android Bootcamp 2016

Android Keystore Attestation

...and other new hawtness

January 21, 2016

Agenda

Hardware-backed KeyStore overview

Attestation

Other KeyStore enhancements

Bootloader changes

CDD requirements

Hardware-backed KeyStore overview

Hardware-backed KeyStore

Keep keys safe

Software-only keystore allows apps to keep critical keys out of their process space. Hardware-backing allows the keys to be kept away from even the Linux kernel, so Android vulnerabilities cannot leak them.

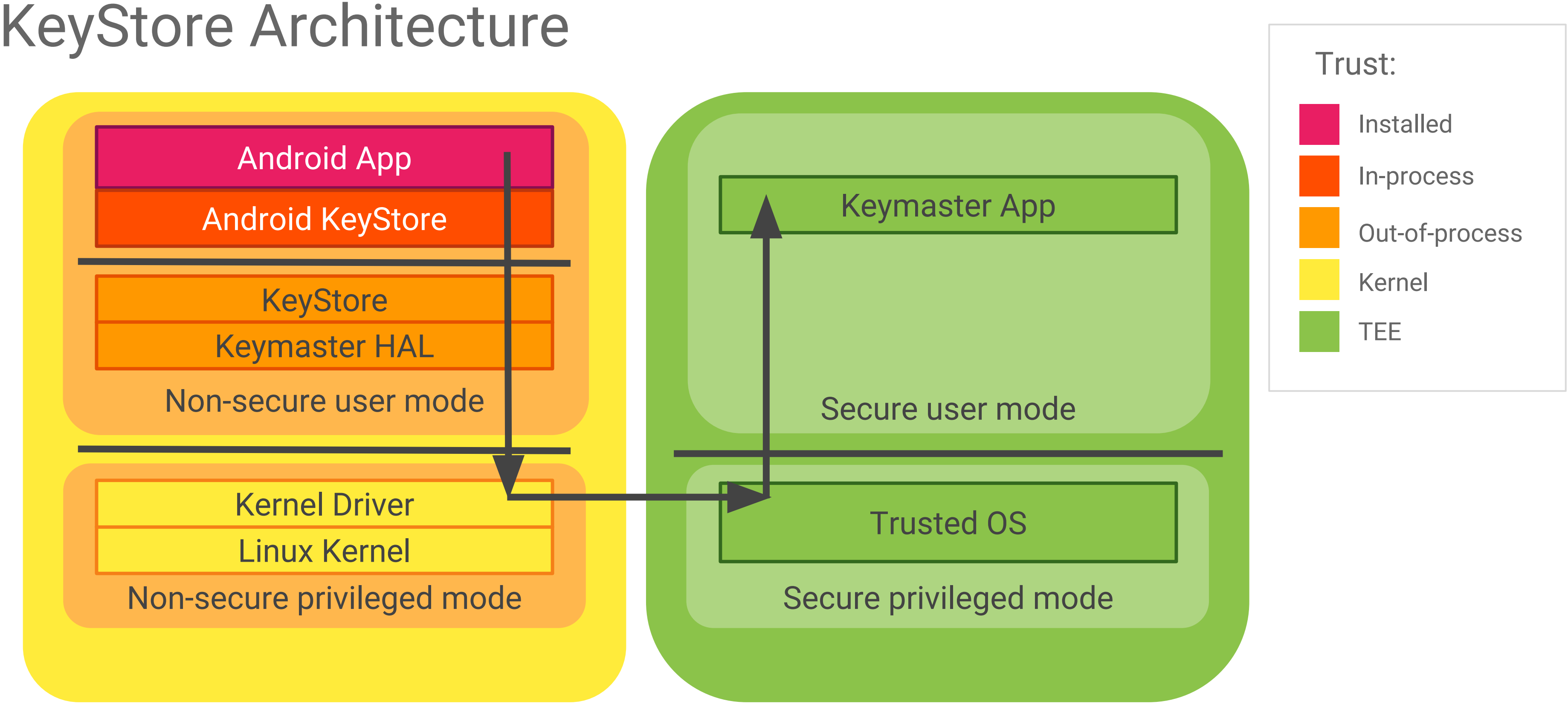
Control key usage

Compromise of the Android system doesn’t provide access to key material, but it could allow attackers unlimited use of the keys. Android Keystore enables key usage to be limited, e.g. access-controlled by user authentication.

Provide adequate toolset

Hardware-backed cryptography is useless if it doesn’t provide the tools that developers need. Android KeyStore provides a core suite of cryptographic algorithms and tools that covers the vast majority of applications.

KeyStore Architecture



Key Attestation

What problem are we trying to solve?

There’s no way to know if keys are in secure hardware.

- The Keymaster HAL could lie.
- The framework could be compromised and misreport HAL response.
- Even if the device is good, apps can lie to off-device parties.

Solution: Have secure hardware produce an “attestation certificate”.

- The certificate describes the key, including its access controls.
- The certificate is signed by an attestation key injected at the factory.
- No runtime dependency on Google.

Attestation details

- Attestation can be applied to RSA or EC keys.
- Attestations are produced in the form of an X.509 certificate.
  - Constraints and validity periods of certs will mirror the values specified for the keys.
  - A custom extension will contain the details of the access control constraints applied, and indicate which are applied in hardware.
  - The signing key will be ECDSA (NIST P-384), with provision to switch to RSA (3072 bits) with a security OTA.
- Attestation keys (ECDSA and RSA) will be provisioned in the factory.
- Google will provide the CA root and will certify attestation keys.

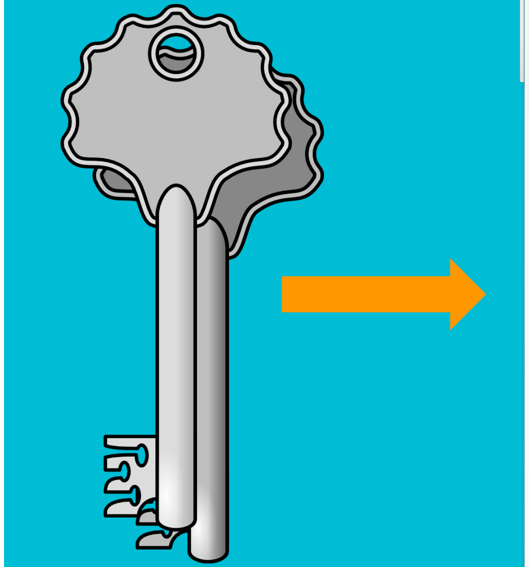
FIDO and Android KeyStore Attestation

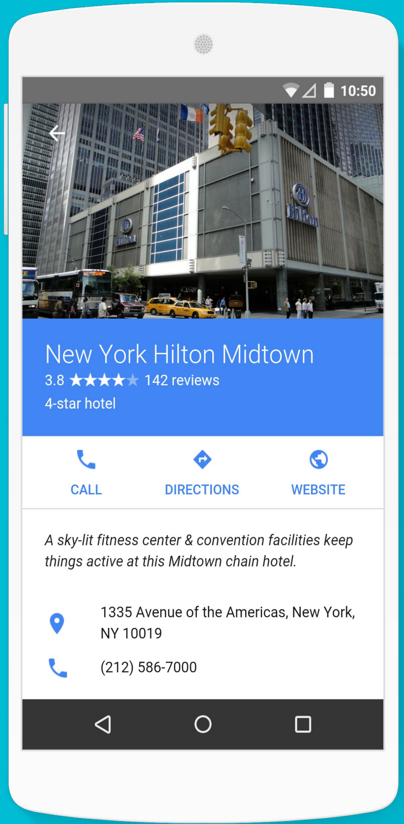
KeyStore attestation is similar to—but not an implementation of—FIDO U2F.

- U2F uses other data structures and protocols that are too specific for a general-purpose crypto toolkit.
- KeyStore attestation *does* provide all of the security properties desired by FIDO relying parties.
- Google will work with FIDO to reconcile the issues.
- Bottom line: **FIDO relying parties will be able to use KeyStore.** This is expected to drive widespread use of KeyStore.

Key Provisioning

- Google will certify attestation keys for **Google-approved** devices.
- Keys will be deployed to device batches: min 10K devices per key.
- Initially, Google will create the keys as well as certify them.
- The process will be very similar to the Widevine key distribution process (will likely use the same delivery method).





10:50

New York Hilton Midtown

3.8 ★★★★☆ 142 reviews

4-star hotel

A sky-lit fitness center & convention facilities keep things active at this Midtown chain hotel.

1335 Avenue of the Americas, New York, NY 10019

(212) 586-7000

Key Revocation

App and server trust depends on attestation key secrecy.

Compromised keys will be revoked via CRL and OSCP:

- Secure key injection can only be done at the factory, so a device with a revoked key will become permanently untrusted.
- Keys are injected into device batches, so revocation affects an entire batch, at minimum.
- Revocations will be applied as broadly as needed, depending on the nature and extent of the leak.

Other KeyStore enhancements

Other KeyStore enhancements

- More elliptic curve functionality
  - ECIES
  - ECDH
- Exportable symmetric keys
- Fingerprint-bound keys that are not revoked on fingerprint enrollment
- OS version binding to protect against OS rollback

Bootloader changes

android

Bootloader changes

New hardware keystore features require some bootloader features:

- Bootloader must provide OS version and patch level to TEE.
- Bootloader must provide Verified Boot public key and lock status to TEE.

CDD requirements

android

CDD requirements

For Android Marshmallow, hardware-backed keystore was STRONGLY RECOMMENDED.

Hardware-backed keystore will be MANDATORY in a future release.

- All algorithms (RSA, AES, ECDSA, ECDH, ECIES, HMAC)
- All hash functions (MD5, SHA1, SHA-2 family)
- Hardware Gatekeeper (on devices with lockscreens)
  - With brute force protection in hardware
- Hardware attestation support

THANK YOU