Elasticclave: An Efficient Memory Model for Enclaves

Jason Zhijingcheng Yu, Shweta Shinde, Trevor E. Carlson, Prateek Saxena

National University of Singapore

ETH Zürich
Spatial Isolation in Intel SGX
Spatial Isolation in Intel SGX
Spatial Isolation in Intel SGX

Applications

Firmware and OS kernel
Spatial Isolation in Intel SGX

Applications

Firmware and OS kernel
Spatial Isolation in Intel SGX

Applications

Firmware and OS kernel

Hardware
Spatial Isolation in Intel SGX

Applications

Firmware and OS kernel

Hardware

Physical memory
Spatial Isolation in Intel SGX

Applications

Firmware and OS kernel

Hardware

Enclave memory

Physical memory
Spatial Isolation in Intel SGX

Applications
Firmware and OS kernel
Hardware

Enclave memory
Physical memory
Spatial Isolation in Intel SGX

Applications
Firmware and OS kernel
Hardware

Enclave memory
Physical memory
Spatial Isolation in Intel SGX

- Applications
- Firmware and OS kernel
- Hardware

Enclave memory

Physical memory
Spatial Isolation in Intel SGX

Applications

Firmware and OS kernel

Hardware

Enclave memory

Public memory

Physical memory
Spatial Isolation in Intel SGX

Applications

Firmware and OS kernel

Hardware

AMD SEV
Sanctum
Keystone

Physical memory

Enclave memory

Public memory
Problems due to Spatial Isolation

Limited in data sharing expressiveness
Problems due to Spatial Isolation

Limited in data sharing expressiveness
Problems due to Spatial Isolation

Limited in data sharing expressiveness

Mutual distrust

apache

---

python
Problems due to Spatial Isolation

Limited in data sharing expressiveness

apache enclave memory

python enclave memory
Problems due to Spatial Isolation

Limited in data sharing expressiveness

apache enclave memory

python enclave memory

apache

python
Problems due to Spatial Isolation

Limited in data sharing expressiveness

apache enclave memory

python enclave memory

Public memory
Problems due to Spatial Isolation

Limited in data sharing expressiveness

apache enclave memory

public memory

python enclave memory

Untrusted software

Public memory
Problems due to Spatial Isolation

Limited in data sharing expressiveness

Untrusted software

Public memory

encrypt

python enclave memory

python

decrypt

apache enclave memory

apache
Problems due to Spatial Isolation

Limited in data sharing expressiveness

apache enclave memory

encrypt

python enclave memory

decrypt

2 extra copies + encryption/decryption

Untrusted software

Public memory
Problems due to Spatial Isolation

Limited in data sharing expressiveness

Client-server

2 extra copies + encryption/decryption

Similar in other patterns:
- Producer-consumer
- Proxy
Huge Overhead of Spatial Isolation
Contributions
Contributions

Data sharing expressiveness

Security
Contributions

Data sharing expressiveness

Spatial isolation

Security
Contributions

Traditional data sharing mechanisms

Data sharing expressiveness

Spatial isolation

Security
Contributions

Elasticclave

Traditional data sharing mechanisms

Data sharing expressiveness

Spatial isolation

Security
Contributions

- Spatialisolation
- Elasticlave
- 1-2 orders of magnitude performance improvement

Data sharing expressiveness

Traditional data sharing mechanisms

Spatial isolation

Security
Threat Model

- Malicious OS
Threat Model

- Malicious OS
- Mutually distrusting applications (compromised during runtime)
Threat Model

- Malicious OS
- Mutually distrusting applications (compromised during runtime)
- DoS attacks are out of scope
Idea: Temporal Isolation

Spatial isolation: memory region is either always private or always public

Temporal isolation: different enclaves may access memory region at different times
Idea: Temporal Isolation

Spatial isolation: memory region is either always private or always public.

Temporal isolation: different enclaves may access memory region at different times.

apache

Shared memory region

python
Idea: Temporal Isolation

Spatial isolation: memory region is either always private or always public

Temporal isolation: different enclaves may access memory region at different times
Idea: Temporal Isolation

Spatial isolation: memory region is either always private or always public

Temporal isolation: different enclaves may access memory region at different times
Idea: Temporal Isolation

Spatial isolation: memory region is either always private or always public

Temporal isolation: different enclaves may access memory region at different times

No extra copies or encryption/decryption
Temporal Isolation Challenge I
Temporal Isolation Challenge I

Each memory region has exactly one enclave as its owner
Each memory region has exactly one enclave as its owner
Only owner can access the memory region
Temporal Isolation Challenge I

Each memory region has exactly one enclave as its owner
Only owner can access the memory region
Owner can pass ownership to others
Each memory region has exactly one enclave as its owner
Only owner can access the memory region
Owner can pass ownership to others
Temporal Isolation Challenge I

Each memory region has exactly one enclave as its owner.
Only owner can access the memory region.
Owner can pass ownership to others.
Each memory region has exactly one enclave as its owner
Only owner can access the memory region
Owner can pass ownership to others

```
python

owns

writes

apache

```
Temporal Isolation Challenge I

Each memory region has exactly one enclave as its owner
Only owner can access the memory region
Owner can pass ownership to others
Temporal Isolation Challenge I

Each memory region has exactly one enclave as its owner
Only owner can access the memory region
Owner can pass ownership to others

Problem: Sharer has no control over how data is accessed after sharing
How does Elasticlave solve this?
Elasticlave Design: Maximum Permissions
Transferring ownership fixed owner
Transferring ownership fixed owner

Owner sets *maximum permissions* for other enclaves
Elasticlave Design: Maximum Permissions

Transferring ownership fixed owner

Owner sets *maximum permissions* for other enclaves

Maximum permissions limit how other enclaves access the memory region
Transferring ownership fixed owner

Owner sets *maximum permissions* for other enclaves

Maximum permissions limit how other enclaves access the memory region
Temporal Isolation Challenge II
Temporal Isolation Challenge II

apache

owns

max: RX

python
Temporal Isolation Challenge II
Temporal Isolation Challenge II

apache

executes

owns

python

executes
Problem: Accessors cannot enforce their own memory protection permissions
Elasticlave Design: Effective Permissions
Elasticlave Design: Effective Permissions
Elasticlave Design: Effective Permissions

apache
owns

max: $RX$
effective: $\emptyset$

python
Elasticlave Design: Effective Permissions

Each enclave can request hardware to change *its own* effective permissions dynamically.
Each enclave can request hardware to change its own effective permissions dynamically.
Connecting Maximum and Effective Permissions
For any (enclave, memory region) pair,
effective permissions $\leq$ maximum permissions
For any (enclave, memory region) pair, effective permissions ≤ maximum permissions.
Connecting Maximum and Effective Permissions

For any (enclave, memory region) pair, effective permissions $\leq$ maximum permissions

Exception
Temporal Isolation Challenge III

apache

owns

own

python
Temporal Isolation Challenge III

apache

max: RWX
effective: R

owns

document

max: R

effective: R

reads

python
Temporal Isolation Challenge III

apache

max: RWX
effective: R

owns

max:
R

effective:
R

reads

python
Temporal Isolation Challenge III

apache

max: RWX
effective: RW
writes

owns

max: R
effective: R
reads

python
Temporal Isolation Challenge III

apache
  max: RWX
  effective: RW
  writes
owns

TOCTTOU
  max: R
  effective: R

python
  max: R
  effective: R
  reads
Temporal Isolation Challenge III

Problem: No mechanism for synchronization
Elasticlave: Lock Bit

apache

max: RWX
effective: RW
writes

owns

Python

max: R
effective: R
reads
Elasticlave: Lock Bit

apache max: RWX
  effective: RW
  writes

owns

max: RL
  effective: RL
  reads

python
When held in effective permissions: exclusive access guaranteed
Elasticlave: Lock Bit

When held in effective permissions: exclusive access guaranteed

When held in ineffective permissions: exclusive access guaranteed

apache

owns

max: RL
effective: RL
reads

python
Elasticlave Summary

Three elements:

- Maximum permissions
Elasticlave Summary

Three elements:

- Maximum permissions
- Effective permissions (≤ maximum permissions)
Elasticlave Summary

Three elements:

- Maximum permissions
- Effective permissions (≤ maximum permissions)
- Synchronization: lock bit
Implementation
Implementation
Implementation

M-mode

Security monitor

CPU

Rocket Core
Implementation

U-mode
Normal applications

S-mode
OS kernel

M-mode
Security monitor

CPU
Rocket Core
Implementation

U-mode
Normal applications

S-mode
OS kernel

M-mode
Security monitor

CPU
Rocket Core
Implementation

U-mode: Normal applications

S-mode: OS kernel

M-mode: Security monitor

CPU: Rocket Core

7 Elasticlave instructions: create, share, change, ...

Apache, Python
Implementation

U-mode: Normal applications

S-mode: OS kernel

M-mode: Security monitor

CPU: Rocket Core

PMP entries

7 Elasticlave instructions: create, share, change, ...

Apache

Python
Implementation

U-mode
- Normal applications
- apache
- python

S-mode
- OS kernel

M-mode
- Security monitor

CPU
- Rocket Core

Elasticlave instructions:
- create, share, change, ...

1 memory region ↔ 1 PMP entry
Implementation

U-mode: Normal applications

S-mode: OS kernel

M-mode: Security monitor

CPU: Rocket Core

1 memory region ↔ 1 PMP entry
Implementation

U-mode
- Normal applications

S-mode
- OS kernel

M-mode
- Security monitor

CPU
- Rocket Core
- PMP entries

1 memory region ↔ 1 PMP entry

apache
python

max: RL
effective: RL

Physical memory
Implementation

U-mode
Normal applications

S-mode
OS kernel

M-mode
Security monitor

CPU
Rocket Core

Physical memory

max: \textit{RL}

effective: \textit{RL}

1 memory region ↔ 1 PMP entry

\begin{itemize}
\item Rocket Core
\item PMP entries
\end{itemize}

\textit{CPU}

\textit{OS kernel}

\textit{Normal applications}

\textit{Security monitor}

\textit{Physical memory}

\textit{max: RL}

\textit{Effective: RL}

\textit{1 memory region ↔ 1 PMP entry}
Evaluation question: Performance of Elasticlave compared to spatial isolation
Evaluations

Evaluation question: Performance of Elasticlave compared to spatial isolation

Benchmarks:
Evaluations

Evaluation question: Performance of Elasticlave compared to spatial isolation

Benchmarks:

- Handcrafted microbenchmarks for data sharing patterns
Evaluations

Evaluation question: Performance of Elasticlave compared to spatial isolation

Benchmarks:

- Handcrafted microbenchmarks for data sharing patterns
- Standard benchmarks: IOZone (I/O), SPLASH-2 (sharing)
Evaluation question: Performance of Elasticlave compared to spatial isolation

Benchmarks:
- Handcrafted microbenchmarks for data sharing patterns
- Standard benchmarks: IOZone (I/O), SPLASH-2 (sharing)

Baselines:
Evaluations

Evaluation question: Performance of Elasticlave compared to spatial isolation

Benchmarks:
- Handcrafted microbenchmarks for data sharing patterns
- Standard benchmarks: IOZone (I/O), SPLASH-2 (sharing)

Baselines:
- Spatial isolation (Keystone)
Evaluations

**Evaluation question:** Performance of Elasticlave compared to spatial isolation

**Benchmarks:**
- Handcrafted microbenchmarks for data sharing patterns
- Standard benchmarks: IOZone (I/O), SPLASH-2 (sharing)

**Baselines:**
- Spatial isolation (Keystone)
- Native Linux execution
Evaluation question: Performance of Elasticclave compared to spatial isolation

Benchmarks:
- Handcrafted microbenchmarks for data sharing patterns
- Standard benchmarks: IOZone (I/O), SPLASH-2 (sharing)

Baselines:
- Spatial isolation (Keystone)
- Native Linux execution

Run on cycle-accurate FPGA-accelerated simulator (FireSim)
SPLASH-2 (sharing-intensive)

Two enclaves performing parallel computation
Two enclaves performing parallel computation

**Elasticclave**: memory region accessible to both enclaves
SPLASH-2 (sharing-intensive)

Two enclaves performing parallel computation

**Elasticlave**: memory region accessible to both enclaves

**Spatial isolation**: passing data through extra copies and encryption/decryption
Two enclaves performing parallel computation

**Elasticclave**: memory region accessible to both enclaves

**Spatial isolation**: passing data through extra copies and encryption/decryption

**Native**: two threads
Evaluation Results on SPLASH-2 (sharing-intensive)
Evaluation Results on SPLASH-2 (sharing-intensive)

100x performance improvement
Evaluation Results on SPLASH-2 (sharing-intensive)

Comparable to native Linux
Conclusions

- Elasticlave: maximum permissions, effective permissions, lock bit
Conclusions

- Elasticlave: maximum permissions, effective permissions, lock bit
- Prototype implementation on RISC-V
Conclusions

- Elasticlave: maximum permissions, effective permissions, lock bit
- Prototype implementation on RISC-V (using PMP)
Conclusions

- Elasticlave: maximum permissions, effective permissions, lock bit
- Prototype implementation on RISC-V (using PMP)
- Evaluation: 1-2 orders of magnitude performance improvement
Conclusions

- Elasticlave: maximum permissions, effective permissions, lock bit
- Prototype implementation on RISC-V (using PMP)
- Evaluation: 1-2 orders of magnitude performance improvement

Artifact available: https://github.com/jasonyu1996/elasticlave
Conclusions

- Elasticlave: maximum permissions, effective permissions, lock bit
- Prototype implementation on RISC-V (using PMP)
- Evaluation: 1-2 orders of magnitude performance improvement

Artifact available: https://github.com/jasonyu1996/elasticlave

Thanks for listening!