ShieldBox

Secure Middleboxes using Shielded Execution

Bohdan Trach, Alfred Krohmer, Franz Gregor, Sergei Arnautov, Pramod Bhatotia, Christof Fetzer
Middleboxes in the Cloud

Significant costs:
- Deployment
- Maintenance
- Management
Security Issues

- Cheap computation resources 😊
- NFV advances 😊
- Low trust environment 😞

- Observe private data
- Extract encryption keys
- Learn configuration

Cloud
State-Of-the-Art Systems

Blindbox [SIGCOMM’15], Embark [NSDI’16]:
- High performance overhead 😞
- Offer limited functionality 😞
Problem Statement

How to securely outsource middleboxes to the untrusted cloud without sacrificing performance while supporting a wide range of NFs?
ShieldBox:
● Middlebox framework with shielded execution

Uses Intel SGX
Design Goals

- **Security** - strong confidentiality and integrity guarantees
- **Performance** - near-native throughput and latency
- **Generality** - supports a wide range of NFs
- **Transparency** - portable, configurable, and verifiable architecture
Outline

● Motivation
● Design
● Evaluation
● Summary
System Overview

<table>
<thead>
<tr>
<th>Security</th>
<th>Performance</th>
<th>Usability</th>
</tr>
</thead>
</table>

ShieldBox

Organization

Endpoint
System Overview

<table>
<thead>
<tr>
<th>✔</th>
<th>Security</th>
<th>Intel SGX</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Performance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Usability</td>
<td></td>
</tr>
</tbody>
</table>
Background: Intel SGX

Intel SGX allows creation and management of enclaves.
Background: Intel SGX

Intel SGX allows creation and management of enclaves.
Background: Intel SGX

Intel SGX allows creation and management of enclaves.
Background: Intel SGX

Intel SGX allows creation and management of enclaves.
Background: Intel SGX

Intel SGX allows creation and management of **enclaves**.

![Diagram showing the architecture of Intel SGX](image-url)
Background: Intel SGX

Intel SGX allows creation and management of **enclaves**.

- Restrictions on allowed instructions:
  - syscall
  - rdtsc
Background: Intel SGX

Intel SGX allows creation and management of enclaves.

- Restrictions on allowed instructions:
  - syscall
  - rdtsc

- High overheads for:
  - Secure memory paging
  - Enclave entry/exit
System Overview

<table>
<thead>
<tr>
<th>Security</th>
<th>Intel SGX &amp; SCONE [OSDI’16]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td></td>
</tr>
<tr>
<td>Usability</td>
<td></td>
</tr>
</tbody>
</table>
System Overview

<table>
<thead>
<tr>
<th>✔ Security</th>
<th>Intel SGX &amp; SCONE [OSDI’16]</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔ Performance</td>
<td>DPDK</td>
</tr>
<tr>
<td>Usability</td>
<td></td>
</tr>
</tbody>
</table>

Organization → ShieldBox

Intel SGX Enclave

DPDK

SCONE

Endpoint
System Overview

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>Intel SGX &amp; SCONE [OSDI’16]</td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>DPDK</td>
<td></td>
</tr>
<tr>
<td>Usability</td>
<td>Click Router [TOCS’00]</td>
<td></td>
</tr>
</tbody>
</table>
Partitioning ShieldBox
Partitioning ShieldBox

1. DPDK outside - ecalls
   - High overhead 😞
   - Reengineering 😞
   - Reduced TCB ☺
Partitioning ShieldBox

1. DPDK outside - ecalls
   - High overhead 😞
   - Reengineering 😞
   - Reduced TCB 😊

2. DPDK outside - sibling core
   - Overhead in some cases 😞
   - Reengineering 😞
   - Reduced TCB 😊
**Partitioning ShieldBox**

1. DPDK outside - ecalls
   - High overhead 😞
   - Reengineering 😞
   - Reduced TCB ☺

2. DPDK outside - sibling core
   - Overhead in some cases 😞
   - Reengineering 😞
   - Reduced TCB ☺

3. DPDK inside enclave
   - Low overhead ☺
   - No reengineering ☺
   - Increased TCB 😞
Partitioning ShieldBox - DPDK

- NIC can’t deliver packets directly to enclave.
Partitioning ShieldBox - DPDK

- NIC can’t deliver packets directly to enclave:
  - Allocate hugepage memory outside
  - Packets and mbufs delivered to hugepages
ShieldBox Features

- **Security**
  - Iago Attack Protection
  - New Elements
  - Remote Attestation and Configuration System

- **Performance**
  - On-NIC Time Source
  - Optimizations over standard Click

- **Features**
  - Middlebox State Persistence
  - Network Function Chaining
  - New Elements
System Workflow with Remote Attestation

Configuration and Attestation Service

Network Operator

ShieldBox / Local Attestation Service

Middlebox Developer

Middlebox Image Repository

Upload middlebox images
System Workflow with Remote Attestation

- **Network Operator**
- **Configuration and Attestation Service**
  - Launch the CAS service on a trusted host
- **Middlebox Developer**
- **Middlebox Image Repository**
- **ShieldBox / Local Attestation Service**
System Workflow with Remote Attestation

- Network Operator
- Middlebox Developer
- Middlebox Image Repository
- ShieldBox / Local Attestation Service
- Install LAS service on a ShieldBox host
- Configuration and Attestation Service
System Workflow with Remote Attestation

Configuration and Attestation Service

Network Operator

ShieldBox / Local Attestation Service

Install ShieldBox from the repository

Middlebox Developer

Middlebox Image Repository
System Workflow with Remote Attestation

- **Network Operator**
  - Provides configuration and secrets to CAS

- **Configuration and Attestation Service**

- **ShieldBox / Local Attestation Service**

- **Middlebox Developer**

- **Middlebox Image Repository**
System Workflow with Remote Attestation

1. **Network Operator** launches **ShieldBox**.
2. **Configuration and Attestation Service** performs remote attestation and configuration.
3. **Middlebox Developer** retrieves the **Middlebox Image Repository**.

- **Launch ShieldBox**
- **Perform remote attestation, configuration**
New Elements

- **ToEnclave:**
  - Copies packet data into enclave
New Elements

● ToEnclave:
  ○ Copies packet data into enclave

● Seal:
  ○ Encrypts packet using AES-GCM
New Elements

- **ToEnclave:**
  - Copies packet data into enclave

- **Seal:**
  - Encrypts packet using AES-GCM

- **Unseal:**
  - Decrypts an AES-GCM encrypted packet
New Elements

- **ToEnclave:**
  - Copies packet data into enclave

- **Seal:**
  - Encrypts packet using AES-GCM

- **Unseal:**
  - Decrypts an AES-GCM encrypted packet

- **HyperScan, DPDKRing, StateFile:**
  - See paper!
Iago Attack Protection
Iago Attack Protection
Iago Attack Protection

Hugepage Memory

Packet Data
(0x7F..0000-0x7F..FFFF)

Packet data transfer

NIC
Iago Attack Protection

**Hugepage Memory**

- Packet Data
  - (0x7F..0000-0x7F..FFFF)
- DPDK mbufs
  - 0x7F..A000
  - 0x7F..B000
  - 0x7F..C000

**NIC**

Packet data transfer
Iago Attack Protection

Enclave

Secret Data
(0x20..0000-0x20..FFFF)

Hugepage Memory

Packet Data
(0x7F..0000-0x7F..FFFF)

DPDK mbufs

0x7F..A000

0x7F..B000

0x7F..C000

0x20..D000

NIC

Packet data
transfer
**Iago Attack Protection**

**Enclave**

- Secret Data (0x20..0000-0x20..FFFF)

**Hugepage Memory**

- Packet Data (0x7F..0000-0x7F..FFFF)

**NIC**

- Packet data transfer

---

Click Packets:
- 0x7F..A000
- 0x7F..B000
- 0x7F..C000
- 0x20..D000

Descriptor creation:
- DPDK mbufs
  - 0x7F..A000
  - 0x7F..B000
  - 0x7F..C000
  - 0x20..D000
Iago Attack Protection

Enclave
- Secret Data (0x20..0000-0x20..FFFF)

Hugepage Memory
- Packet Data (0x7F..0000-0x7F..FFFF)
  - DPDK mbufs
    - 0x7F..A000
    - 0x7F..B000
    - 0x7F..C000
    - 0x20..D000

NIC
- Packet data transfer
Iago Attack Protection

Enclave

Secret Data
(0x20..0000-0x20..FFFF)

Click Packets

- ✔ 0x7F..A000
- ✔ 0x7F..B000
- ✔ 0x7F..C000

Hugepage Memory

Packet Data
(0x7F..0000-0x7F..FFFF)

DPDK mbufs

- 0x7F..A000
- 0x7F..B000
- 0x7F..C000
- 0x20..D000

Packet data transfer

NIC

Mbuf check

✔ 0x7F..A000
✔ 0x7F..B000
✔ 0x7F..C000
Iago Attack Protection

**Enclave**

- Secret Data (0x20..0000-0x20..FFFF)
- Click Packets
  - ✔ 0x7F..A000
  - ✔ 0x7F..B000
  - ✔ 0x7F..C000
  - ✗ Discarded

**Hugepage Memory**

- Packet Data (0x7F..0000-0x7F..FFFF)
- DPDK mbufs
  - 0x7F..A000
  - 0x7F..B000
  - 0x7F..C000
  - 0x20..D000

**NIC**

- Packet data transfer

- Mbuf check
On-NIC Time Source

- `clock_gettime`
  - Hot sthread: reduce performance
On-NIC Time Source

- `clock_gettime`
  - Hot sthread: reduce performance
  - Cold sthread: huge overhead
On-NIC Time Source

- **clock_gettime**
  - Hot sthread: reduce performance
  - Cold sthread: huge overhead

- **rdtsc**
  - Causes enclave exit
  - Performance loss due to TLB flush
On-NIC Time Source

- **clock_gettime**
  - Hot sthread: reduce performance
  - Cold sthread: huge overhead

- **rdtsc**
  - Causes enclave exit
  - Performance loss due to TLB flush

- **On-NIC Timer**
  - Acceptable performance
On-NIC Time Source

- **clock_gettime**
  - Hot sthread: reduce performance
  - Cold sthread: huge overhead

- **rdtsc**
  - Causes enclave exit
  - Performance loss due to TLB flush

- **On-NIC Timer**
  - Acceptable performance

All of these time sources are untrusted.
Outline

- Motivation
- Design
- Evaluation
- Summary
Evaluation

- What is the throughput and latency of our system?
- What is the influence of ToEnclave element on the performance?
- Other questions: see in the paper.
Throughput: Router Use Case
Throughput: Router Use Case

Packet Size, bytes

Throughput, Gb/s

Native  ShieldBox  Others
Throughput: Router Use Case

The higher the better
Throughput: Router Use Case

Line rate at most common packet size

The higher the better
Latency: Router Use Case
Latency: Router Use Case

![Latency vs Packet Size Graph]

- **Native**
- **ShieldBox w/o opt.**
- **ShieldBox w. opt**
- **ShieldBox+NIC timer**

---

- **Packet Size, bytes**: 64, 128, 256, 512, 1024, 1500
- **Latency, μsec**: 0 to 60
Latency: Router Use Case

The lower the better
Optimizations bring latency to native level
ToEnclave Influence: EtherMirror

- Cheap NF → worst-case example
ToEnclave Influence: EtherMirror

![Graph showing throughput vs packet size for different configurations: Native, Native + ToEnc, ShieldBox, ShieldBox + ToEnc. The x-axis represents packet size in bytes (64, 128, 256, 512, 1024, 1500), and the y-axis represents throughput in Gb/s (0 to 40). The graph illustrates the performance comparison across these configurations.]
ToEnclave Influence: EtherMirror

The higher the better.
ToEnclave Influence: EtherMirror

- ~15% throughput reduction due to the extra memory copy.

The higher the better.
Outline

- Motivation
- Design
- Evaluation
- Summary
Summary

- Cloud:
  - Abundant computational resources 😊
  - Limited trust to platform 😞
Summary

- Cloud:
  - Abundant computational resources 😊
  - Limited trust to platform 😞

- TEEs allow construction of middleboxes in the cloud:
  - Achieve end-to-end trust
  - Flexible frameworks for NF construction available
Summary

● High performance:
  ○ Line rate with typical Network Functions by using DPDK
  ○ Minimal overhead from ToEnclave element
Summary

● **High performance:**
  ○ Line rate with typical Network Functions by using DPDK
  ○ Minimal overhead from ToEnclave element

● **Secure:**
  ○ End-to-end trusted NF system with Intel SGX and SCONE
  ○ Enables use of modern cryptography
Summary

- **High performance:**
  - Line rate with typical Network Functions by using DPDK
  - Minimal overhead from ToEnclave element

- **Secure:**
  - End-to-end trusted NF system with Intel SGX and SCONE
  - Enables use of modern cryptography

- **Practical:**
  - Allows construction of wide range of Network Functions
  - Easy management using Dockerfiles from SCONE remote configuration
Summary

- **High performance:**
  - Line rate with typical Network Functions by using DPDK
  - Minimal overhead from ToEnclave element
- **Secure:**
  - End-to-end trusted NF system with Intel SGX and SCONE
  - Enables use of modern cryptography
- **Practical:**
  - Allows construction of wide range of Network Functions
  - Easy management using Dockerfiles from SCONE remote configuration

Thank You!
Funding

This project was funded by the European Union’s Horizon 2020 program under grant agreements No. 645011 (SERECA), No. 690111 (SecureCloud), and No. 690588 (Selis)