Analysis

Question: what is the network impact vs. security protection technique?

→ **Precursor** has faster server processing that keeps steady with increasing payload size.
Conclusion

Challenge: How to leverage SGX for securing key-value stores and how to secure applications that utilize RDMA?

- **PRECURSOR**: a key-value store with strong confidentiality & integrity
  - Lowers the server-load to benefit from RDMA
  - Reduces the copy overhead and keeps a small TCB
  - Achieves high throughput than existing SGX-based key-value stores

Questions?
messadi@ibr.cs.tu-bs.de