Sanctuary: ARMing TrustZone with User-space Enclaves

Ferdinand Brasser, David Gens, Patrick Jauernig, Ahmad-Reza Sadeghi, Emmanuel Stapf

Technische Universität Darmstadt
Research Problem

How can we construct an ecosystem of mutually distrusted enclaves on mobile devices?
Commodity OS

- App
- App
- Sensitive App

Operating System

- Trusted
- Untrusted
- Compromised
Commodity OS

- OS large attack surface
- Insufficient sensitive app protection
Commodity OS

- OS large attack surface
- Insufficient sensitive app protection
Commodity OS

- OS large attack surface
- Insufficient sensitive app protection

ARM TrustZone
Commodity OS

- OS large attack surface
- Insufficient sensitive app protection

ARM TrustZone

Normal World | Secure World
---|---

Trusted | Untrusted | Compromised
Commodity OS

- OS large attack surface
- Insufficient sensitive app protection

ARM TrustZone

- Normal World
  - App
  - App
- Secure World
  - OS

Trusted
Untrusted
Compromised
Commodity OS

- OS large attack surface
- Insufficient sensitive app protection

ARM TrustZone

Normal World
- App
- Sensitive App

Secure World
- Sensitive App
- Sensitive App

Trusted OS
Commodity OS

- OS large attack surface
- Insufficient sensitive app protection

ARM TrustZone

- Normal World
  - App
  - App
  - OS

- Secure World
  - Trusted OS
  - Sensitive App
  - Sensitive App

Trusted Firmware

- Trusted
- Untrusted
- Compromised
ARM TrustZone
The Solution?

![Diagram of ARM TrustZone](image)

- Normal World
  - App
  - App
  - OS

- Secure World
  - Sensitive App
  - Sensitive App
  - Trusted OS

- Trusted Firmware

Legend:
- Trusted
- Untrusted
- Compromised
ARM TrustZone
The Solution?

- Trusted OS still large attack surface
- High costs to protect apps with TrustZone
- Main problem: Single security domain
ARM TrustZone
The Solution?

- Trusted OS still large attack surface
- High costs to protect apps with TrustZone
- Main problem: Single security domain
ARM TrustZone
The Solution?

Normal World
- App
- App
- OS

Secure World
- Sensitive App
- Sensitive App
- Trusted OS

Trusted Firmware

• Trusted OS still large attack surface
• High costs to protect apps with TrustZone
• Main problem: Single security domain

Sanctuary
Multiple Security Domains

Normal World
- App
- Sensitive App
- Sensitive App
- OS

Secure World
- Vendor Code

Trusted Firmware
ARM TrustZone
The Solution?

Trusted Firmware

- Trusted OS still large attack surface
- High costs to protect apps with TrustZone
- Main problem: Single security domain

Sanctuary
Multiple Security Domains

Trusted Firmware

Vendor Code

- Trusted
- Untrusted
- Compromised
ARM TrustZone
The Solution?

- Trusted OS still large attack surface
- High costs to protect apps with TrustZone
- Main problem: Single security domain

Sanctuary
Multiple Security Domains

- Sensitive apps can remain untrusted
- Make TrustZone protection available to third parties

Trusted Firmware
Challenges

- No new hardware components
- No replacement of existing code bases (e.g. Trusted OS)
- Only minor impact on commodity OS

- Trusted OS still large attack surface
- High costs to protect apps with TrustZone
- Main problem: Single security domain

- Sensitive apps can remain untrusted
- Make TrustZone protection available to third parties
Adversary Model

- Adversary can
  - compromise normal-world software at all privilege levels
  - run malicious sensitive apps

- Adversary cannot
  - compromise TrustZone (trust anchor)
  - perform physical attacks
Related Work
Existing Research Proposals

Software Approaches
Utilizing TrustZone

<table>
<thead>
<tr>
<th>Normal World</th>
<th>Secure World</th>
</tr>
</thead>
<tbody>
<tr>
<td>App</td>
<td>Sensitive App</td>
</tr>
<tr>
<td>App</td>
<td>App</td>
</tr>
</tbody>
</table>
Existing Research Proposals

Software Approaches
Utilizing TrustZone

<table>
<thead>
<tr>
<th>Normal World</th>
<th>Secure World</th>
</tr>
</thead>
<tbody>
<tr>
<td>App</td>
<td>Sensitive App</td>
</tr>
<tr>
<td>App</td>
<td></td>
</tr>
</tbody>
</table>

Improve isolation of sensitive apps without add. HW features among others,

[Ferraiuolo et al., SOSP 2017]
[Sun et al., DSN 2015]
Existing Research Proposals

### Software Approaches Utilizing TrustZone

- **Normal World**
  - App
  - App
- **Secure World**
  - Sensitive App

Improve isolation of sensitive apps without add HW features

among others,

[Ferraiuolo et al., SOSP 2017]
[Sun et al., DSN 2015]

### Hypervisor-based Isolation Utilizing TrustZone

- **App**
- **App**
- **Sensitive App**

Use virtualization to protect sensitive apps. Use TrustZone as trust anchor

among others,

[Hua et al., USENIX 2017]
[Cho et al., USENIX 2016]
Existing Research Proposals

**Software Approaches Utilizing TrustZone**

- Normal World
  - App
  - App
- Secure World
  - Sensitive App

**Hypervisor-based Isolation Utilizing TrustZone**

- App
- App
- Sensitive App
- Hypervisor

**Architectural Modifications**

- App
- Sensitive App
- Cache
- Memory

Improve isolation of sensitive apps without add. HW features

- [Ferraiuolo et al., SOSP 2017]
- [Sun et al., DSN 2015]

Use virtualization to protect sensitive apps. Use TrustZone as trust anchor

- [Hua et al., USENIX 2017]
- [Cho et al., USENIX 2016]

Overcome TrustZone’s shortcomings with own HW

- [Costan et al., USENIX 2016]
- [Evtyushkin et al., MICRO 2014]
Existing Research Proposals - Problems

Software Approaches Utilizing TrustZone
- Normal World
- Secure World
  - App
  - Sensitive App

Hypervisor-based Isolation Utilizing TrustZone
- App
- Sensitive App
- Hypervisor

Architectural Modifications
- App
- Sensitive App
- Cache
- Memory

- Replaces existing code base
- Hypervisor blocked
- New hardware design
Existing Research Proposals - Problems

Software Approaches Utilizing TrustZone

- Normal World
  - App
  - Sensitive App
- Secure World
  - App
  - Sensitive App

Hypervisor-based Isolation Utilizing TrustZone

- App
- Hypervisor
- Sensitive App

Architectural Modifications

- App
- Sensitive App
- Cache
- Memory

Problems

- Replaces existing code base
- Hypervisor blocked
- New hardware design
- Slows down commodity OS
- Additional TCB component
- Additional HW for DMA access control
- OS needs to be suspended
- No support for multi-core environments
- High deployment costs
- Low adoption by industry
Sanctuary provides Multi-Domain Isolation

NW = Normal World
SW = Secure World
Sanctuary provides Multi-Domain Isolation

Secure world forms our trust anchor

NW = Normal World
SW = Secure World
Sanctuary provides Multi-Domain Isolation

Secure world forms our trust anchor

Contains security primitives provided by the device vendor

NW = Normal World
SW = Secure World
Sanctuary provides Multi-Domain Isolation

Every core can access **same** security primitives

NW = Normal World
SW = Secure World
Sanctuary provides Multi-Domain Isolation

Core A
- NW
- SW
- App
- App
- Security Primitives

Core B
- NW
- SW
- NW (Core B)
- Security Primitives

Core C
- NW
- SW
- Security Primitives

Isolate core using TrustZone features

NW = Normal World
SW = Secure World
Sanctuary provides Multi-Domain Isolation

Core A
NW
App
App
SW
Security Primitives

Core B
NW
Sensitive App I
SW
Security Primitives

Core C
NW
SW
Security Primitives

Memory

Isolate core using TrustZone features

NW = Normal World
SW = Secure World
Sanctuary provides Multi-Domain Isolation

**Core A**
- NW: Application
- SW: Security Primitives

**Core B**
- NW: Sensitive App I
- SW: Security Primitives

**Core C**
- NW: Sensitive App II
- SW: Security Primitives

NW = Normal World
SW = Secure World
Sanctuary provides Multi-Domain Isolation

- Only using TrustZone features. No new HW design.
- Vendor can keep existing code in SW.
- No heavy influence on commodity OS.

Challenges revisited:
- NW = Normal World
- SW = Secure World

Memory

Core A
- NW
- SW

App
App

Security Primitives

Core B
- NW
- SW

Sensitive App I
Sensitive App II

Security Primitives

Core C
- NW
- SW

Sensitive App II

Security Primitives

Sanctuary I (Core B)
Sanctuary II (Core C)
Technical Details of Sanctuary
Going Beyond TrustZone ...

NM = Normal Mode
SM = Secure Mode
Going Beyond TrustZone ...

Hardware
- Core A
- Core B

Software

Hardware
- Memory Controller
- NM = Normal Mode
- SM = Secure Mode
- NS = Non-Secure
- NM = 0/1
- SM = 0/1
- NS = 0/1

RAM
- Normal World
- Secure World

NM = Normal Mode
SM = Secure Mode
NS = Non-Secure
Going Beyond TrustZone ...

SA = Sensitive App
TOS = Trusted OS
NW = Normal World
SW = Secure World
NM = Normal Mode
SM = Secure Mode
NS = Non-Secure
Going Beyond TrustZone ...

Hardware

RAM

Normal World Secure World

Memory Controller

Software

Core A

NW

App

OS

Trusted Firmware

SW

TOS

SA

Core B

NW

App

OS

Trusted Firmware

SW

TOS

SA

Software

Core A

NW

App

OS

Trusted Firmware

SW

TOS

SA

Core B

NW

App

OS

Trusted Firmware

SW

TOS

SA

Core A

NW

App

OS

Trusted Firmware

SW

TOS

SA

Core B

NW

App

OS

Trusted Firmware

SW

TOS

SA

Hardware

NM

SM

NS = 0/1 NSAID = A

NS = 0/1 NSAID = B

TOS

SM = Secure Mode

NM = Normal Mode

SA = Sensitive App

TOS = Trusted OS

NW = Normal World

SW = Secure World

NS = Non-Secure

NSAID = Non-Secure Access ID
Going Beyond TrustZone ...

SA = Sensitive App  
TOS = Trusted OS  
NW = Normal World  
SW = Secure World  
NM = Normal Mode  
SM = Secure Mode  
NS = Non-Secure  
NSAID = Non-Secure Access ID
Going Beyond TrustZone …

SA = Sensitive App
TOS = Trusted OS
NW = Normal World
SW = Secure World
NM = Normal Mode
SM = Secure Mode
NS = Non-Secure
NSAID = Non-Secure Access ID

SA = Sensitive App
TOS = Trusted OS
NW = Normal World
SW = Secure World
NM = Normal Mode
SM = Secure Mode
NS = Non-Secure
NSAID = Non-Secure Access ID
Going Beyond TrustZone ...

Core A

Software
- NW
  - App
  - OS
- SW
  - Trusted Firmware

Hardware
- NM
  - SM
    - NS = 0/1
    - NSAID = A
- NM
  - SM
    - NS = 0/1
    - NSAID = B

Memory Controller

RAM
- NW – Core A
- NW – Core B
  - Secure World

Core B

Software
- NW
  - App
  - OS
- SW
  - Trusted Firmware

Hardware
- NM
  - SM
    - NS = 0/1
    - NSAID = A
- NM
  - SM
    - NS = 0/1
    - NSAID = B

Memory Controller

RAM
- NW – Core A
- NW – Core B
  - Secure World

Abbreviations:
- SP = Security Primitives
- SA = Sensitive App
- TOS = Trusted OS
- NW = Normal World
- SW = Secure World
- NM = Normal Mode
- SM = Secure Mode
- NS = Non-Secure
- NSAID = Non-Secure Access ID
Going Beyond TrustZone ...

Core A
- NW: App
- OS: Trusted Firmware

Core B
- NW: App
- OS: Trusted Firmware

Sanctuary (Core C)

Software
- NW
- SW
- App
- SP
- OS

Hardware
- NM
- SM
- NS = 0/1
- NSAID = A
- NS = 0/1
- NSAID = B
- NS = 0/1
- NSAID = C

Memory Controller

RAM
- Normal World
- Sanctuary
- Secure World
Going Beyond TrustZone ...

- **Core A**
  - NW: Application
  - SW: Operating System
  - Trusted Firmware

- **Core B**
  - NW: Application
  - SW: Operating System
  - Trusted Firmware

- **Sanctuary (Core C)**
  - NW: Sensitive App
  - SW: Sanct. Library
  - Trusted Firmware

- **Hardware**
  - NM: Normal Mode
  - SM: Secure Mode
  - NS = 0/1
  - NSAID = A/B/C

- **Memory Controller**
  - Normal World
  - Sanctuary
  - Secure World

**Diagrams**
- OS
- Trusted Firmware
- App
- Memory Controller
- Normal Mode
- Secure Mode
Going Beyond TrustZone ...

Provides mgmt. functionalities. Not part of system TCB

Software:
- Core A: NW App, OS, Trusted Firmware
- Core B: NW App, OS, Trusted Firmware
- Sanctuary (Core C): NW Sensitive App, Sanct. Library, Trusted Firmware

Hardware:
- NM SM: NS = 0/1, NSAID = A, B, C

Memory Controller:
- Normal World
- Sanctuary
- Secure World

RAM
Evaluation of Sanctuary PoC
Security Considerations

✔ Protection from compromised OS
  before, after and during runtime of sensitive app

✔ Protection from malicious sensitive app
  sensitive app isolated from OS and other sensitive apps

✔ Protection from cache-side channel attacks
  flush exclusive caches, exclude sensitive apps from shared caches
Performance Evaluation
Performance Evaluation

Sanctuary Life Cycle
Performance Evaluation

Sanctuary Life Cycle

OS Execution
Performance Evaluation

Sanctuary Life Cycle

OS Execution

Sensitive app execution triggered
Performance Evaluation

Sanctuary Life Cycle

- OS Execution
- Sanctuary Setup
- Sensitive App Execution

Sensitive app execution triggered

OS still runs in parallel on other cores
Performance Evaluation

Sanctuary Life Cycle

- OS Execution
- Sanctuary Setup
- Sensitive App Execution
- Sanctuary Teardown

Sensitive app execution triggered
OS still runs in parallel on other cores
Performance Evaluation

Sanctuary Life Cycle

- OS Execution
- Sanctuary Setup
- Sensitive App Execution
- Sanctuary Teardown
- OS Execution

Sensitive app execution triggered

OS still runs in parallel on other cores
Performance Evaluation

Sanctuary Life Cycle

- **OS Execution**
- **Sanctuary Setup**
- **Sensitive App Execution**
- **Sanctuary Teardown**
- **OS Execution**

**Sensitive app execution triggered**

**OS still runs in parallel on other cores**

**Return core to OS**
Performance Evaluation

Sanctuary Life Cycle

- **OS Execution**: Sensitive app execution triggered
- **Sanctuary Setup**: 440 ms w/o shared cache
- **Sensitive App Execution**: OS still runs in parallel on other cores
- **Sanctuary Teardown**: 110 ms w/o shared cache
- **OS Execution**: Return core to OS
Performance Evaluation

Sanctuary Life Cycle

- OS Execution
- Sanctuary Setup
- Sensitive App Execution
- Sanctuary Teardown
- OS Execution

Sensitive app execution triggered

Provision OTP Key: 1.2 s (w/o shared cache)

Display OTP: 600 ms (w/o shared cache)

Return core to OS
Code Modifications

- Normal World
  - Sensitive App
  - Sanctuary Library
- Secure World
  - Security Primitives

- Trusted Firmware
Code Modifications

- Normal World
  - Sensitive App
  - Sanctuary Library
- Secure World
  - Security Primitives

- Trusted Firmware

Reused OP-TEE
Modified 530 LOC
Code Modifications

- Reused ARM-TF modified 92 LOC
- Reused OP-TEE Modified 530 LOC

Diagram:

- Normal World
  - Sensitive App
  - Sanctuary Library
- Secure World
  - Security Primitives
  - Trusted Firmware
Code Modifications

- Used Zircon micro kernel modified 211 LOC
- Reused ARM-TF modified 92 LOC
- Reused OP-TEE Modified 530 LOC
Conclusion

• Sanctuary provides an ecosystem of mutually distrusted enclaves on ARM devices
Conclusion

• Sanctuary provides an ecosystem of mutually distrusted enclaves on ARM devices

✔ Sanctuary does not introduce new HW designs
Conclusion

• Sanctuary provides an ecosystem of mutually distrusted enclaves on ARM devices
  
  ✅ Sanctuary does not introduce new HW designs
  
  ✅ Sanctuary does not replace existing code bases
Conclusion

• Sanctuary provides an ecosystem of mutually distrusted enclaves on ARM devices
  ✓ Sanctuary does not introduce new HW designs
  ✓ Sanctuary does not replace existing code bases
  ✓ Sanctuary does not impact the commodity OS heavily
Conclusion

• Sanctuary provides an ecosystem of mutually distrusted enclaves on ARM devices
  ✔ Sanctuary does not introduce new HW designs
  ✔ Sanctuary does not replace existing code bases
  ✔ Sanctuary does not impact the commodity OS heavily

• Current Work
  ➢ Implement further use cases (IP protection of ML algorithms, digital car key)
  ➢ Sanctuary for RISC-V
Questions?

emmanuel.stapf@trust.tu-darmstadt.de