

Fingerprints On Mobile Devices: Abusing And Leaking

Tao Wei and Yulong Zhang

More And More Mobile Vendors Equip Fingerprint Scanners



Saygus



Apple



Samsung

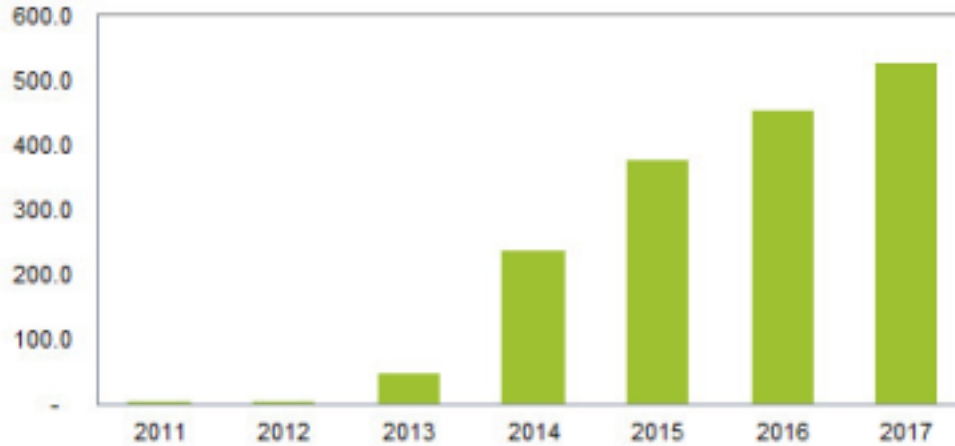


HTC



Huawei

Global Forecast of Shipments of Cellphones with Integrated Fingerprint Scanners
(Millions of Units)



Source: IHS Inc. November 2013

50% of smartphone shipments will have a fingerprint sensor by 2019

-- Research Capsule

Functionalities Associated with Fingerprints

- ◆ Authentication
 - ◆ System screen unlock
 - ◆ Authentications in FIDO Alliance services
- ◆ Authorization
 - ◆ iTunes/App store pay
 - ◆ Apple Pay
 - ◆ Transaction authentication using FIDO

Risks: Leaking Fingerprint Is A Disaster

Password leaked? Fine, you can easily replace it with a new one.



Risks: Leaking Fingerprint Is A Disaster

- ◆ Fingerprint leaked? Well, it is leaked for the rest of your life.
- ◆ Moreover, it is associated with your identity record, criminal history, immigration history, banking credential, etc.



<http://www.cnn.com/2010/WORLD/europe/07/05/first.biometric.atm.europe/>



https://en.wikipedia.org/wiki/Office_of_Biometric_Identity_Management

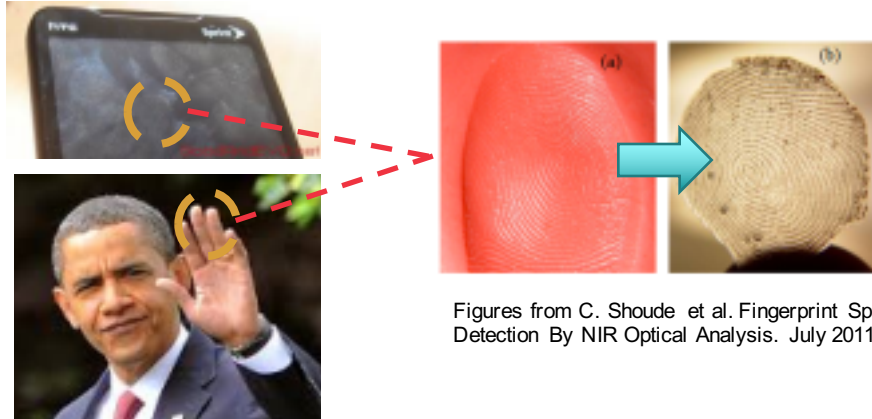


Image from: <https://s-media-cache-ak0.pinimg.com/736x/9f/64/55/9f64556ec24b6c9639b649f6e4a9b2c5.jpg>

It would be even worse if the attacker can **remotely harvest** fingerprints in a **large scale**.

Existing Optical Attacks

- ◆ Fingerprints can be stolen from its owner if a person touched any object with a polished surface like glass or a smartphone screen.
- ◆ Fingerprints can even be extracted from a waving hands photo.
- ◆ Attackers can spoof fingerprints accordingly using electrically conductive materials.



Figures from C. Shoude et al. Fingerprint Spoof Detection By NIR Optical Analysis. July 2011.

System Attacks against Fingerprints?!

- ❖ This talk will rather focus on:
 - Confused Authorization Attack
 - Unsecure Fingerprint Data Storage
 - Fingerprint Sensor Spying Attack
 - Backdoor of Pre-embedding Fingerprints

To our knowledge, we are the first to discuss system attacks against fingerprint auth frameworks

Outline

- ❖ **Design of Android Fingerprint Frameworks**
 - Fingerprint Recognition
 - Mobile Fingerprint Frameworks
- ❖ **System Attacks against Fingerprints**
 - Confused Authorization Attack
 - Unsecure Fingerprint Data Storage
 - Fingerprint Sensor Spying Attack
 - Backdoor of Pre-embedding Fingerprints
- ❖ **Discussion**

Outline

- ❖ Design of Android Fingerprint Frameworks
 - **Fingerprint Recognition**
 - Mobile Fingerprint Frameworks
- ❖ System Attacks against Fingerprints
 - Confused Authorization Attack
 - Unsecure Fingerprint Data Storage
 - Fingerprint Sensor Spying Attack
 - Backdoor of Pre-embedding Fingerprints
- ❖ Discussion

Fingerprint Minutiae Extraction



Grayscale
Image

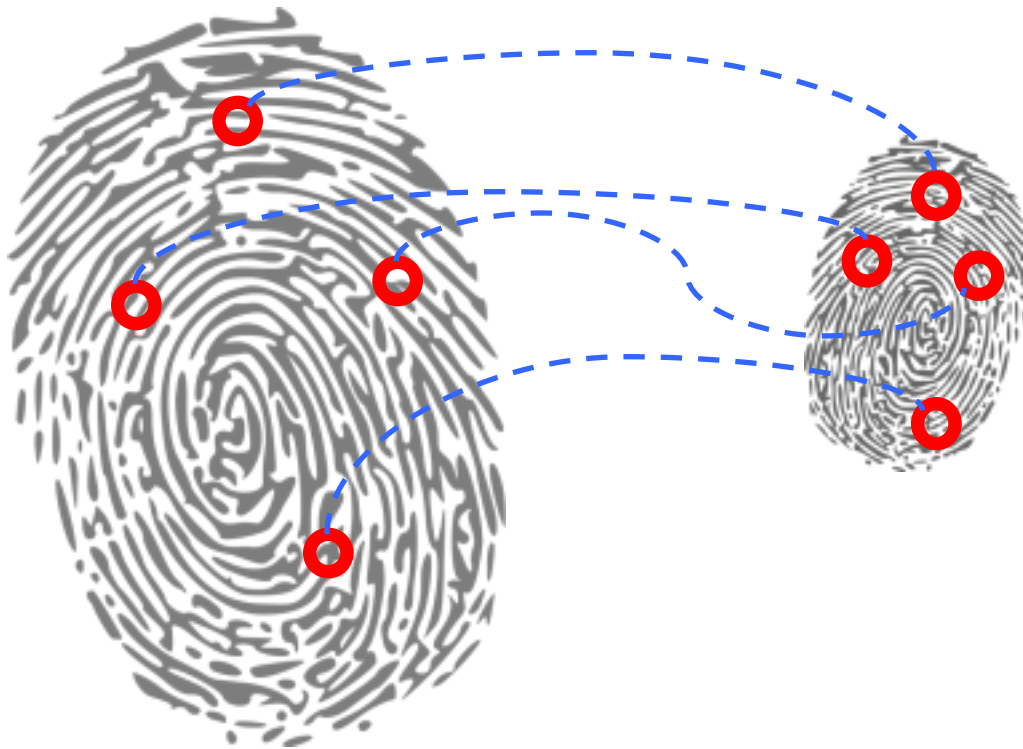
Phase
Image

Skeleton
Image

Minutiae

Figures from J. Feng and A. Jain, Fingerprint Reconstruction: From Minutiae to Phase
IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 33, NO. 2, FEBRUARY
2011

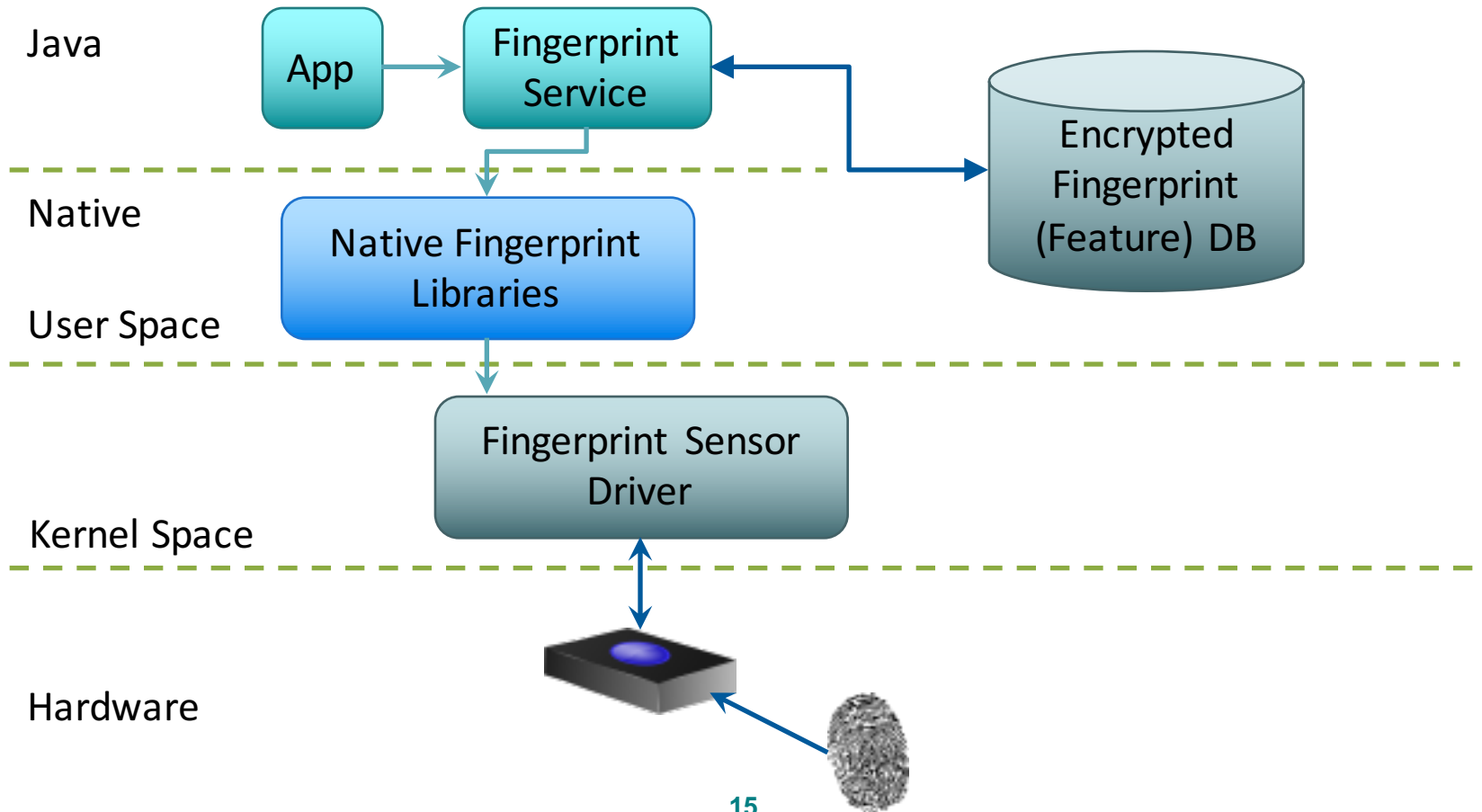
Fingerprint Minutiae Matching



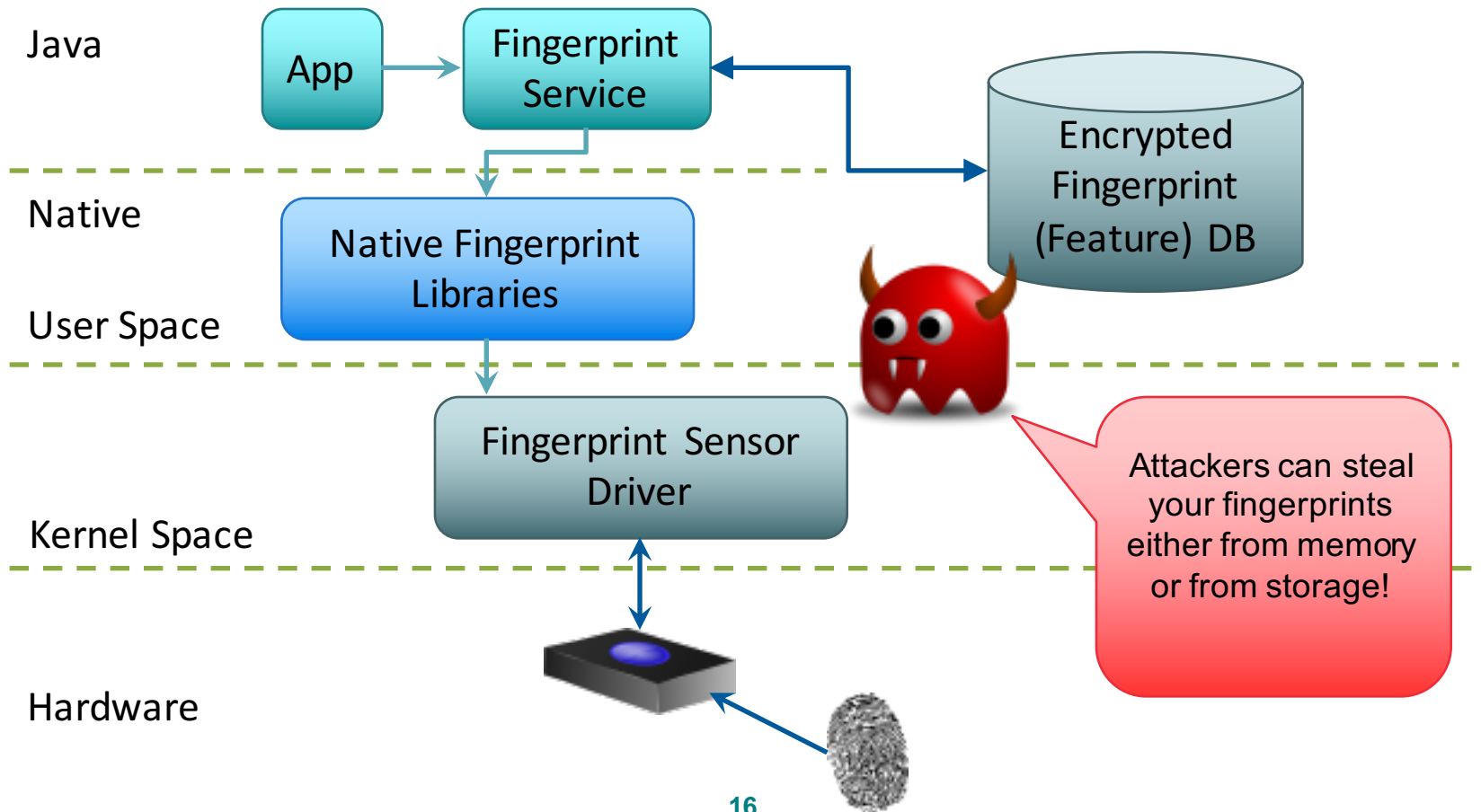
Outline

- ❖ Design of Android Fingerprint Frameworks
 - Fingerprint Recognition
 - **Mobile Fingerprint Frameworks**
- ❖ System Attacks against Fingerprints
 - Confused Authorization Attack
 - Unsecure Fingerprint Data Storage
 - Fingerprint Sensor Spying Attack
 - Backdoor of Pre-embedding Fingerprints
- ❖ Discussion

Fingerprint Framework without TrustZone



Threat: Rooting Attacks

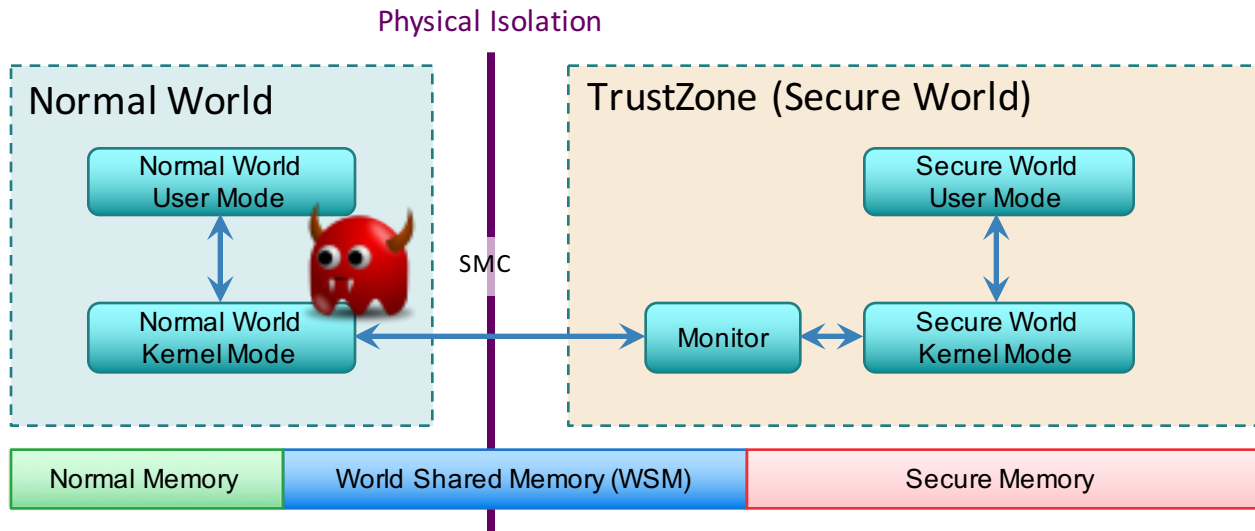


How to Defend against Rooting Attacks?

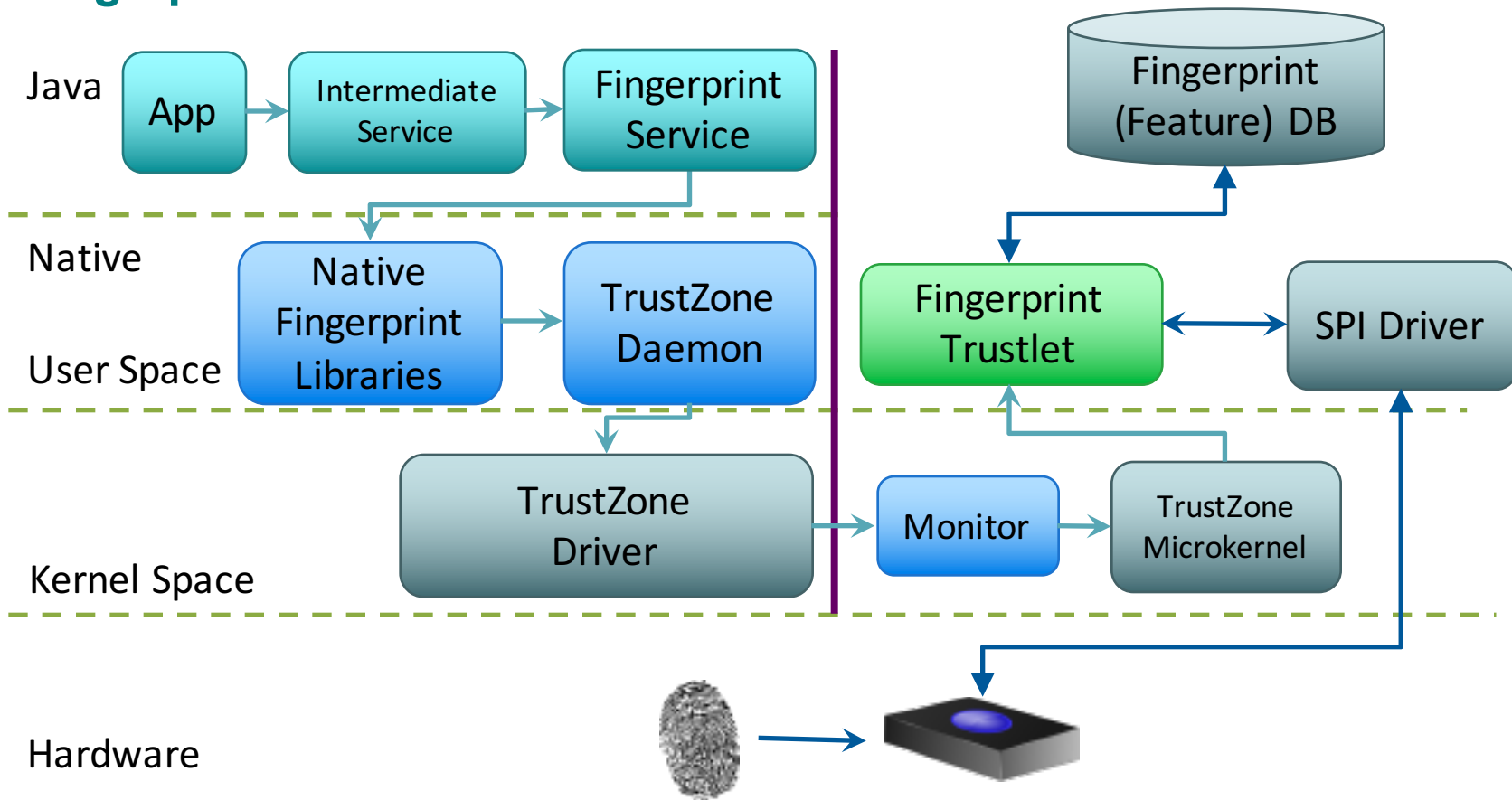
TrustZone



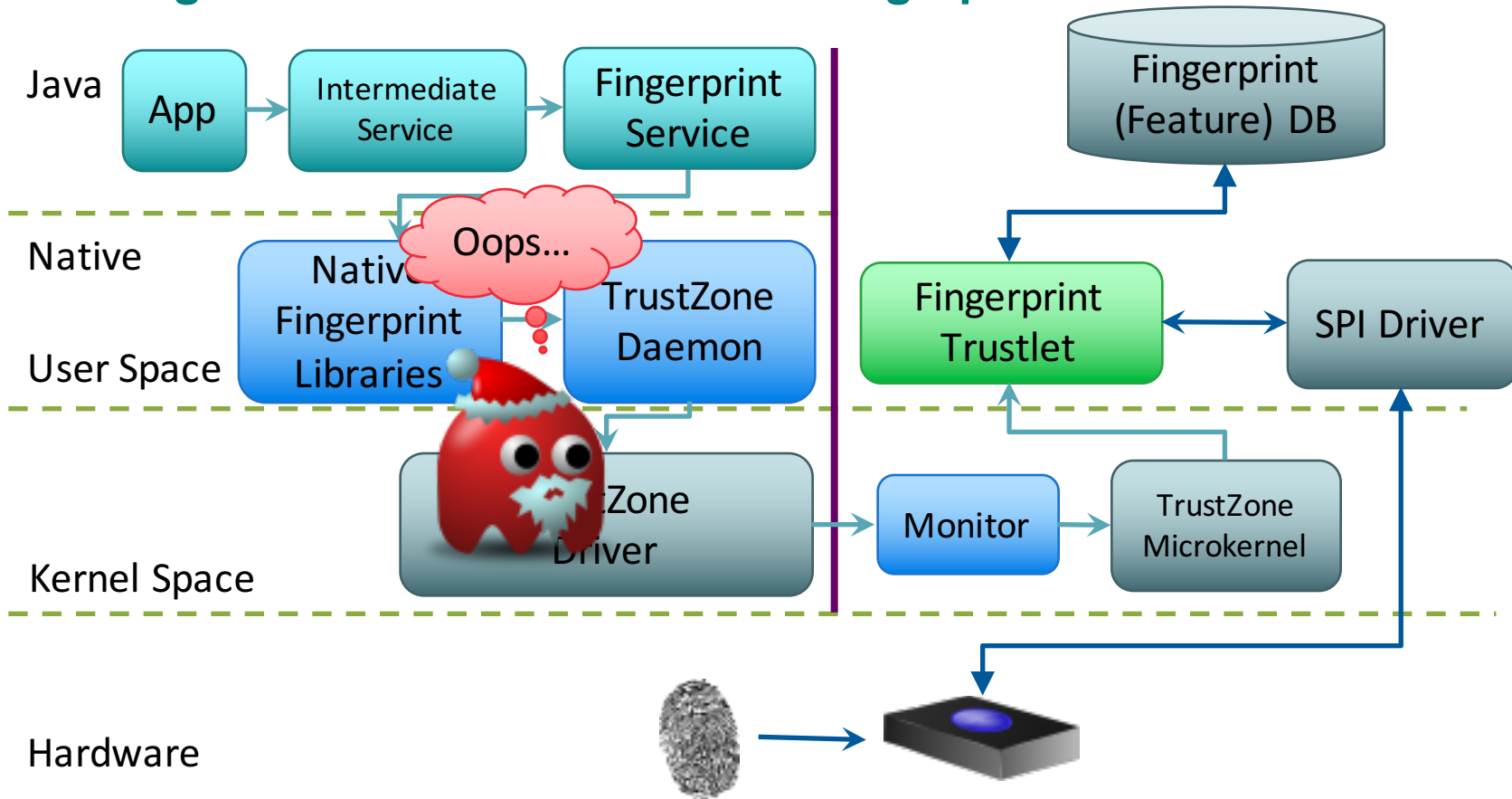
- ◆ Separate the system to the Normal World, and the Secure World
- ◆ Contain potential compromises in the Normal World



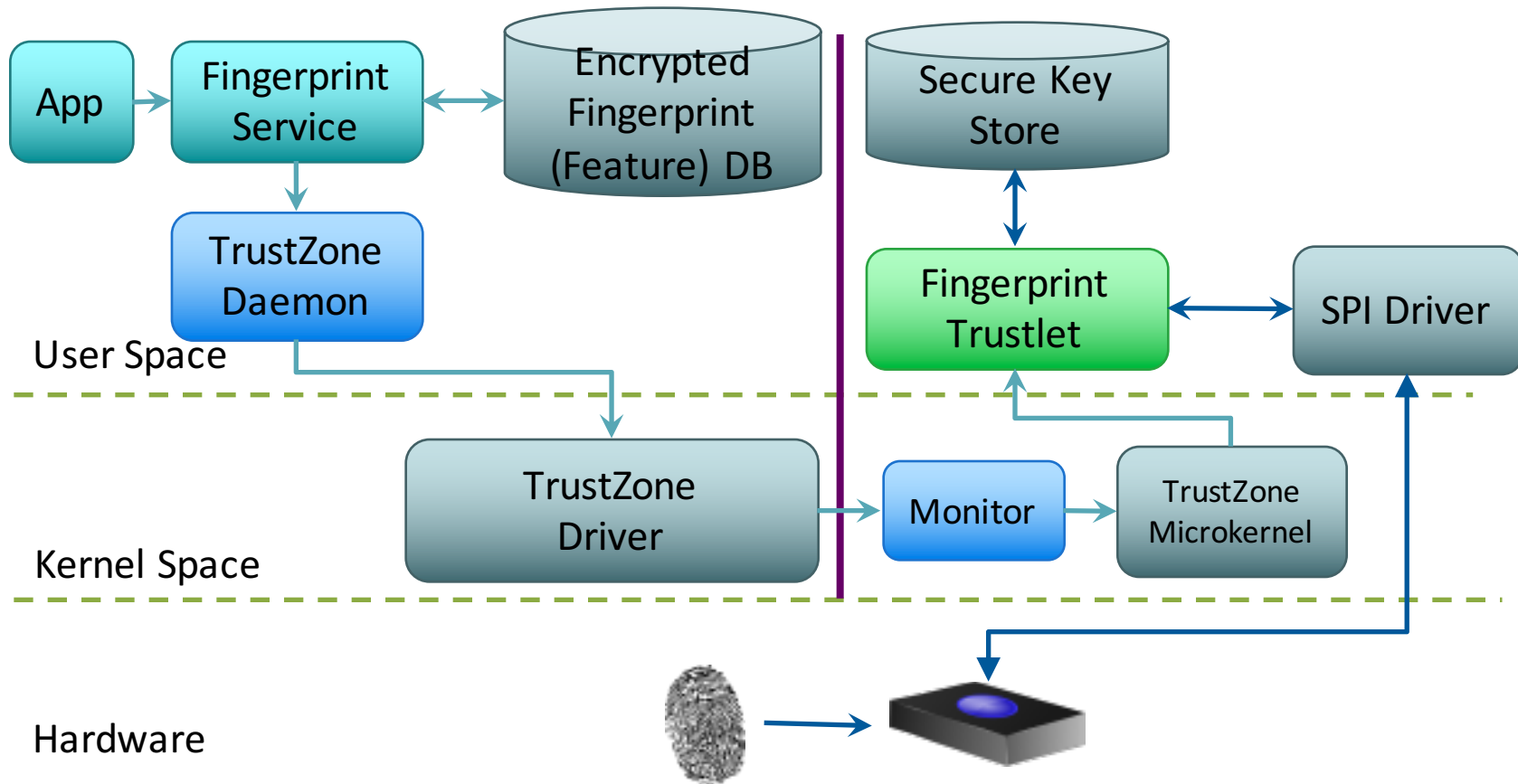
Fingerprint Framework with TrustZone



Rooting Attackers Cannot Access Fingerprints in TrustZone



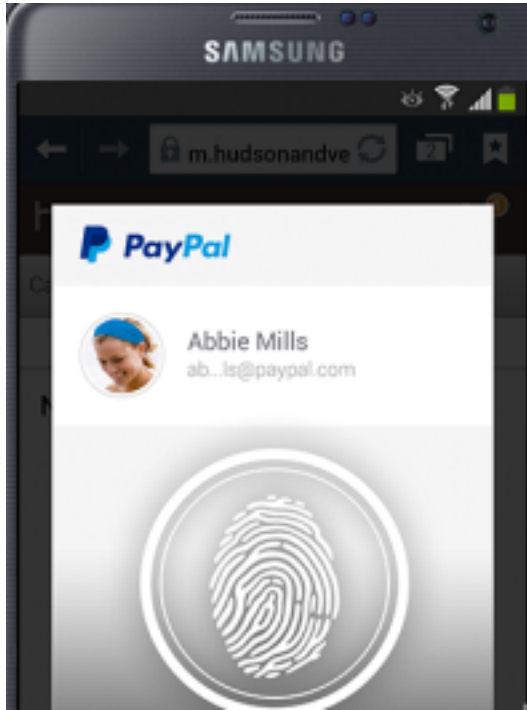
Fingerprint Authorization Framework with TrustZone



We Are Secure! Let's Ally: FIDO Alliance



Samsung Galaxy S5 (octa-core) Fingerprint Framework



PayPal™

➤ Money Transaction Service

**Nok Nok
LABS**

➤ Auth Protocol Implementation

SAMSUNG

➤ Phone Framework

Synaptics®

➤ Fingerprint Sensor

ARM®

➤ TrustZone Isolation of Exynos 5

Outline

- ❖ Design of Android Fingerprint Frameworks
 - Fingerprint Recognition
 - Mobile Fingerprint Frameworks
- ❖ **System Attacks against Fingerprints**
 - Confused Authorization Attack
 - Unsecure Fingerprint Data Storage
 - Fingerprint Sensor Spying Attack
 - Backdoor of Pre-embedding Fingerprints
- ❖ Discussion

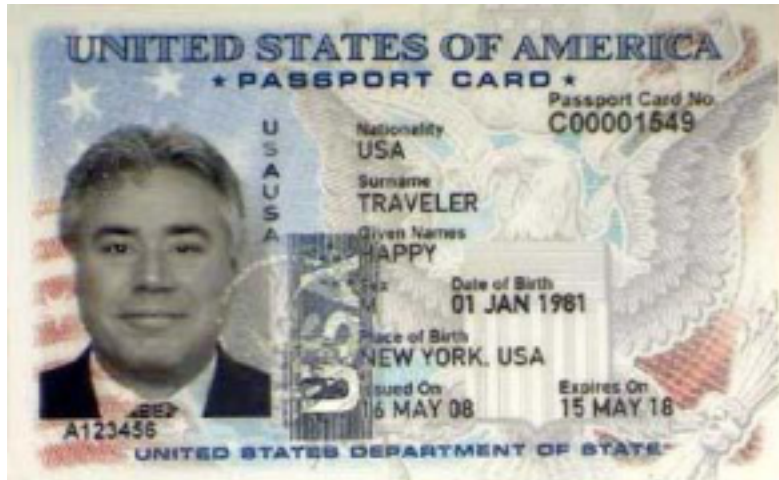
Outline

- ❖ Design of Android Fingerprint Frameworks
 - Fingerprint Recognition
 - Mobile Fingerprint Frameworks
- ❖ System Attacks against Fingerprints
 - **Confused Authorization Attack**
 - Unsecure Fingerprint Data Storage
 - Fingerprint Sensor Spying Attack
 - Backdoor of Pre-embedding Fingerprints
- ❖ Discussion

Confused Authorization Attack

Authentication

- ◆ Who you are (Passport)



Authorization

- ◆ What you can do (Visa)



Figures from Wikipedia

Authenticating



Figures from fcssl.com

Authorizing



Figures from dailytech.com

Authorizing: Context!



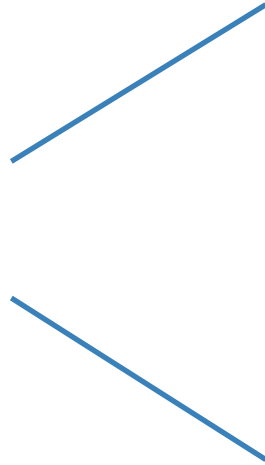
Figures from dailytech.com

To Swipe or Not To Swipe, without A Context?



Figures from dailytech.com

What are your fingerprints?



OR



Confused Authorization Attack

Demo!

Confused Authorization Attack

- ◆ Do you ever have a second thought when you swipe to unlock the device?

It can enable background attacker to steal your money from your mobile payment account!!!



Confused Authorization Attack

◆ Questions

How can I testify what's happening behind the finger swiping?

You can't tell...

What's the difference of swiping to unlock the device with swiping to authorize a mobile payment transaction?

You can't tell...

Confused Authorization Attack

- ◆ Applications often mistakenly treat **authorization** as **authentication**, and fail to provide context proofs for **authorization**.
- ◆ Without proper context proof, the attacker can mislead the victim to **authorize a malicious transaction** by disguising it as an **authentication** or **another transaction**.
- ◆ In the demo
 - ◆ The attacker fakes a lock screen to fool the victim to think that he/she is “**swiping finger to unlock the device**”, but the fingerprint is actually used to **authorize** a money transfer in the background.

FIDO Alliance's Specification



- Basically if a FIDO UAF Authenticator has a transaction confirmation display capability, FIDO UAF architecture makes sure that the system supports **What You See is What You Sign mode (WYSIWYS)**. A number of different use cases can derive from this capability -- mainly related to authorization of transactions (send money, perform a context specific privileged action, confirmation of email/address, etc).
- The transaction confirmation display component implementing **WYSIWYS** needs to be **trusted**

However...

- ◆ The original fingerprint auth framework (without TrustZone) has no reliable way to provide the **authorization context proof**.
- ◆ The framework with TrustZone can be improved to achieve this goal (the Trustlet modules in TrustZone can be modified to provide the **context proof**), but so far (June 2015) we haven't seen any major vendor that implemented this feature.

Outline

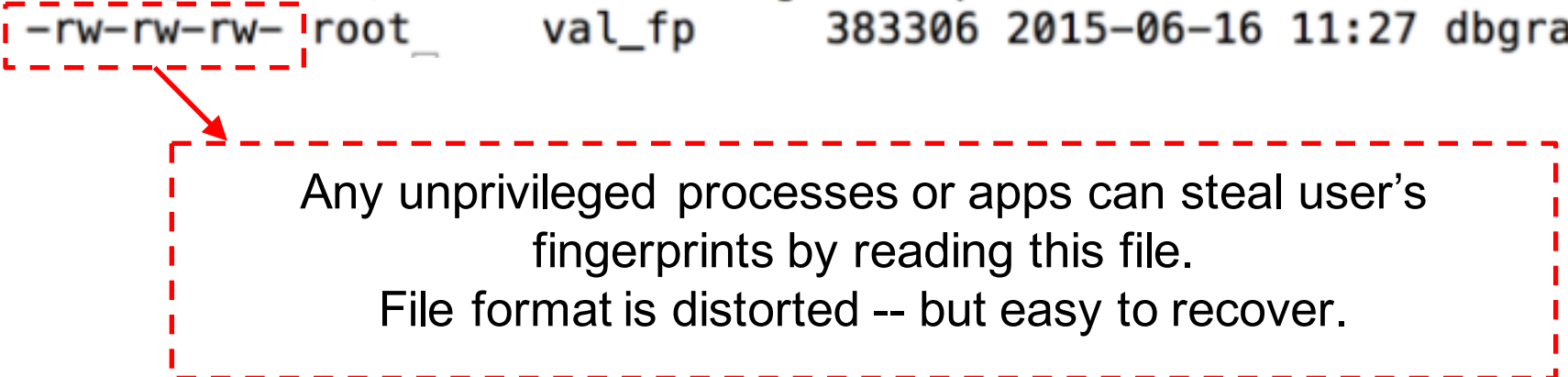
- ❖ Design of Android Fingerprint Frameworks
 - Fingerprint Recognition
 - Mobile Fingerprint Frameworks
- ❖ System Attacks against Fingerprints
 - Confused Authorization Attack
 - **Unsecure Fingerprint Data Storage**
 - Fingerprint Sensor Spying Attack
 - Backdoor of Pre-embedding Fingerprints
- ❖ Discussion

What you thought your fingerprint should be...



What the reality is...

```
shell@t6wl:/ $ ls -l /data/dbgraw.bmp  
-rw-rw-rw- root_ val_fp 383306 2015-06-16 11:27 dbgraw.bmp
```



Any unprivileged processes or apps can steal user's fingerprints by reading this file.
File format is distorted -- but easy to recover.

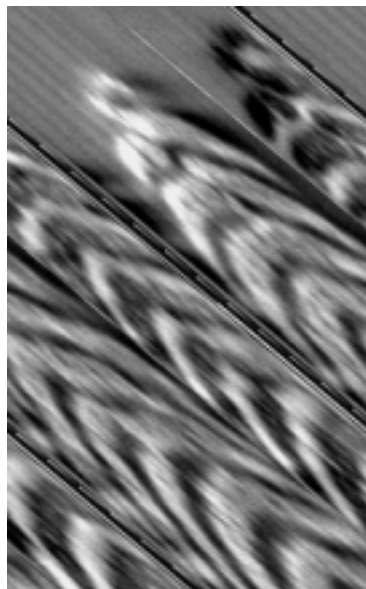
Problem found on HTC One Max. HTC has patched it by working with its vendor after our notification.

Fingerprint Image Format

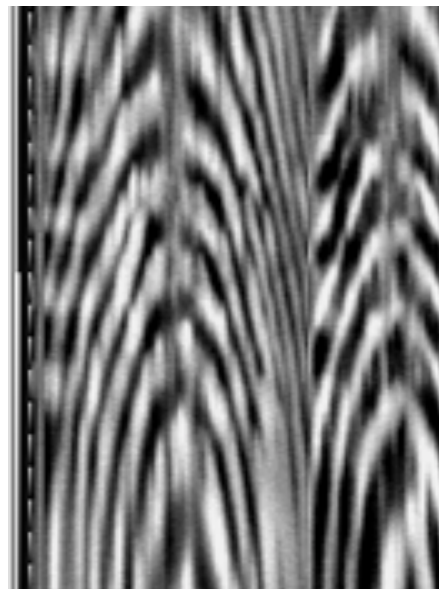
```
01 FE 02 00 09 09 14 20 60 50 70 70 70 40 60 50
70 70 70 70 70 70 70 70 60 70 60 70 70 80 70 80
80 70 70 70 70 80 90 A0 A0 A0 A0 A0 A0 B0 A0 80
[...]
B0 B0 B0 A0 B0 A0 A0 A0 A0 A0 A0 A0 90 A0 A0 90
90 90 90 90 90 90 80 80 70 B0 70
```

- It's a bitmap image
- Each line starts with 0xFE01
- Each line is not properly 4-byte aligned (can be fixed by padding)

Fingerprint Bitmap Recovery



Padding
-----▶



Then... how about fingerprints stored in TrustZone?

- ◆ TrustZone is NOT unbreakable, if vendor's code is buggy

Dan Rosenberg, QSEE TrustZone Kernel Integer Overflow, BlackHat USA 2014

Josh Thomas and Nathan Keltner, Here be Dragons, RECON Canada 2014

Di Shen, Attacking Your Trusted Core: Exploiting TrustZone on Android, BlackHat USA 2015

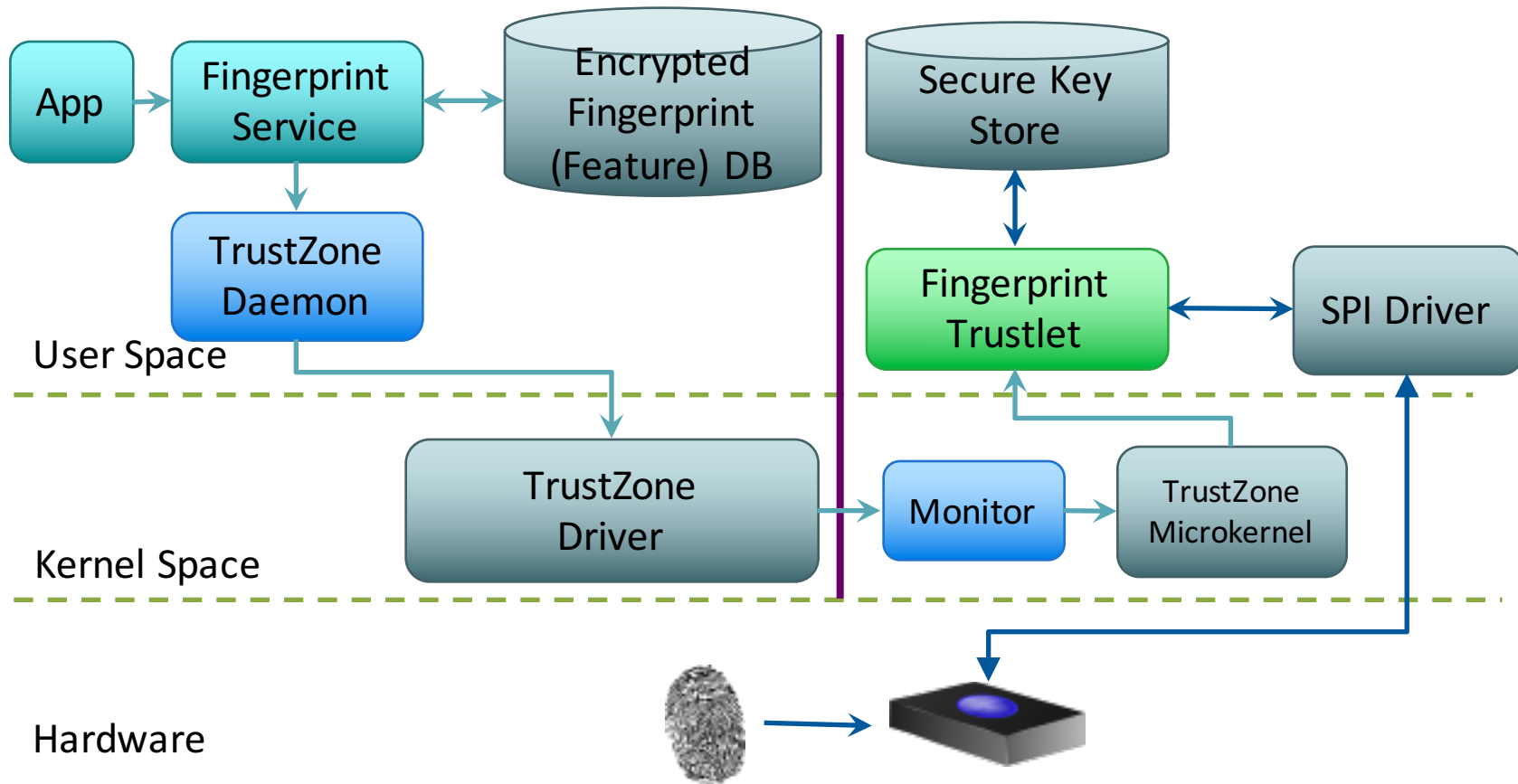


Arbitrary code execution in TrustZone

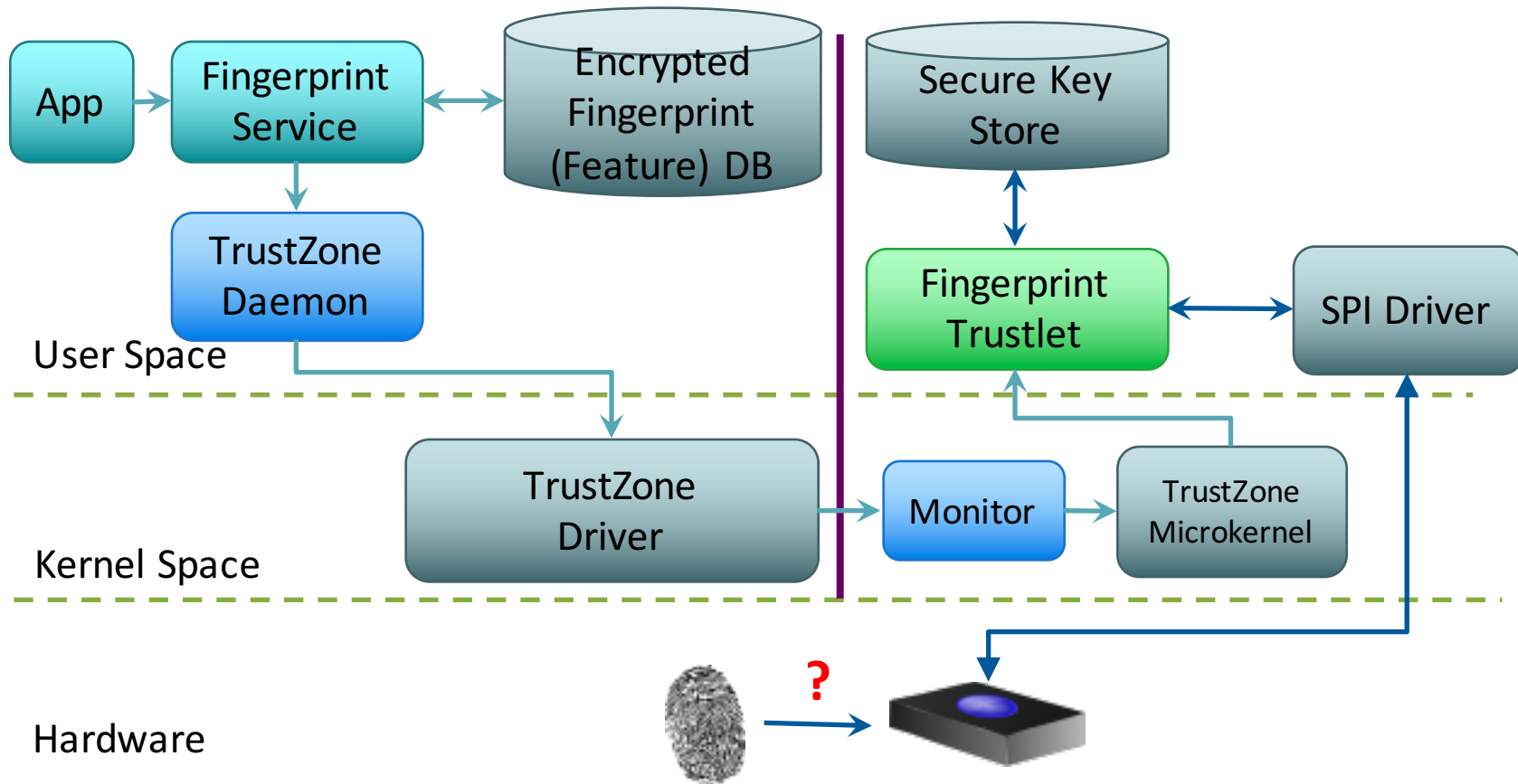
Outline

- ❖ Design of Android Fingerprint Frameworks
 - Fingerprint Recognition
 - Mobile Fingerprint Frameworks
- ❖ System Attacks against Fingerprints
 - Confused Authorization Attack
 - Unsecure Fingerprint Data Storage
 - **Fingerprint Sensor Spying Attack**
 - Backdoor of Pre-embedding Fingerprints
- ❖ Discussion

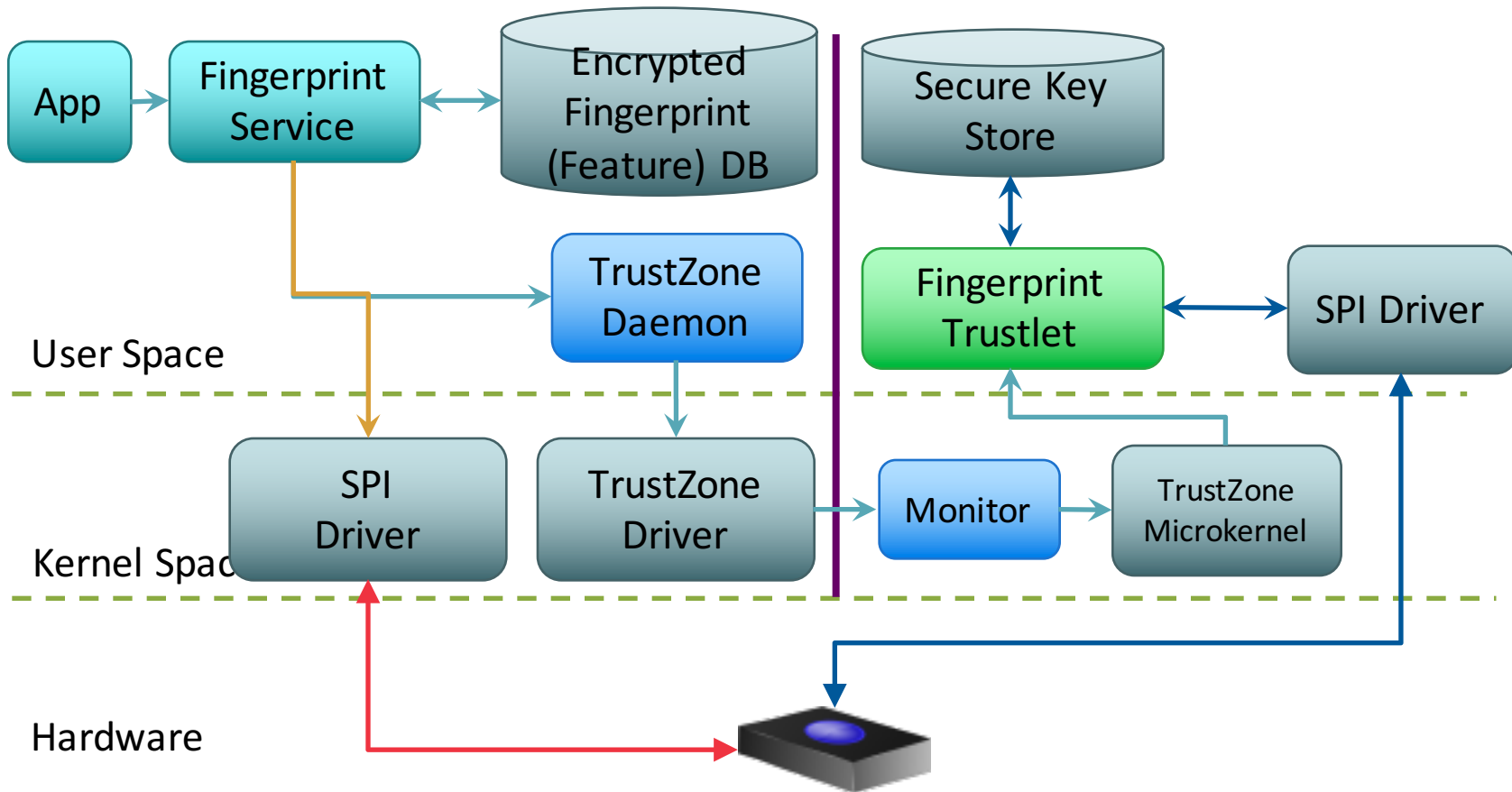
Fingerprint Authorization Framework with TrustZone



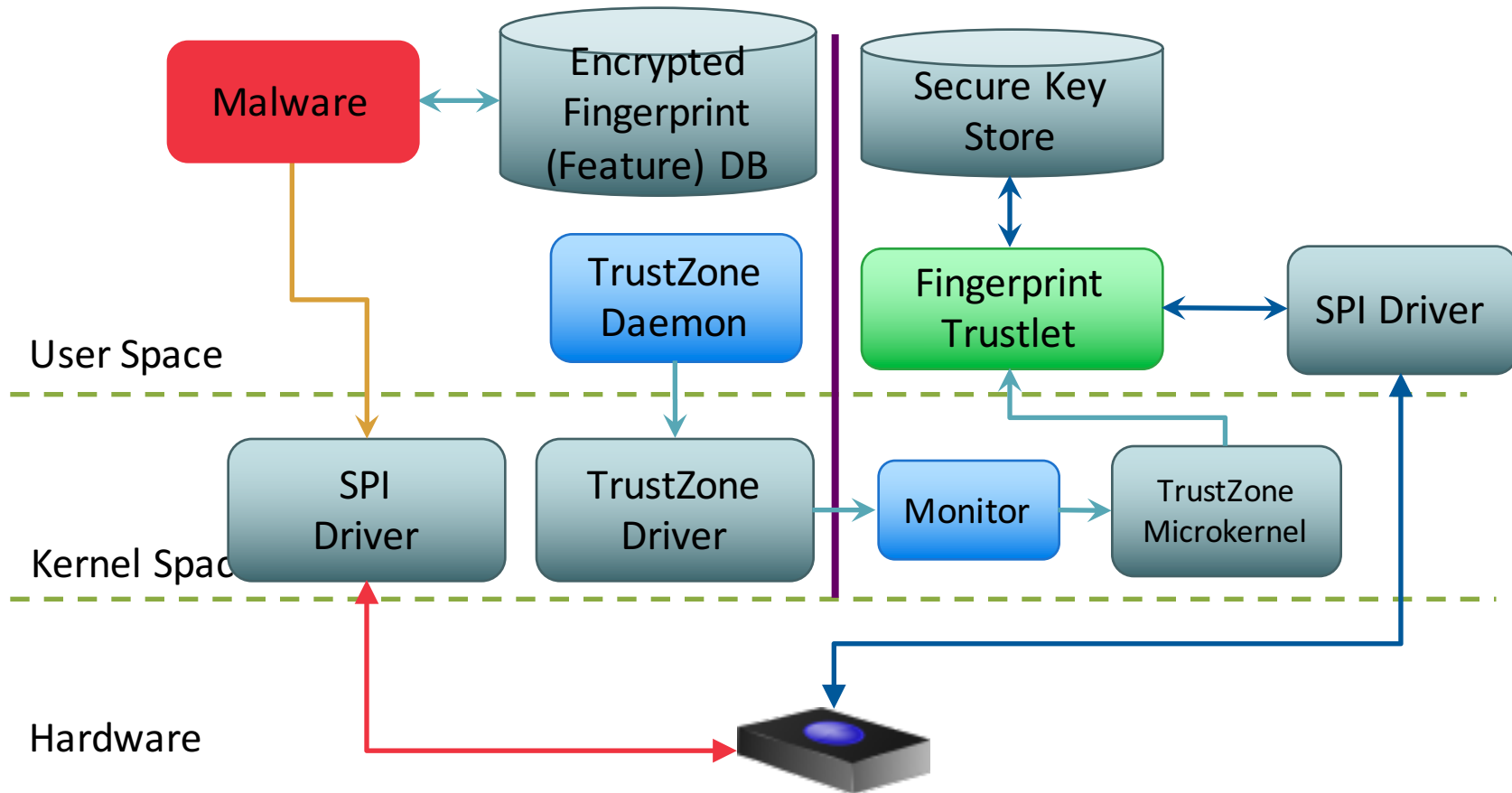
How about the isolation of fingerprint sensor devices?



Fingerprint Framework on Some Devices



No isolation & depend on access from the normal world



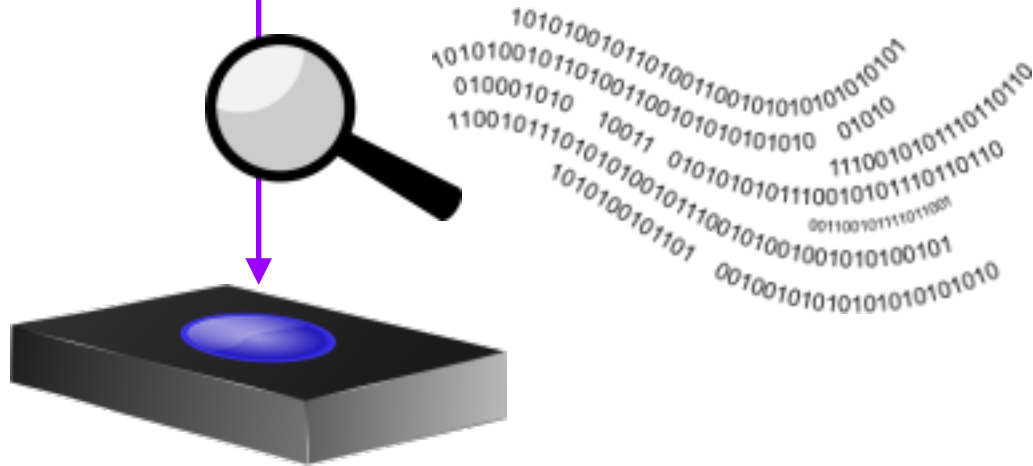
Fingerprint Sensor Operations

(Can Be Obtained from Vendors' Open-source Kernel Code)

<code>IOCTL_POWER_ON</code>	<code>IOCTL_POWER_OFF</code>
<code>IOCTL_DEVICE_RESET</code>	<code>IOCTL_SET_CLK</code>
<code>IOCTL_CHECK_DRDY</code>	<code>IOCTL_SET_DRDY_INT</code>
<code>IOCTL_REGISTER_DRDY_SIGNAL</code>	<code>IOCTL_SET_USER_DATA</code>
<code>IOCTL_GET_USER_DATA</code>	<code>IOCTL_DEVICE_SUSPEND</code>
<code>IOCTL_STREAM_READ_START</code>	<code>IOCTL_STREAM_READ_STOP</code>
<code>IOCTL_RW_SPI_MESSAGE</code>	<code>IOCTL_GET_FREQ_TABLE</code>
<code>IOCTL_DISABLE_SPI_CLOCK</code>	<code>IOCTL_SET_SPI_CONFIGURATION</code>
<code>IOCTL_RESET_SPI_CONFIGURATION</code>	<code>IOCTL_GET_SENSOR_ORIENT</code>

Sensor Communication Protocol Can Be Reversed by Hooking R/W/RW Methods

```
read(), write(), ioctl(), ...
```



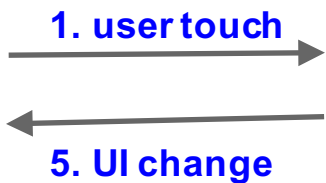
Fingerprint Sensor Spying Attack

Demo!

Fingerprint Sensor Spying Attack

- We have confirmed this vulnerability on devices including HTC One Max and Samsung Galaxy S5, etc. On Samsung devices the attacker has to root the device and load it with a carefully crafted custom ROM before leveraging the vulnerability for anything malicious.
- Both vendors have provided patches per our notification.
- It should be a general problem shared by most vendors though.

Why?



Normal world UI

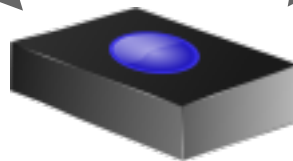


2. dev op request



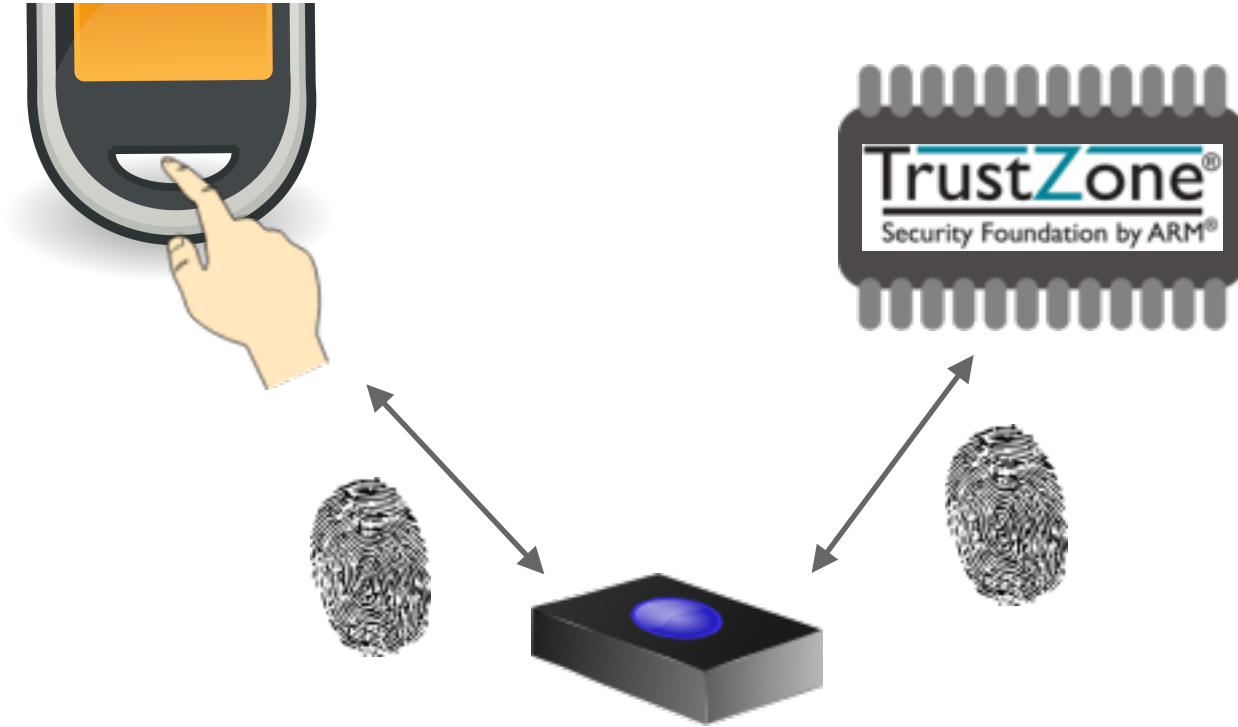
4. dev state change notification (async)

3. dev read

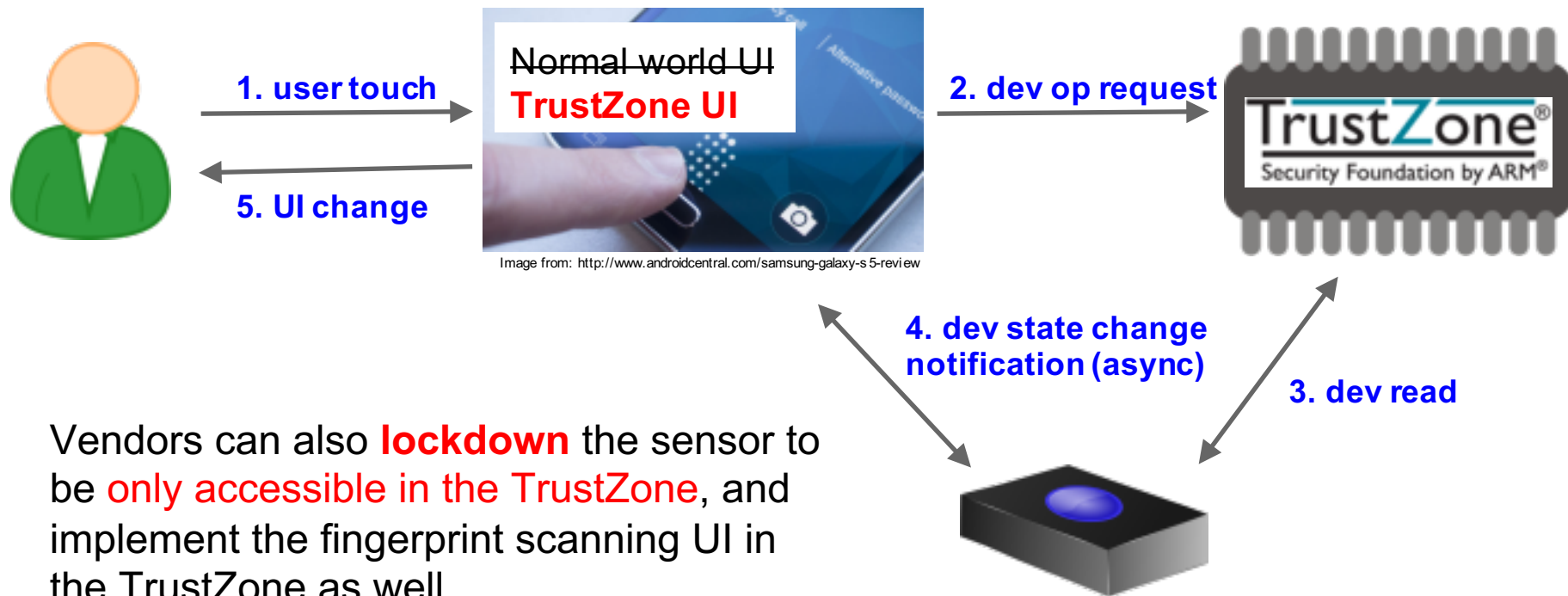


- Normal world UI needs to reflect scanning state change in real time
- So it will be easier to let it **directly control** the device (reset/enable/disable/set frequency/etc.) and receive signals from the device.

This Is **Insecure** If Fingerprint Sensor Serves Data in Plaintext

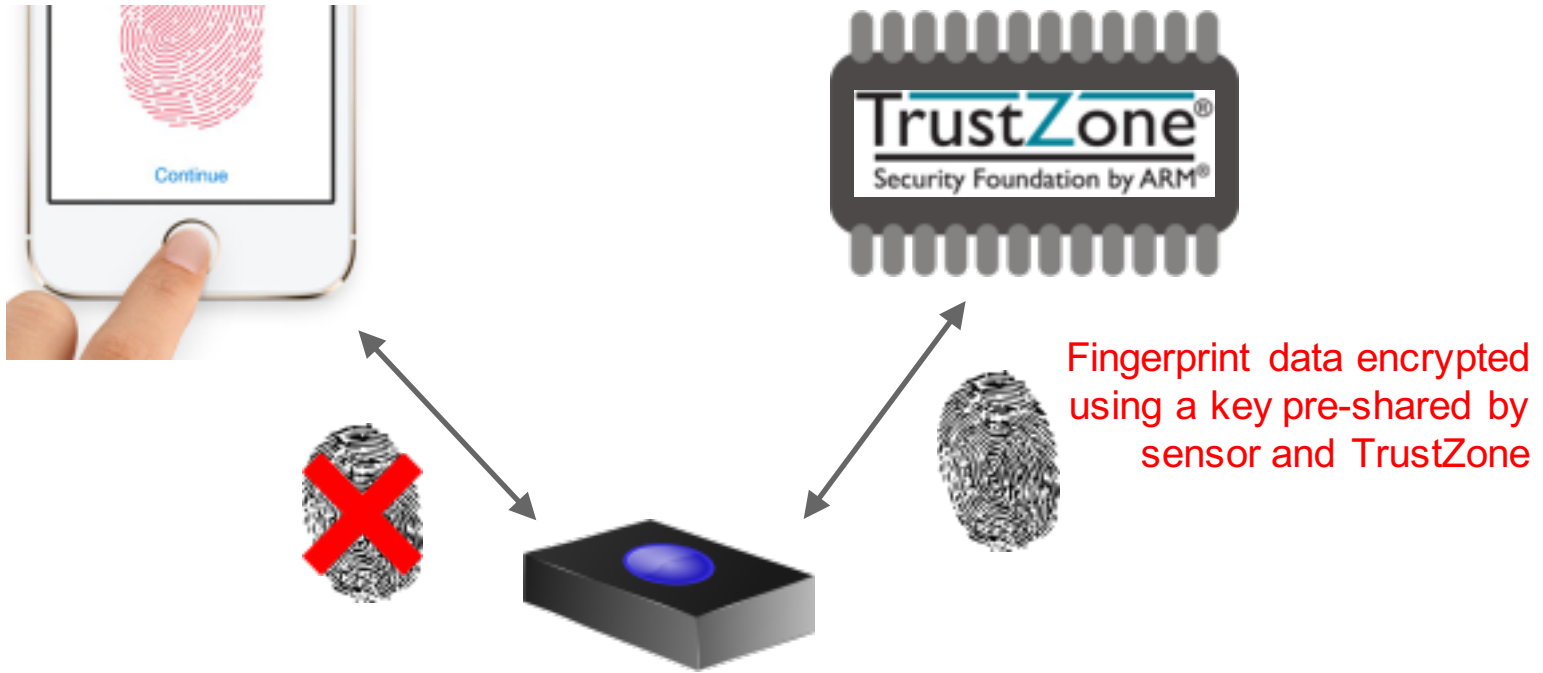


How Samsung Solves It? -- Trusted UI



Vendors can also **lockdown** the sensor to be **only accessible in the TrustZone**, and implement the fingerprint scanning UI in the TrustZone as well.

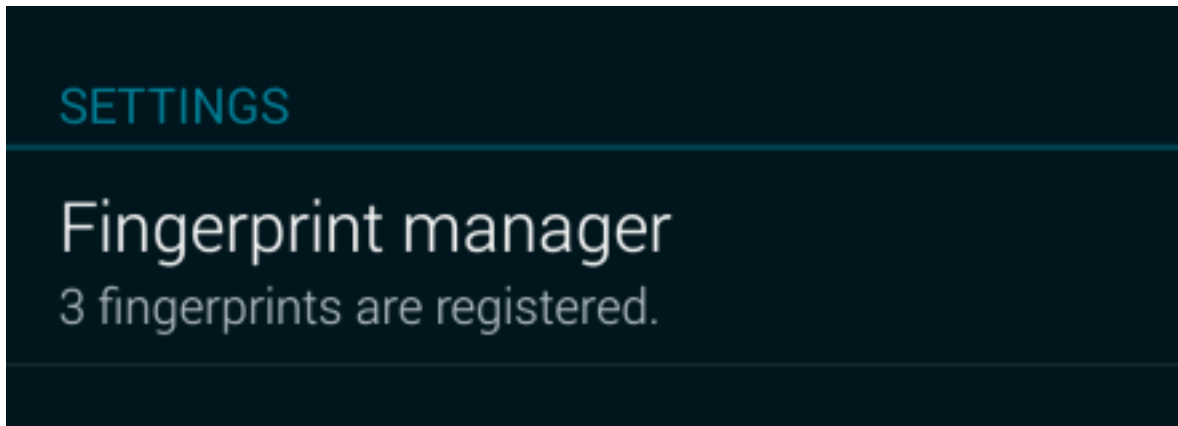
How Apple Solves It



Outline

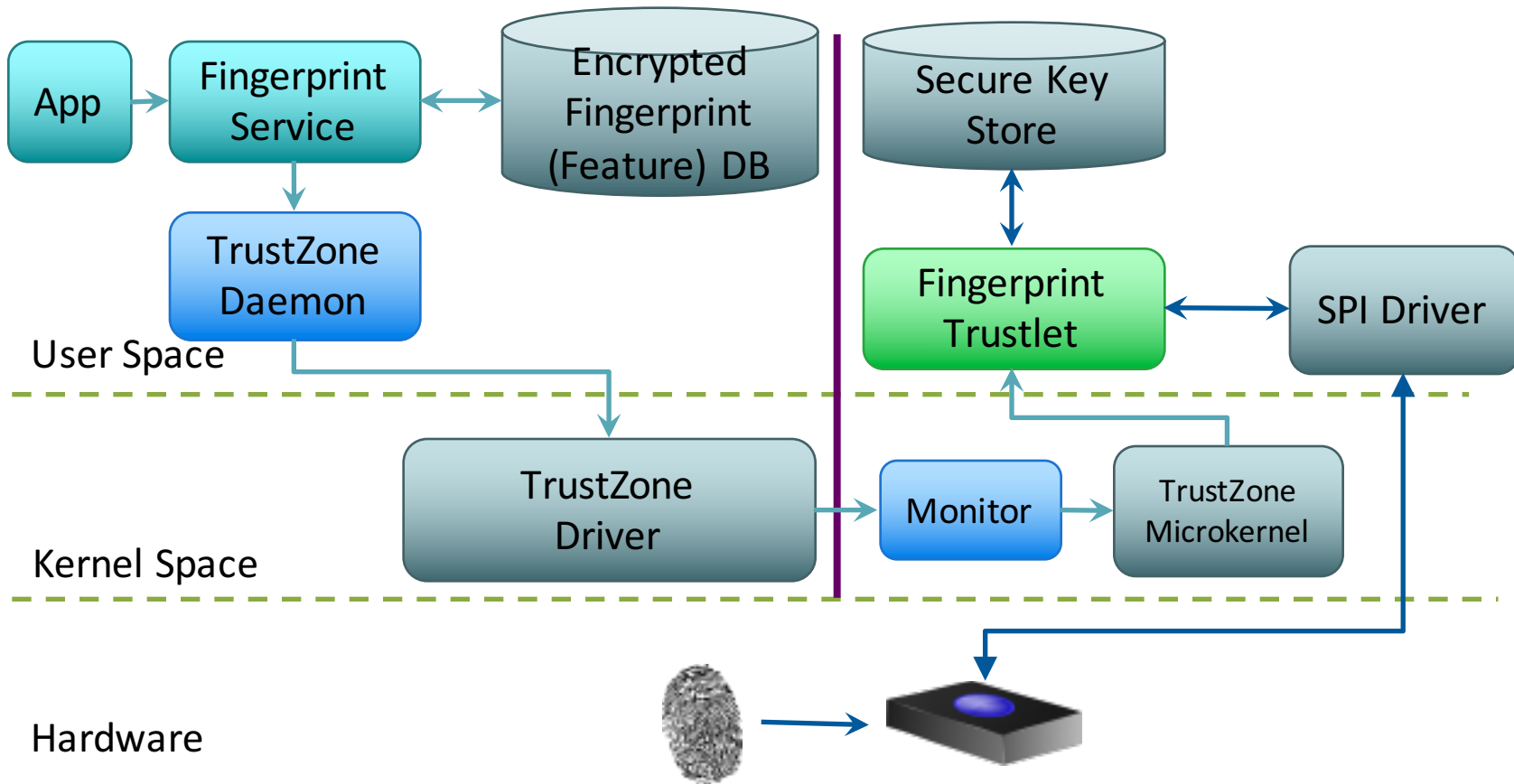
- ❖ Design of Android Fingerprint Frameworks
 - Fingerprint Recognition
 - Mobile Fingerprint Frameworks
- ❖ System Attacks against Fingerprints
 - Confused Authorization Attack
 - Unsecure Fingerprint Data Storage
 - Fingerprint Sensor Spying Attack
 - **Backdoor of Pre-embedding Fingerprints**
- ❖ Discussion

Fingerprint Settings

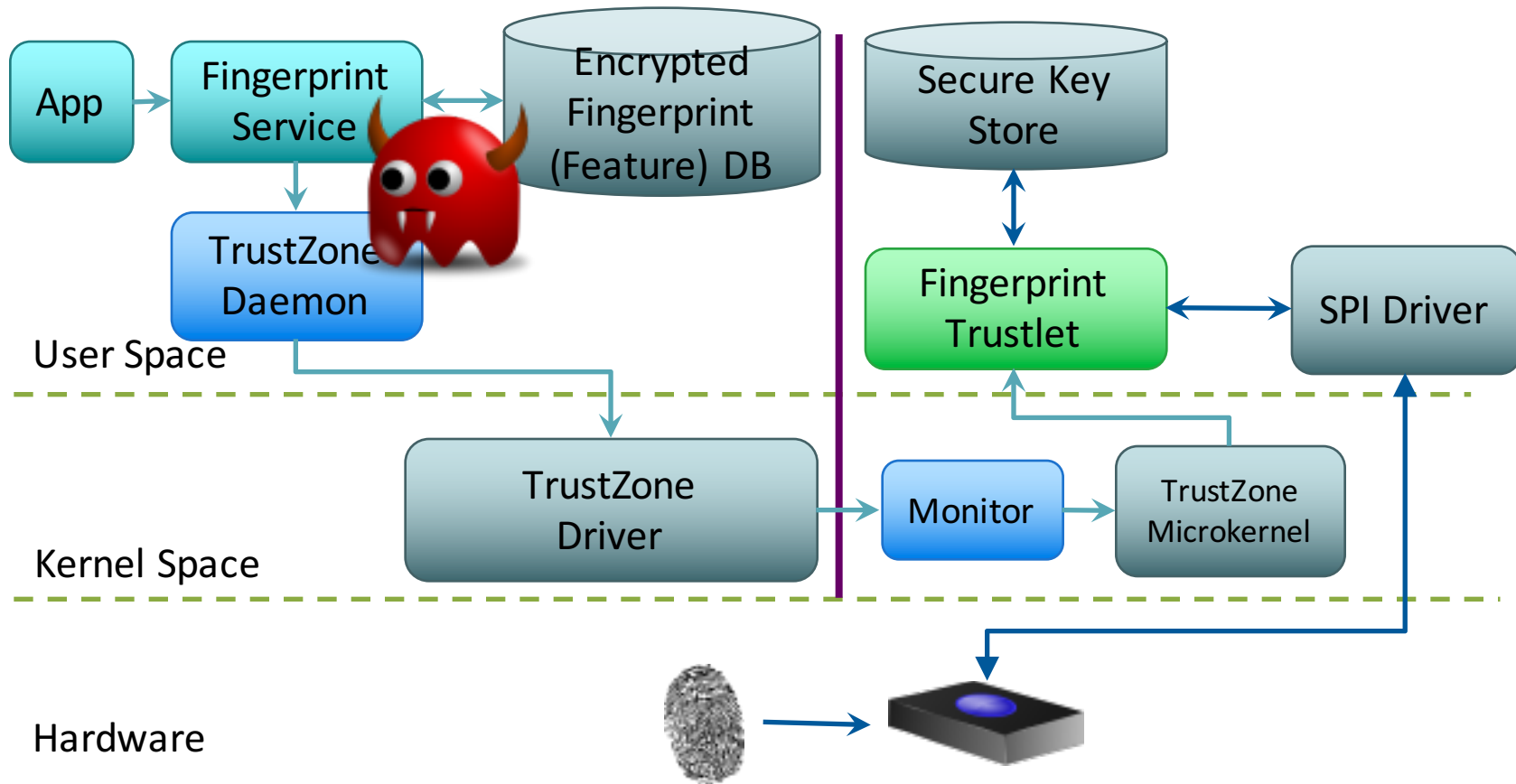


- ◆ How can you attest that only 3 fingerprints were registered?

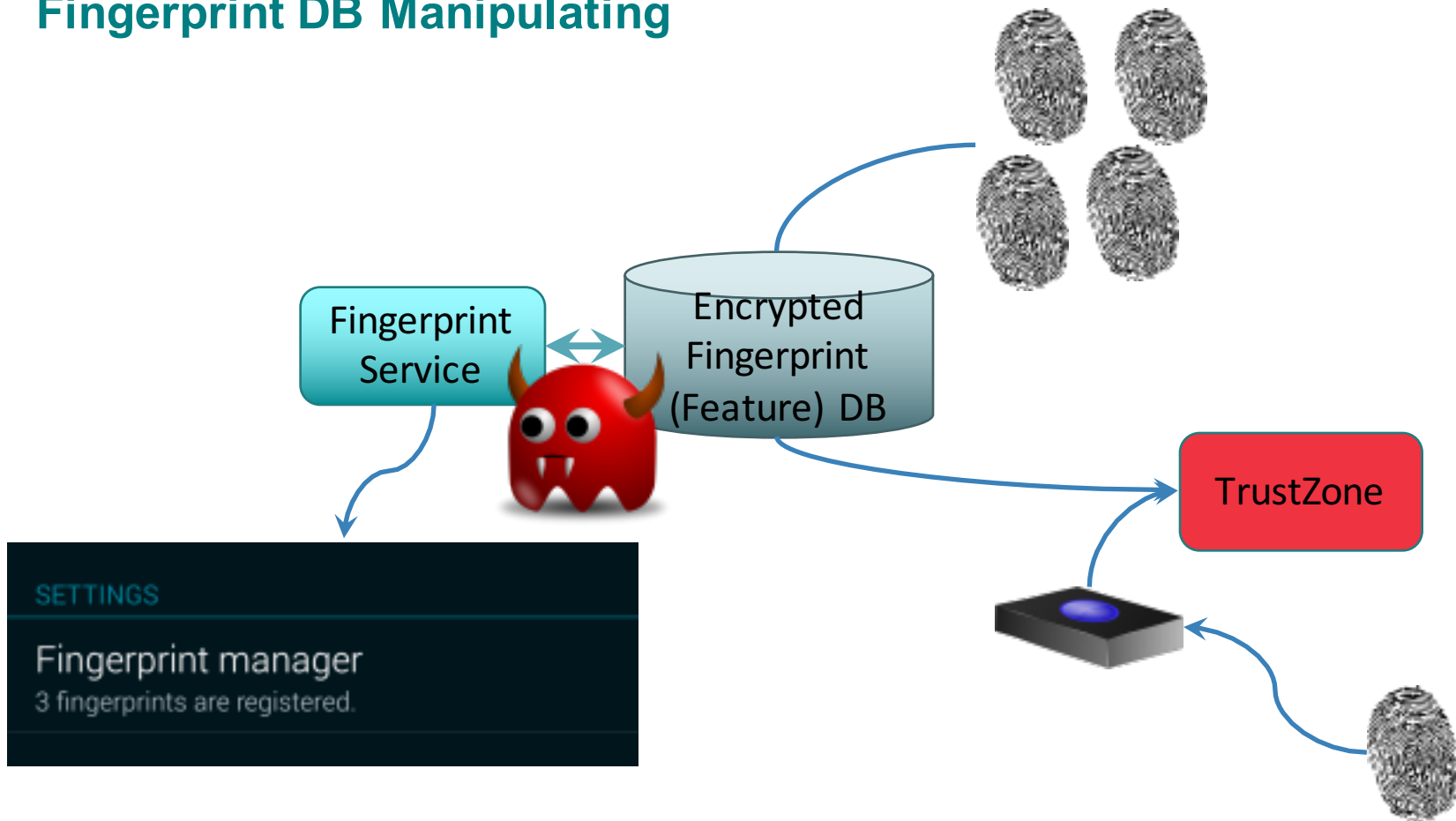
Fingerprint Authorization Framework with TrustZone



Fingerprint Authorization Framework with TrustZone



Fingerprint DB Manipulating



Fingerprint Backdoor

- ◆ TrustZone just scans a fingerprint and matches it against encrypted fingerprints fed from the normal world
 - ◆ It knows nothing about the number of fingerprints stored by the normal world
- ◆ An attacker can tamper the normal world framework to stealthily pre-embed special fingerprint blob (maybe fake)
 - ◆ So he/she can unlock the device or authorize other operations
 - ◆ Leave no explicit traces

Fingerprint Backdoor

It is usually the Settings app that displays the registered fingerprint number to the users.

- ◆ For example, on some devices, attacker with root privilege can modify the `enrolledFingerprintNum` method of the class `com/android/settings/fingerprint/FingerprintSettings` in `SecSettings.apk`.
- ◆ He/she can change the return value of `getEnrolledFingers` to be $n-m$, where n is the actual registered fingerprint number and m is the number of fingerprints pre-embedded by the attacker.

Fingerprint Backdoor

- ◆ Note that replacement of the Settings app (a system app) requires disabling the system signature checking.
- ◆ Most devices enforce the system signature checking based on the `compareSignatures` method in the class `com/android/server/pm/PackageManagerService` implemented in `/system/framework/services.jar`. It will return zero if signature match, and non-zero otherwise.
- ◆ Therefore, one can modify this method to always return zero, so that the system signature checking will always success.

Fingerprint Backdoor

Demo!

Outline

- ❖ Design of Android Fingerprint Frameworks
 - Fingerprint Recognition
 - Mobile Fingerprint Frameworks
- ❖ System Attacks against Fingerprints
 - Confused Authorization Attack
 - Unsecure Fingerprint Data Storage
 - Fingerprint Sensor Spying Attack
 - Backdoor of Pre-embedding Fingerprints
- ❖ **Discussion**

Key Takeaways

- ◆ Mobile devices with fingerprint sensors are more and more popular
- ◆ But they still have severe security challenges, such as
 - ◆ Confused Authorization Attack
 - ◆ Unsecure Fingerprint Data Storage
 - ◆ Fingerprint Sensor Spying Attack
 - ◆ Backdoor of Pre-embedding Fingerprints
- ◆ Such security flaws can lead fingerprint leakages
- ◆ Industry should pay more attention to audit existing design and implementations of fingerprint frameworks

Suggestions to Mobile Users

- ◆ Stick to mobile device vendors with timely patching/upgrading to the latest version (e.g. Android Lollipop), and always keep your device up to date
- ◆ Always install popular apps from reliable sources
- ◆ Enterprise/government users should seek for professional services to get protections against advanced targeted attacks
- ◆ To provide a better level of protection the end-user should NOT root their device if unnecessary, rooting a device will exploit a device to unknown risks

Suggestions to Mobile Vendors

- ◆ Mobile device vendors should improve the security design of the fingerprint auth framework
 - ◆ Improved recognition algorithm against fake fingerprint attacks
 - ◆ Better protection of both fingerprint data and the devices
 - ◆ Differentiating authorization with authentication
- ◆ The existing fingerprint auth standard should be further improved to provide more detailed and secured guidelines for developers to follow
- ◆ Given a security standard, vendors still need professional security vetting/audits to enforce secure implementations

Further Suggestions

- Actually all the four vulnerabilities/attacks described here are commonly applicable to ALL the fingerprint based authentication/authorization platforms.
- For example, many high-end laptops equip fingerprint scanners to authenticate and authorize user login.



Image from: <http://www.bootic.com/lenovo/electronics/computers/laptops/lenovo-a-3000-n200>

Further Suggestions (Cont.)

- For external fingerprint scanners used for identity recognition (e.g. in the custom house, immigration office, and the DMV), door access control, or money transaction in banks, the situation is similar.
- So we suggest that the fingerprint auth framework for ALL platforms should also be improved to better protect fingerprint data and sensor (and provide defense of any other attacks described in this paper if applicable).

Q & A

For more details, please refer to our whitepaper:

Fingerprints On Mobile Devices: Abusing And Leaking

Y. Zhang, Z. Chen, H. Xue, and T. Wei