



Monthly Research 2017.3
TrustZone use case and trend

FFRI, Inc.

<http://www.ffri.jp/en>

E-Mail: [research-feedback\[at\]ffri.jp](mailto:research-feedback[at]ffri.jp)

Twitter: @FFRI_Research

Table of Contents

- About TrustZone
 - Use case of TrustZone
 - Cortex-A TrustZone
 - Cortex-M TrustZone
 - TEE implementation
- Vulnerability of TEE implementation
- Conclusions
- References

About TrustZone

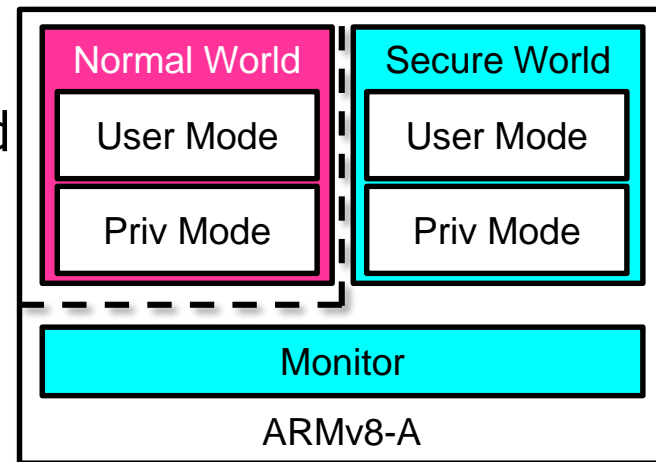
- About ARM TrustZone
 - Security mechanism of ARM processor
 - Cortex-A series and Cortex-M series are different mechanism
 - It makes a trusted world and non-trusted world on memory
 - For payment system, NFC, etc.,
- TEE (Trusted Execution Environment)
 - Secure execution environment realized at hardware level
 - Enactment by Global Platform and Trusted Computing Group
 - Global Platform create and publish specifications for secure chip and more

Use case of TrustZone

- Android 7.0 requires the implementation of a keystore using TrustZone etc.
- ARM Ltd. proposes the following use cases of TrustZone
 - Mobile payment
 - Credit card information and transaction are protected in a trusted world
 - Digital Rights Management
 - DRM data are protected in a trusted world
 - Credential information
 - Credential information such as fingerprint are protected in a trusted world

Cortex-A TrustZone

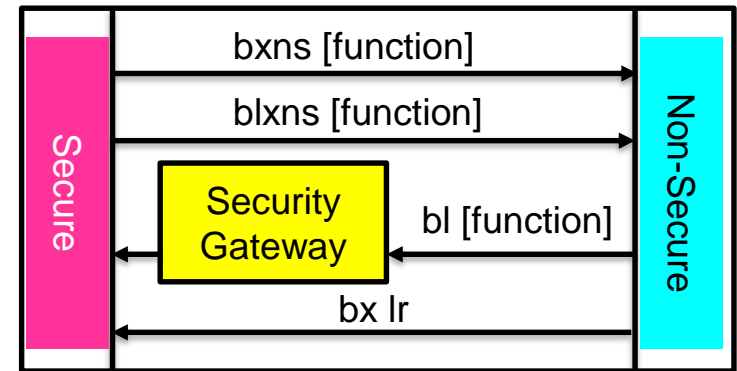
- Cortex-A series
 - Used a mobile and network device and more
 - ARMv7-A and ARMv8-A
- TrustZone
 - Memory isolation of Normal World and monitor mode
 - Trusted OS works in the Secure World
 - Switch the monitor mode by the SMC instruction



- For more information FFRI Monthly Research 2013.3

Cortex-M TrustZone

- TrustZone for low-power chips
 - ARMv8-M, s Cortex-M23/M33
- Memory isolation mechanism using state
 - Each memory area have state
 - Secure State, Non-Secure State, Non-Secure Callable State
- Non-Secure State call the Secure State
 - Via security gateway by the BL
 - Return to Secure State by the BX
- Secure State call the Non-Secure State
 - Use the BXNS and BLXNS
- For more information FFRI Monthly Research 2016.2



Well-known TEE implementation

- Trusty TEE
 - Developed by Google
 - Third party can't add the trusted application
 - Source code has published in AOSP
- OP-TEE
 - Developed by STMicroelectronics and Linaro Security Working Group
 - Source code has published in GitHub
 - Supports many board using Cortex-A series processor
- ARM mbed
 - An IoT platform developed by ARM Ltd.
 - Source code has published in the bed official site
 - It realizes TEE on board with Cortex-M

Security Research for TEE implementation

- Gal Beniamini (2017) TrustZone TEEs An Attacker's Perspective
 - Lecture at BlueHat IL Security Conference held by Microsoft
 - Analysis target are Qualcomm TEE and Trustonic TEE
- Both TEEs also were pointed out problems
 - ASLR is unused or entropy deficiency
 - Stack cookie is unused or implementation deficit
- The speaker recommends TEE implementation will be open source to get many reviews
- There is a risk that the vulnerability of TA is exploited and TEE breaks down if the security mechanism of the TEE implementation is insufficient

Vulnerability detail

- Qualcomm TEE
 - Insufficient entropy of ASLR
 - Non-Secure World OS can load a Trusted App(TA) into the Secure World user area(QSEE)
 - Trusted OS (QSEOS) load the TA at QSEE, but available memory is very limited
 - Buffer overflow
 - A TA check an overflow using stack cookie, but stack cookie is not random because many TA use BSS buffer
 - TA stack exist immediately after BSS, and a guard page doesn't exist
 - System call
 - QSEOS receives pointers at a system call from QSEE
 - QSEOS kernel memory can destruction because QSEOS don't check pointer value
- Trustonic TEE
 - ASLR and stack cookie are not available for TA

Conclusions

- TrustZone isolates memory for the trusted world and the non-trusted world
 - Security mechanism to realize TEE(Trusted Execution Environment)
 - TrustZone protect the confidentiality code or resource
 - Available in Cortex-A and Cortex-M
- TEE implementation with TrustZone
 - OP-TEE, Trusty TEE, mbed and more
- Vulnerability of TEE implementation
 - ASLR and stack cookie are incomplete
 - TEE implementation should be open sourced and reviewed more
 - Should use TEE implementation with solid ASLR and stack protection

References

- ARM Security Technology
 - http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf
- TrustZone TEEs An Attacker's Perspective
 - <https://microsofttrnd.co.il/Press%20Kit/BlueHat%20IL%20Decks/GalBeniamini.pdf>
- Trusty TEE | Android Open Source Project
 - <https://source.android.com/security/trusty/>
- mbed OS | mbed
 - <https://www.mbed.com/en/platform/mbed-os/>
- OP-TEE
 - <https://www.op-tee.org/>