vTZ: A Case for TrustZone Virtualization

Zhichao Hua, Jinyu Gu, Wenhao Li, Yubin Xia, Haibo Chen

Institute of Parallel and Distributed Systems, Shanghai Jiao Tong University

ABSTRACT

ARM's TrustZone [2] hardware security extension has been widely used for years, like in Samsung's S5 and Apple's iPhone. Based on TrustZone, many Trusted Execution Environment(TEEs) are developed, including Trustonic, Qualcomm's QSEE and T6 [1]. Most of these TEEs have been deployed on mobile phones to host various Trusted Applications (TAs) for different functionalities. At the same time, more and more virtualization features are supported in ARM platforms, like new CPU mode *HYP mode*, two-stage address translation and System Memory Management Unit (SMMU). Such virtualization extensions make ARM more suitable for server applications. Unfortunately, as TrustZone currently provide almost no support for virtualization, all virtual machines have to share one secure world. As a result, this violates both the isolation and transparency requirement by virtualization.

To address this issue, we introduce vTZ, which aims at providing transparent virtualization of TrustZone while still maintaining the strong isolation among virtualized TrustZone instances. vTZ leverages the virtualizaion extensions of ARM to trap and emulate all functionalities of a real TrustZone, enabling guest VMs to deploy more complex systems in its virtualized private secure world. Further, vTZ leverages the real TrustZone to partition computing resources between a VM's normal world and its virtualized secure world, which results in strong isolation among virtualized secure worlds.

1. DESIGN AND IMPLEMENTATION

Figure 1 shows the design of vTZ. vTZ uses a Trusted Virtual Machine(TVM) to emulate the secure world of every guest, while the normal VM works as the normal world of guest. Guest can uses SMC instruction, which is trap and emulated by vTZ, to switch between its normal VM and TVM. To ensure one guest's normal VM and TVM work as one virtual machine's two execution mode, vTZ uses a modified scheduler to manage them. vTZ leverages Nest Page Table(NPT) to isolate and share the memory between guest's different execution environments. It controls the GPA-to-HPA mapping so that the secure memory accessing from guest's normal VM will trigger a page fault, then vTZ can forbid such operation. For the normal memory, vTZ maps the normal physical memory of one guest's VM and TVM to the same host physical memory, so that both of guest's execution environments can share the normal memory. vTZ uses a Device Privilege Controller(DPC) to isolate the device and interrupt of each guest. One Device Privilege Table(DPT) for every guest is used to manage guest's device and interrupt. For I/O devices, DPC intercepts the I/O request from guest's VM and TVM, then checks whether such request is legal or not according to the DPT. DPC controls the interrupt by intercept all interrupts injected to guest's VM and TVM, then redirects them to the legal target.

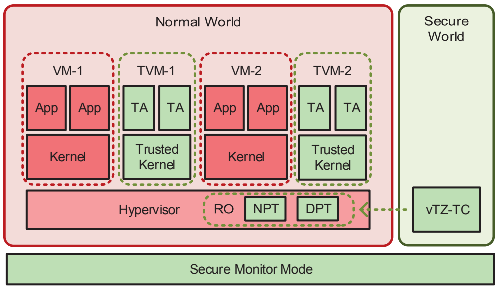

Figure 1: Overview of vTZ

Security Enhancement: vTZ exploits the real TrustZone technology to enhance the system security. It uses a vTZ-TC running in the secure world to manage all NPT and DPT in the normal world. Whenever hypervisor needs to modify NPT or DPT to configure guest's resource partition, it has to send a request to the vTZ-TC to finish such operation. vTZ-TC will uses some polices to check each request, for example the HPA containing guest's secure memory cannot by mapped to any other VM or hypervisor itself. With the help of vTZ-TC, vTZ only have a very small Trusted Computing Base(TCB), and provides a strong isolation.

2. REFERENCES

[1] T6. http://trustkernel.org.

[2] T. Alves and D. Felton. Trustzone: Integrated hardware and software security. *ARM white paper*.

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235