Mobile Hardware Security

Vikas Chandra and Rob Aitken
ARM R&D
Hot Chips, August 2014
Outline

- Mobile security background
- Trusted Execution Environment
- ARM TrustZone™
- Implementation and use cases
- Authentication
About ARM…

- 50 Billion ARM-based chips shipped (>10B /year)
- ~ $1.2B Revenue/year
- ~3000 Employees
- >95% Smartphone & Tablet market share
The Mobile Threat Environment

- Increasing risks
  - Social engineering – Trojans, phishing, APT
  - Malware
  - Physical loss or theft leading to risk to data – calendar, phonebook and email
  - Improperly secured devices – no PIN lock
  - User intervention – jailbreaking, unlocking
  - Mobile has become the enterprise security boundary

**Need to design in the right system-wide security (not just more security)**
Whose Data Is Involved?

- **User**
  - Personal information, contacts, location, photos, etc.

- **Enterprise(s)?**
  - Bring your own device (BYOD)

- **Carrier**
  - Network interface

- **Apps**
  - Content providers
    - DRM for movies, songs, etc.
  - Finance companies
    - Account data, passwords
  - IoT
    - Home automation, health, etc.
Security Profiles

- **Invasive Hardware Attacks**
  - Well resourced and funded
  - Unlimited time, money & equipment

- **Non-invasive Hardware Attacks**
  - Side channels (DEMA, DPA)
  - Physical access to device – JTAG, bus probing, IO pins, etc.

- **Software Attacks**
  - Malware & viruses
  - Social engineering

- **TrustZone® technology-based TEE**

- **SmartCards/HSMs**

Cost/Effort To Attack vs. Cost/Effort to Secure
Mobile Solution Is Not PC Solution

- PC-era security
  - Add layers of software security (SSO, etc.)
  - Add hardware security (CVC, key fobs, etc.)
- Too unwieldy and confusing for mobile environment
Mobile Security Approach

- Hypervisor (with hardware support) separating large pieces of code

- Small, certifiable Trusted Execution Environment (TEE) inside application processor isolated using ARM TrustZone technology protecting against software attacks

- Secure element for tamper-proof security (where needed)
Trusted Execution Environment

- Hardware root of trust
  - A basis for system integrity
- Integrity through Trusted boot
- Secure peripheral access
  - Screen, keypad, fingerprint sensor, etc.
- Secure application execution
- Trust established outwards
  - With normal world apps
  - With internet/cloud apps
Castle Analogy

- Layers of defense
- Reducing attack surface
- Increasing isolation
- Principle of least privilege
- Most precious assets protected by multiple layers of security
Castle Analogy

But…

- Modern OS/Framework is ~10GB + GBs of apps
- So maybe we should think of a walled city and castle
- Attacks happen
- Everyone knows what the assets are and which room they are in
- Where to put high-value assets such as keys?

Implementation details matter!
Castle Analogy with TrustZone Based TEE

- TrustZone technology-based TEE creates a second (much smaller security boundary) castle with only one door, carefully designed entry/exit & APIs
- Keys only used in secure world, protected crypto, encrypted storage, secure execution, secure peripherals
- Offers: Integrity (part of trusted boot) Confidentiality
- TrustZone TEE castle is invisible to normal world

10-20 GB

1-2MB
Castle Analogy with TrustZone Based TEE

Secure World

Isolated Trusted Apps

Trusted OS

e.g. Trustonic t-base300

Normal World

GlobalPlatform Client API

SMC calls at EL3

e.g. ARM Trusted Firmware

10-20 GB

1-2MB

Confidential
In pre-TrustZone systems:

- Rigid allocation of MHz/resources independent of the application
- Silicon costs with redundant hardware that is idle most of the time
- Complex control logic and deficient performance and power consumption
TrustZone Basics

Key advantages over separate secure processor solutions:

- CPU MHz/resources are dynamically shared
- Two domains in same machine
  - Difficult to give precise “overhead” values since secure and non-secure tightly integrated from design standpoint
- Use exceptions to move between modes
# AArch64 Exception Levels

## Exception Level (EL)

- **EL0**: S User
- **EL1**: S Priv
- **EL2**: S Monitor
- **EL3**: TrustZone Monitor

### System Components

- **NS User**: App1, App2, App1, App2
- **NS Priv**: GuestOS1, GuestOS2
- **NS Hyp**: Virtual Machine Monitor (VMM) or Hypervisor
- **S Monitor**: TrustZone Monitor
- **Secure OS**: Sec App1, Sec App2
AArch64: Exception Model

- 4 exception levels: EL3-EL0
  - Forms a privilege hierarchy, EL0 the least privileged (user mode)

- Exception link register written on exception entry
  - Interrupt masks set on exception entry
  - 32-bit to 64-bit exception zero-extends the link address

- Exceptions can occur to the same or a higher exception level
  - Different vector base address registers for EL1, EL2, and EL3

- Vectors distinguish:
  - Exception type: synchronous, IRQ, FIQ or system error
  - Exception origin (same or lower exception level) and register width

http://www.arm.com/products/processors/armv8-architecture.php
## AArch64 Registers

<table>
<thead>
<tr>
<th></th>
<th>X0</th>
<th>X8</th>
<th>X16</th>
<th>X24</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>X9</td>
<td>X17</td>
<td>X25</td>
<td></td>
</tr>
<tr>
<td>X2</td>
<td>X10</td>
<td>X18</td>
<td>X26</td>
<td></td>
</tr>
<tr>
<td>X3</td>
<td>X11</td>
<td>X19</td>
<td>X27</td>
<td></td>
</tr>
<tr>
<td>X4</td>
<td>X12</td>
<td>X20</td>
<td>X28</td>
<td></td>
</tr>
<tr>
<td>X5</td>
<td>X13</td>
<td>X21</td>
<td>X29</td>
<td></td>
</tr>
<tr>
<td>X6</td>
<td>X14</td>
<td>X22</td>
<td>X30*</td>
<td></td>
</tr>
<tr>
<td>X7</td>
<td>X15</td>
<td>X23</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* _procedure_ LR

<table>
<thead>
<tr>
<th>EL0</th>
<th>EL1</th>
<th>EL2</th>
<th>EL3</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP = Stack Ptr</td>
<td>SP_EL0</td>
<td>SP_EL1</td>
<td>SP_EL2</td>
</tr>
<tr>
<td>ELR = Exception Link Register</td>
<td>ELR_EL1</td>
<td>ELR_EL2</td>
<td>ELR_EL3</td>
</tr>
<tr>
<td>Saved/Current Process Status Register</td>
<td>SPSR_EL1</td>
<td>SPSR_EL2</td>
<td>SPSR_EL3</td>
</tr>
</tbody>
</table>
Attack Approach: Man In The Middle

Can then access memory used to communicate between client app and trusted app.

Malicious app can intercept traffic, replace it, modify it or eavesdrop.

Secure call to TEE

Malicious app somehow attacks OS/kernel.
Side-Channel Attacks

Rich OS

Client App

Malicious App

OS Kernel

TEE Lib

TEE Kernel

Trusted App

Trusted App

Trusted Execution Environment

Secure call to TEE

Monitor

Confidential
Defenses

- Normal World to Secure World communications are always exposed and vulnerable

Mitigation

- Don’t design systems that rely on secure communications between Normal World and Secure World
- Always use trustworthy components – crypto library, TEE and protocols

Malicious app can intercept traffic, replace it, modify it or eavesdrop
Propagating System Security

NS: NOT Secure, treated like an address line
## TrustZone Controllers – Vital Statistics

<table>
<thead>
<tr>
<th>Code</th>
<th>Product</th>
<th>Main Function</th>
<th>Key Features</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>TZC-380</td>
<td>TrustZone Address Space Controller</td>
<td>Partition external DRAM into secure and non-secure regions</td>
<td>Configurable up to 16 regions of size 32K to 4G, each with 8 sub-regions (down to 4K). Configurable registering to meet timing constraints with minimum latency. AXI interface for compatibility with NIC-301 and DMC-34x.</td>
<td>10-100k gates</td>
</tr>
<tr>
<td>BP141</td>
<td>TrustZone Internal Memory Wrapper</td>
<td>Protects internal SRAM</td>
<td>Manages a single secure region within the SRAM. AXI interface.</td>
<td>&lt;1k gates</td>
</tr>
<tr>
<td>BP147</td>
<td>TrustZone Protection Controller</td>
<td>Prevents non-secure accesses to peripherals</td>
<td>Allows peripherals to be safely shared by the Secure and Non-Secure worlds. APB interface.</td>
<td>&lt;1k gates</td>
</tr>
</tbody>
</table>
Application of Hypervisor for BYOD

Two personas

- Mutual distrust model between OSs
- Ensuring enterprise OS Security, while protecting consumer OS privacy
- Enabling enterprises to have control of their own assets in case of loss
Secure Content Path: SoC Requirements

Firmware protected against tampering:
- Any software component directly used in setting up protected memory path
- Decoders, mixers, renderers, DRM
- Critical components placed in secure processing space
- Integrity checked at boot time

Unencrypted content protected:
- After DRM protection removed
- Unencrypted content never accessible to processes running in HLOS
- Unencrypted content only ever written to protected memory

Memory buffers protected by hardware control:
- All memory used in processing, decoding, mixing and rendering
- Sufficient memory for video bitstream and frame buffer
- Not accessible by HLOS or unauthorized HW or SW
- Output only to internal display or via protected export clients such as HDCP and DTCP
Secure Implementation Example

**Normal World**
- Video Player
- DRM Client
- HLOS

**Secure World (TEE)**
- Video Trusted App
- DRM Trusted App
- Secure OS in TEE
- Secure Monitor/Boot

**Low cost and complexity**
- Secure CPU, bus fabric and Video from a single source
- System IP designed to work together
- Simple SW integration – create a secure session then manage scheduling/control as normal

**Minimal memory fragmentation**
- Major issue for HD content
- Video MMU can be used for secure sessions by TEE
- No need to assign large, contiguous secure buffers

**Increased flexibility and protection**
- Simultaneous protected and un-protected video streams
- Additional protection of video firmware (read-only) and data (non-executable)

**Components**
- ARM CPU with TrustZone Extensions
- Mali-V500
- Mali Display & Composition
- "Firewall" (e.g. ARM TZC400)
- Rich OS Memory
- Trusted “Protected” Memory
Developing Security – Hierarchy of Trust

Rich Domain
- User Applications
- Android or other OS

Protected Domain
- Protected Video Path
- BYOD System Mgmt
- Virtual machines and bus masters isolated by a hypervisor
- Hypervisor HYP

Trusted Domain
- Secure Firmware Device Management
- Key Management
- Trusted applications executing from a Trusted Execution Environment
- TrustZone® TEE

Secure Domain
- Secure Element Smartcard
- SIM & TPM
- Tamper proof, physically isolated, EAL certified
- SecurCore™ SEE
## Use Cases for Hierarchy of Trust Domains

<table>
<thead>
<tr>
<th></th>
<th>Client Asset Protection</th>
<th>Content Protection</th>
<th>Mobile Payment</th>
<th>Enterprise</th>
<th>Government</th>
<th>Automotive</th>
<th>Server</th>
<th>Certification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secure</td>
<td>UICC for smartphones</td>
<td>Secure Storage (CA)</td>
<td>Credit Card Payment Wallet NFC</td>
<td>mPOS</td>
<td>Strong Authentication of user credentials</td>
<td>Strong Authentication of user credentials (eID)</td>
<td></td>
<td>GlobalPlatform EMVCO</td>
</tr>
<tr>
<td>Trusted</td>
<td>MDM SIM Lock</td>
<td>Trusted Storage (Keys)</td>
<td>Cloud Payment Strong Authentication - Trusted UI Wallet</td>
<td>(TEE + SE)</td>
<td>MDM BYOD Trusted UI Integrity Trusted VPN Trusted FOTA</td>
<td>MDM Trusted UI Key Store Trusted Services (eTS)</td>
<td>Infotainment Integrity</td>
<td>Trusted boot Integrity</td>
</tr>
<tr>
<td>Protected</td>
<td>Apps isolation</td>
<td>SW DRM</td>
<td>Isolation</td>
<td>BYOD</td>
<td>Dual Persona</td>
<td>Isolate Infotainment and CAN</td>
<td>VMs KVM, XEN…</td>
<td>FIPS 140.2</td>
</tr>
<tr>
<td>Rich</td>
<td>SW Crypto</td>
<td>SW DRM</td>
<td>Web Remote Payment (SSL)</td>
<td>SW BYOD SW FOTA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Web Payment Example: MasterPass

1. Payment Details
   Merchant and price of the transaction are confirmed before payment takes place.

2. Secure Handover
   Control of user interface is handed over to TrustZone

3. PIN Entry
   The PIN is captured through TrustZone hardware and secured by a trusted application before being passed back to the

4. PIN Check & Authorisation
   The rich OS regains control, submits the encrypted PIN to the Secure Element for checking, and submits the transaction to merchant payment gateway

Merchant Site
Merchant World

Merchant Shopping
The merchant ‘pay now’ element drives the TrustZone app to launch, delivering the necessary information to be displayed

Merchant Confirm
A confirmation is returned to the merchant to confirm the payment was successfully taken
Current Practice

Three recent whitepapers from Apple, Samsung and Microsoft give good insight into modern mobile security practice.

CESG General Security Recommendations

- Assured Data at Rest
- Assured Data in Transit
- Authentication
- Secure Boot
- Platform Integrity
- Application white listing
- Malicious Code Detection
- Security Policy Enforcement
- External Interface Protection
- Device Update Policy
- Event Collection
- Incident Response

Encrypted Data at Rest and Data in Transit

- Assured data at rest: Data at rest should be suitably encrypted
  - Typical ARM SoC has a crypto hardware engine to encrypt/decrypt files
    - Also crypto extensions in AArch64
  - Hardware Unique Key available only to Trusted OS, fused into silicon can be used to derive other keys
  - Key material can be kept on Secure World side or encrypted “wrapped” and stored as metadata
  - System Integrity as determined by Trusted Boot can be verified before the data is decrypted

- Assured data in transit: IPSec VPN of “assured foundation grade” & configured appropriately
  - TrustZone technology-based TEE can add strong 2-factor authentication for remote working
Authentication and Secure Boot

- Authentication:
  - User to Device, User to Service, Device to Service
  - Trusted peripherals are handled only by the Secure World
  - Protocols such as FIDO will simplify the silo nature of the authentication status quo

- Secure Boot:
  - Should not be modifiable by unauthorized entity and attempts should be detected
  - Device boots into Secure World and runs only cryptographically verified boot loaders
  - Device starts Trusted OS before main OS is started
  - Measurements of boot process can be made to test for tampering
Need for Authentication

- People use the same simple passwords (Analysis of 6m accounts showed that 10k common passwords would give access to 98.8% of the accounts)
  - 1k passwords give access to 90% of the accounts see https://xato.net/passwords/more-top-worst-passwords
  - 10k passwords give access to 98.8% see https://xato.net/passwords/more-top-worst-passwords/

- People reuse them
  - In 2007: People had 25 accounts and used 6.5 passwords (see Large Scale Study on Web Password habits)
  - 73% of the users shared their online banking password with at least one non-financial site
FIDO Functionality

- Discovery of authenticators on the client
- Registration
- Authentication
- Transaction confirmation
FIDO User Experiences

PASSWORDLESS EXPERIENCE (UAF standards)

Transaction Detail

Show a biometric

Done

SECOND FACTOR EXPERIENCE (U2F standards)

Login & Password

Insert Dongle, Press button

Done
How Does FIDO UAF* Work?

FIDO Authenticators

Verification

FIDO SERVER

*Universal Authentication Framework
FIDO - Universal Authentication Framework

**User Side**

- Plugin
- SDK
- FIDO Client
  - Authenticator Abstraction
    - Authenticators
      - Private Keys
      - Authentication Keys
      - Attestation Key

**Relying Party**

- Web Application
  - FIDO UAF Server
    - Attestation Manager
    - Crypto
      - Policy Rules
    - Public Keys
      - Authentication Keys
      - Attestation Key

UAF Protocol

Registration, Authentication & Transaction Confirmation

Ideal for TrustZone technology-based TEE
Conclusions - What Is Needed?

- Mobile security is built on hardware root of trust
- Requires hardware, software and services to work together
- Not all security is equal – consider ARM’s Hierarchy of Trust model
- CESG recommendations and OEM security whitepapers useful for orientation
- FIDO can help us move beyond passwords
Acronyms

ASID = address space identifier
BIU = bus interface unit
BYOD = bring your own device
CA = certificate authority
CESG = Communications-Electronics Security Group (UK)
CVC = card verification code (or card verifiable certificate for smart cards)
DNSSEC = DNS security extensions
DPA = differential power attack
DEMA = differential electromagnetic attack
DTCP = digital transmission content protection
FIDO = Fast IDentity Online
FIQ = fast interrupt request
FOTA = firmware over the air (secure updates)
HDCP = high bandwidth digital content protection
HLOS = high level operating system
HSM = hardware security module
IRQ = interrupt request
mPOS = mobile point of sale
NS = not secure
NSTID = nonsecure table identifier
PA = physical address
SMC = secure monitor calls
SSO = single sign-on
TEE = trusted execution environment
TLB = translation lookaside buffer
TLS = transport layer security
UAF = Universal authentication framework
UICC = universal integrated circuit card
VA = virtual address