Putting It (almost) all Together: iOS Security

Konstantin Beznosov

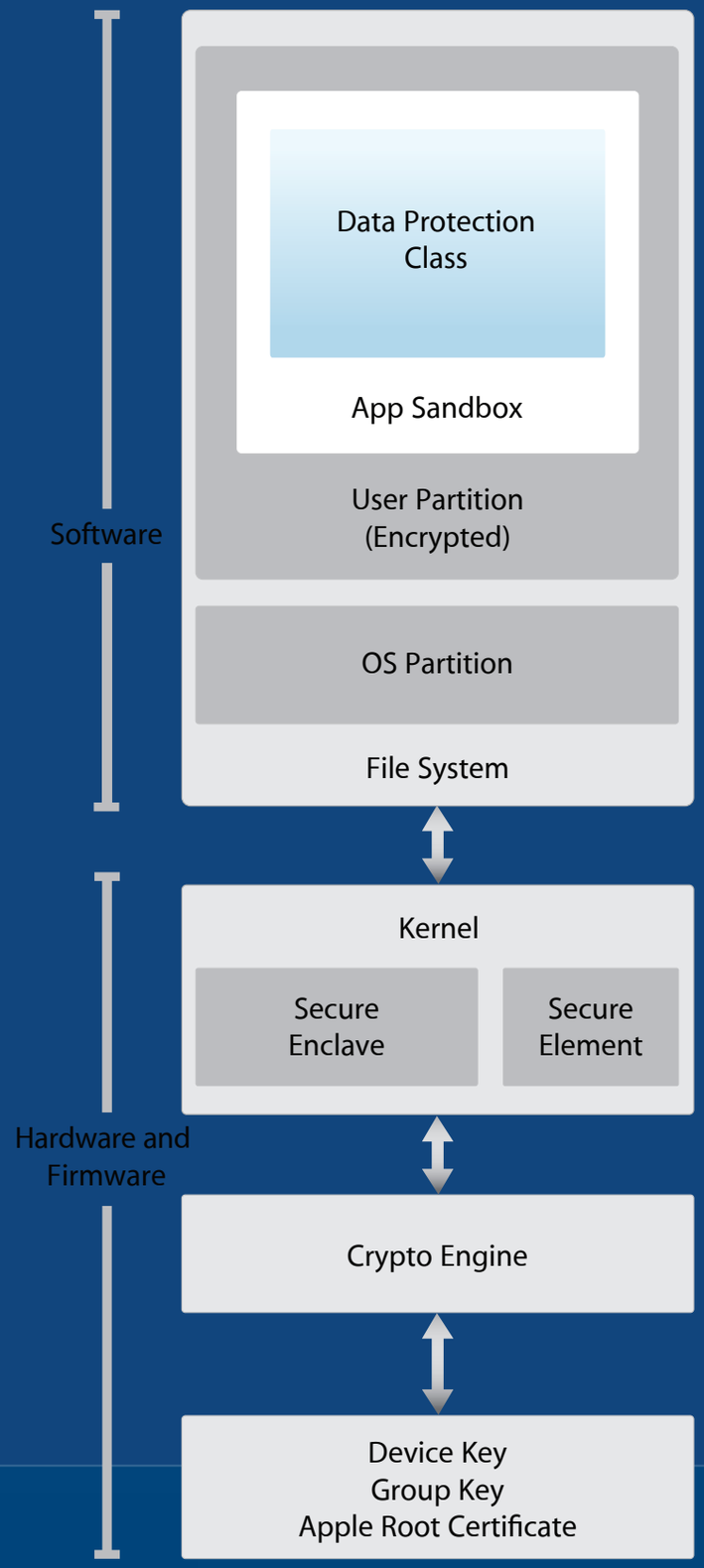
a place of mind
THE UNIVERSITY OF BRITISH COLUMBIA

ece Electrical and Computer Engineering

- BSD based OS
- Chain of trust during boot
- Secure Enclave
- Effaceable Storage (Secure deletion)
- Touch Id (Usable authentication)
- Per file encryption, chain of keys

- PBKDF2 tangled with Device ID and iteration count is calibrated to take 80ms
- application sandboxing
- rigorous vetoing process for iOS app developers

overall stack



source: “iOS Security” Apple, September 2015

boot chain ensures integrity, and proceeds only after verifying the chain of trust

1. processor executes **Boot ROM**
   - immutable
   - contains Apple Root CA public key
   - hardware root of trust — implicitly trusted
2. Boot ROM verifies that **Lowe-Level Bootloader (LLB)** is signed by Apple
3. LLB verifies signature of and runs **iBoot**
4. iBoot verifies signature of and runs **iOS kernel**

- on devices with cellular access

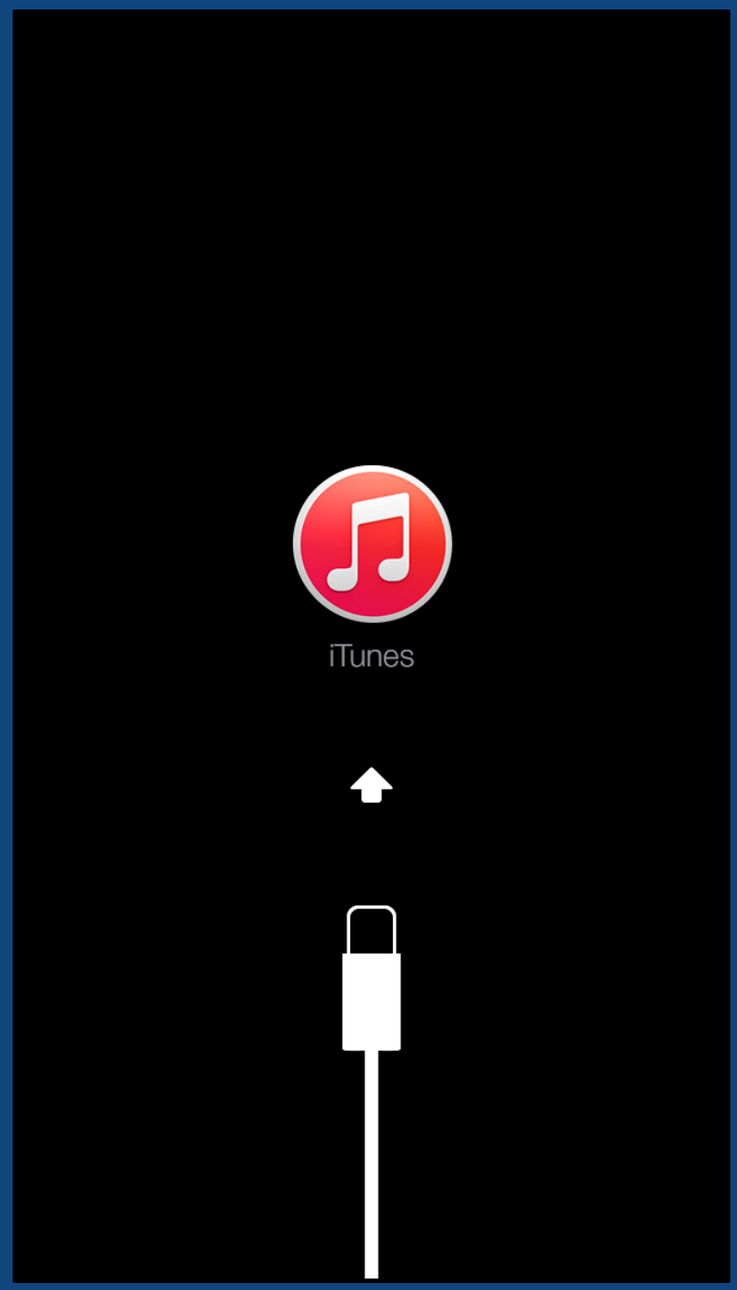
  baseband subsystem boots similarly

- on devices with A7 or later processor

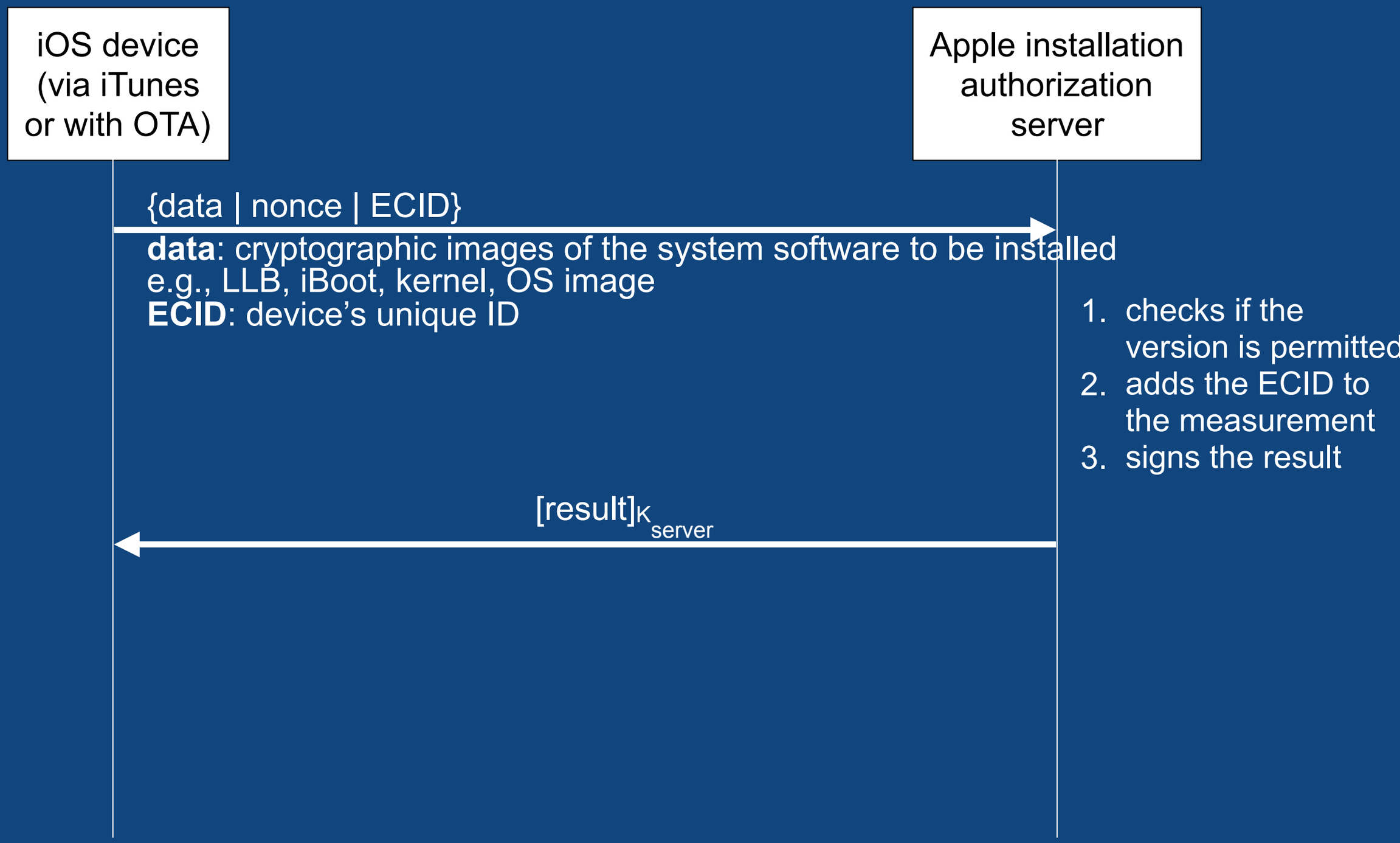
  Secure Enclave co-processor goes through similar boot process

in case of failure to boot securely

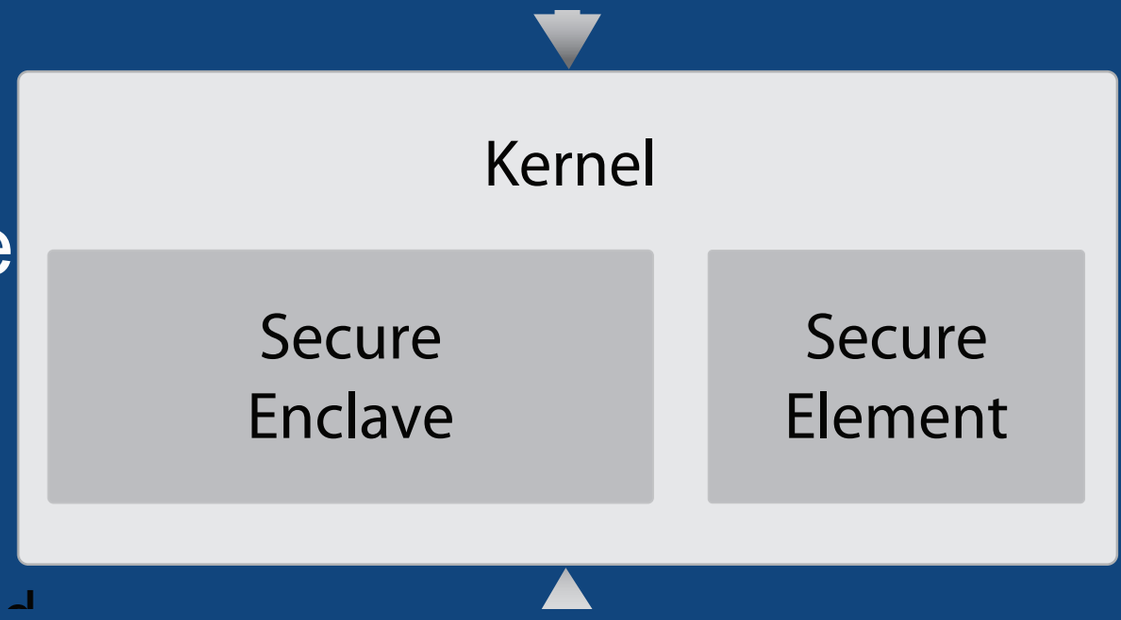
- recovery mode
  - “Connect to iTunes”
- if boot ROM unable to load
  - Device Firmware Upgrade (DFU) mode



system software authorization



secure enclave



- co-processor in A7 or later
- its own boot and software update
- all crypto for data protection
- random number generator
- encrypted memory
- shared memory data buffer and mailbox for communications with the kernel
- UID per enclave instance
- data stored on disk is encrypted with UID-entangled key
- processes fingerprint data from TouchID
  - AES-CCM encryption of the data exchanged
  - session key establishment

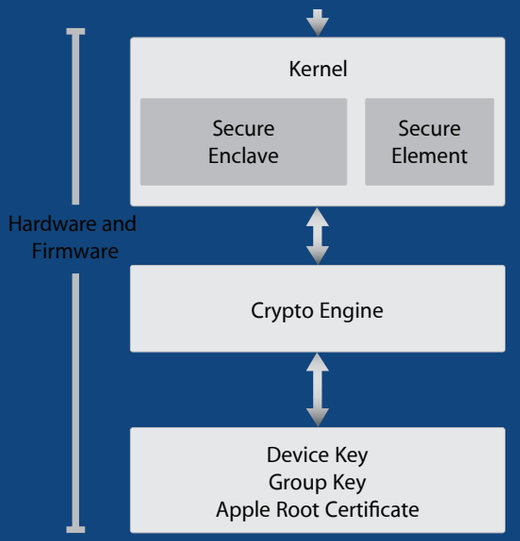

9:41 AM
100%
Fingerprints
Place Your Finger
Lift and rest the edge of your finger on the Home button repeatedly.

Touch ID

TouchID — fingerprint sensing system

- active when home button ring detects the touch of a finger
  - advanced imaging array scans the finger
  - sends to Secure Enclave
  - raster scan is temporarily stored in encrypted memory within the Secure Enclave
  - analysis utilizes “sub-dermal ridge flow angle mapping, which is a lossy process that discards minutia data that would be required to reconstruct the user’s actual fingerprint”
  - never sent to Apple or backed up to iCloud or iTunes

crypto engine



- AES 256 crypto engine built into the DMA path between the flash storage and main system memory
- file encryption highly efficient.
- SHA-1 is implemented in hardware.
- UID and a device group ID (GID) are AES 256-bit keys fused into the application processor during manufacturing.
  - No software or firmware can read them directly;
  - can see only the results of encryption or decryption performed using them.

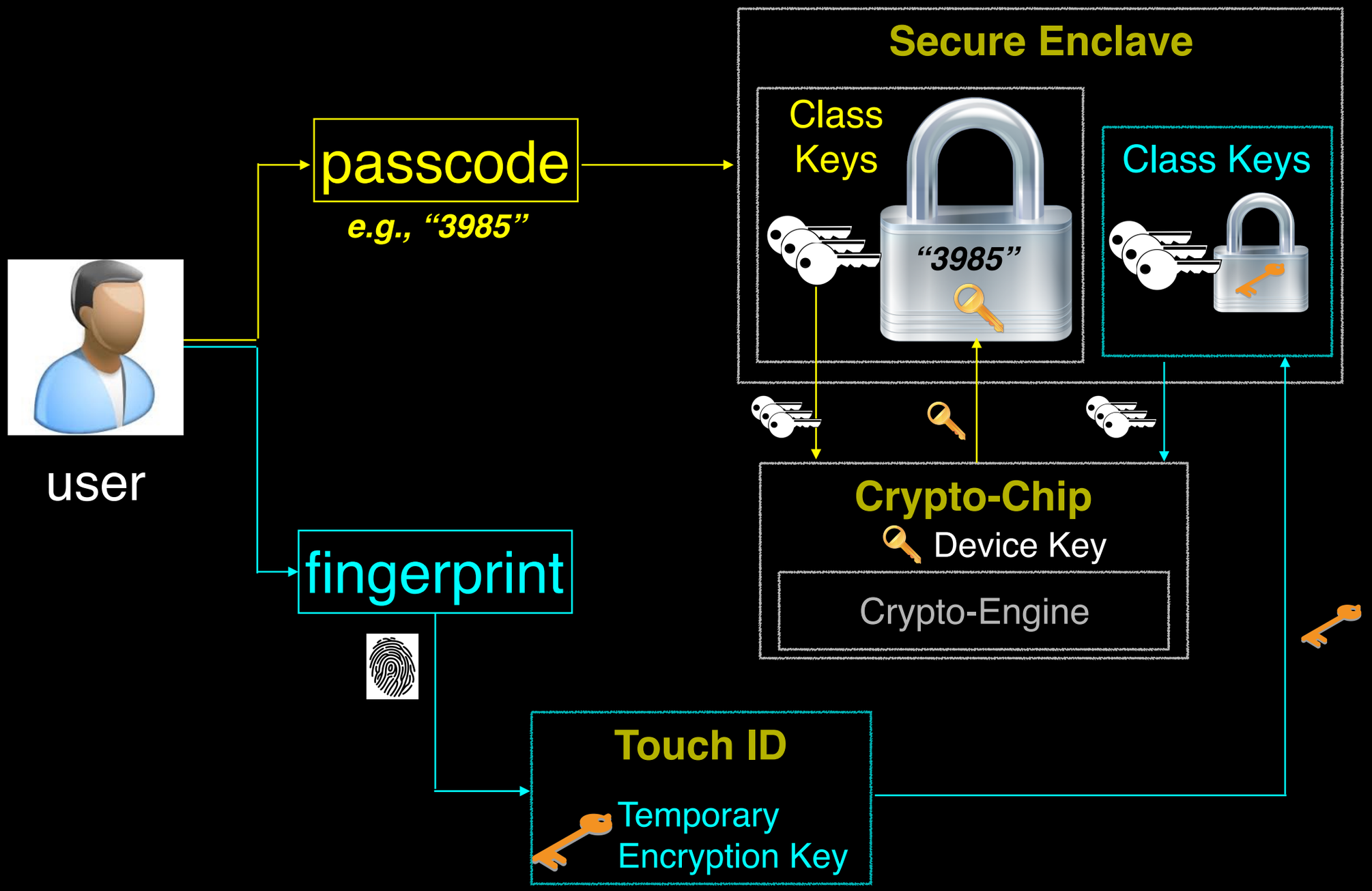
file data protection — class keys

- Complete Protection
  - protected with a key derived from the user passcode and the device UID.
  - after the user locks a device, the decrypted class key is discarded.
- Protected Unless Open
  - After the device is unlocked, your app can open and use the file.
  - If the user has a file open and locks the device (e.g., by pressing the sleep button), the app can continue to access the file.
- Protected Until First User Authentication
  - the same as Complete Protection, except that the decrypted class key is not removed from memory when the device is locked.
- No Protection
  - protected only with the UID, and is kept in Effaceable Storage.

keychain

- for sensitive data like passwords and keys
- encrypted container that holds passwords for multiple applications and secure services.
- keychain items can only be shared between apps from the same developer.
- each app always has access to its own keychain items
- the user is never asked to unlock the keychain.
- app can access only its own keychain items.
- protected using a class structure similar to file data protection.

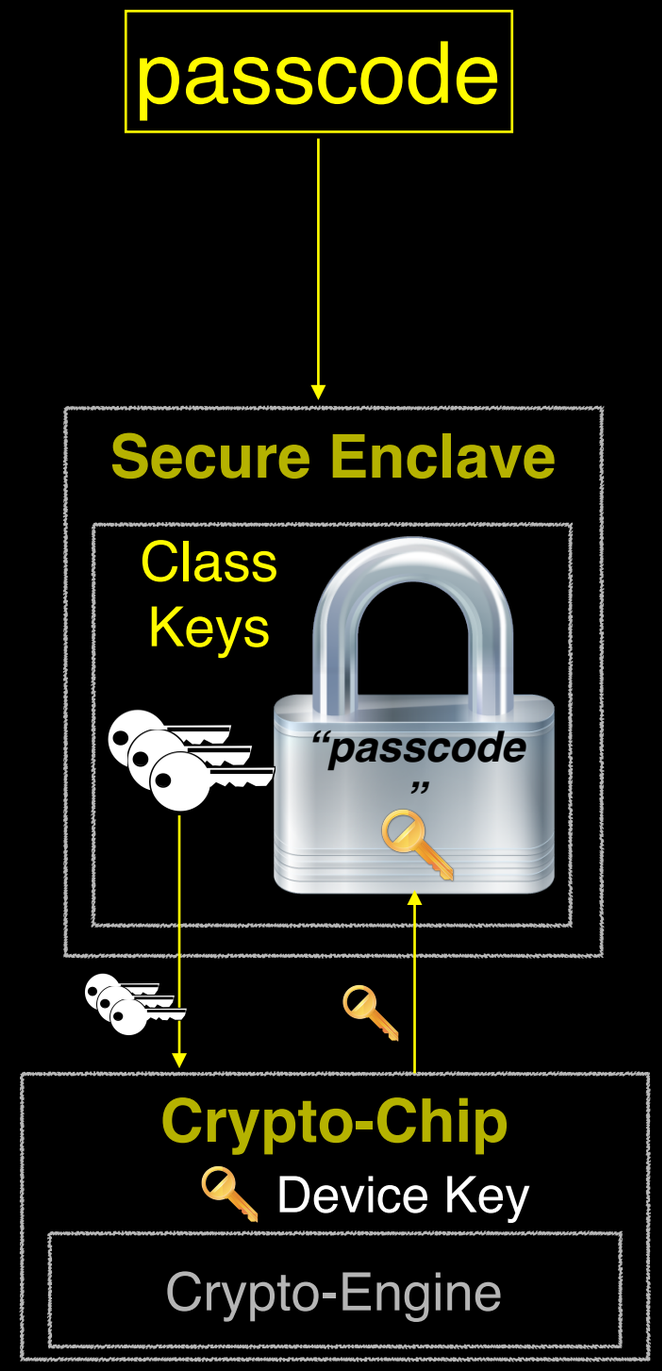
how unlock works



why passcode strength matters

on-device guessing attack

- 80 ms/guess
- requires exploitable bug in the boot-chain
- LimeRa1n for iPhone 3GS, 4
- Blackra1n, October 2009
- Limera1n/greenpois0n, October 2010
- exploit by iH8sn0w, February 2014



keybags

keep keys for both file and keychain data protection

- **System keybag**: the wrapped class keys used in normal operation of the device.
- **Backup keybag**: created when an encrypted backup is made by iTunes and stored on the computer to which the device is backed up.
- **Escrow keybag**: for iTunes syncing and Mobile Device Management (MDM).
  - allows iTunes to back up and sync without requiring the user to enter a passcode, and it allows an MDM server to remotely clear a user's passcode.
- **iCloud Backup keybag**: similar to the Backup keybag.
  - All the class keys in this keybag are asymmetric (using Elliptic Curve public key crypto with Curve25519, like the Protected Unless Open Data Protection class), so iCloud backups can be performed in the background.

app security assurance

- all executable code must be signed using an Apple-issued certificate.
- Third-party apps must also be validated and signed using an Apple-issued certificate.
- iOS App Developer
  - must register with Apple and join the iOS Developer Program.
  - has a real-world identity, Apple verifies the identity and gives them the certificate.
  - signs apps and submits program to the App Store with the cert.

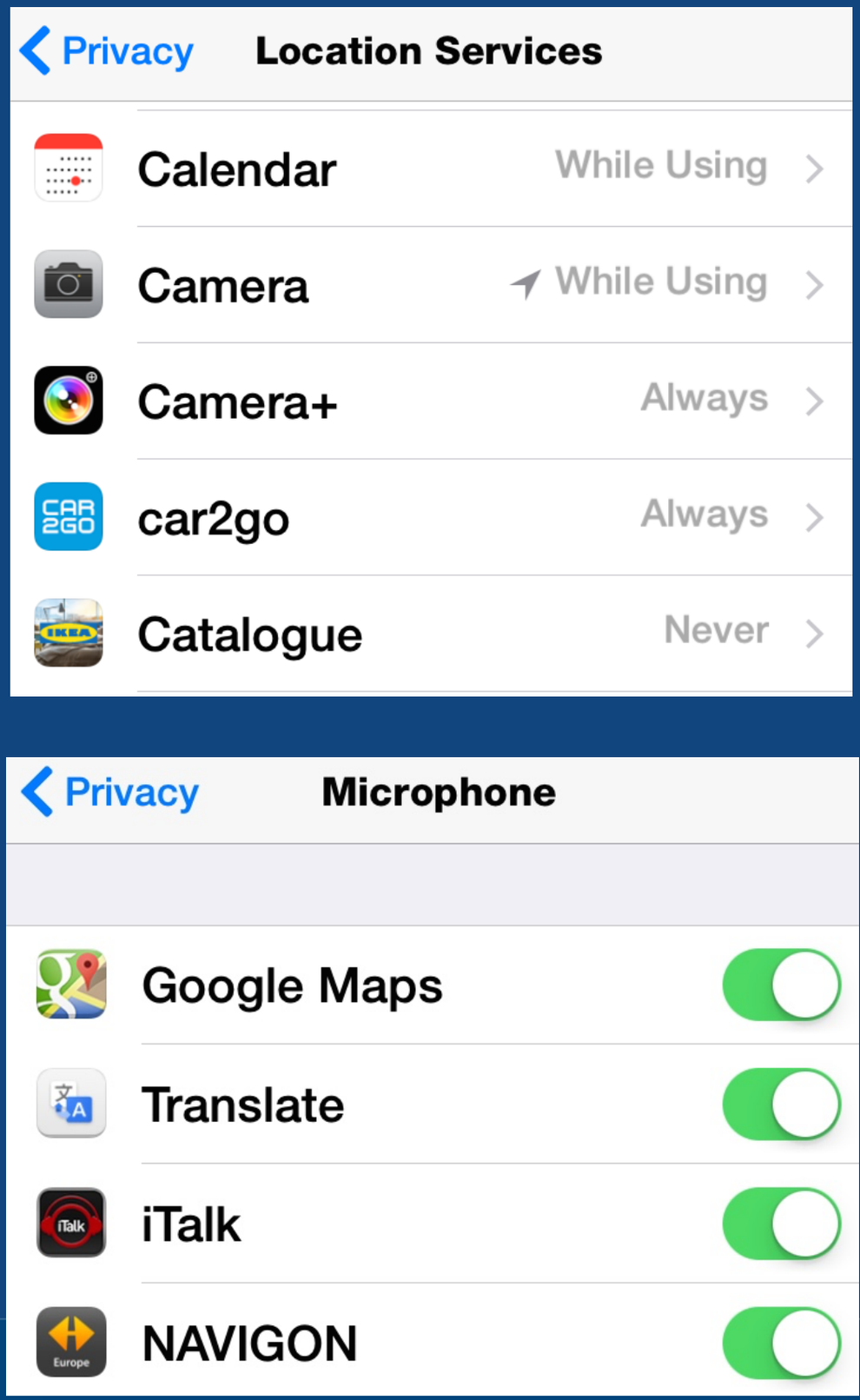
access control

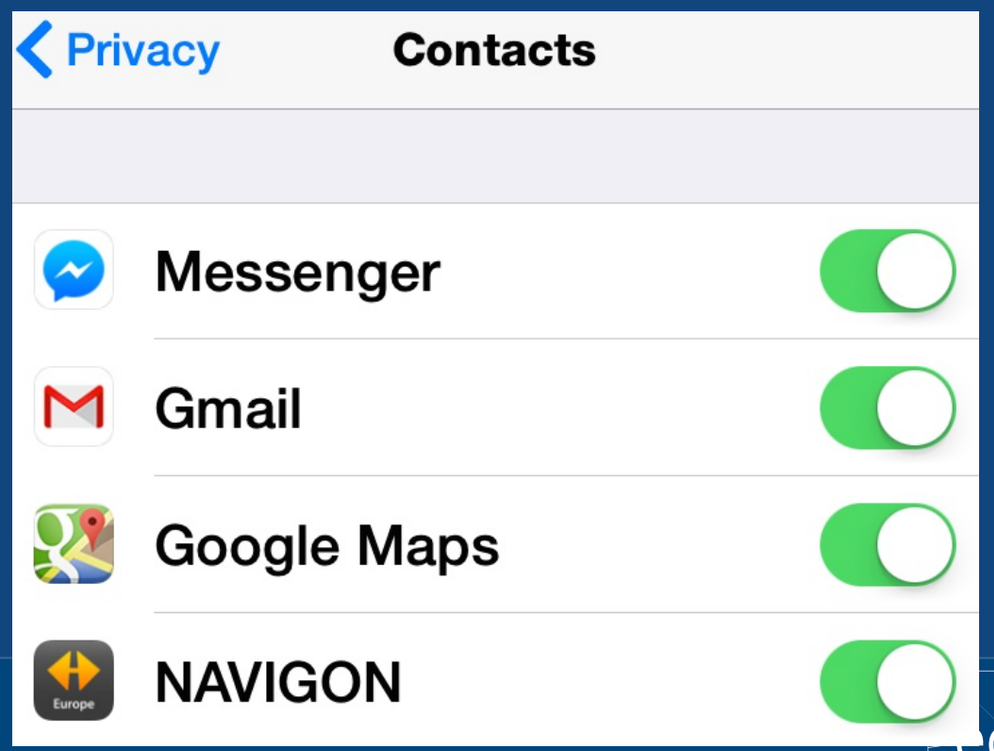
- each application can access only its own data on the file system
- User has to grant explicitly permissions to access

Location Service	Reminders	Photos
Calendars	Contacts	Social media accounts
Microphone	Camera	Motion activity
HomeKit	Bluetooth sharing	HealthKit

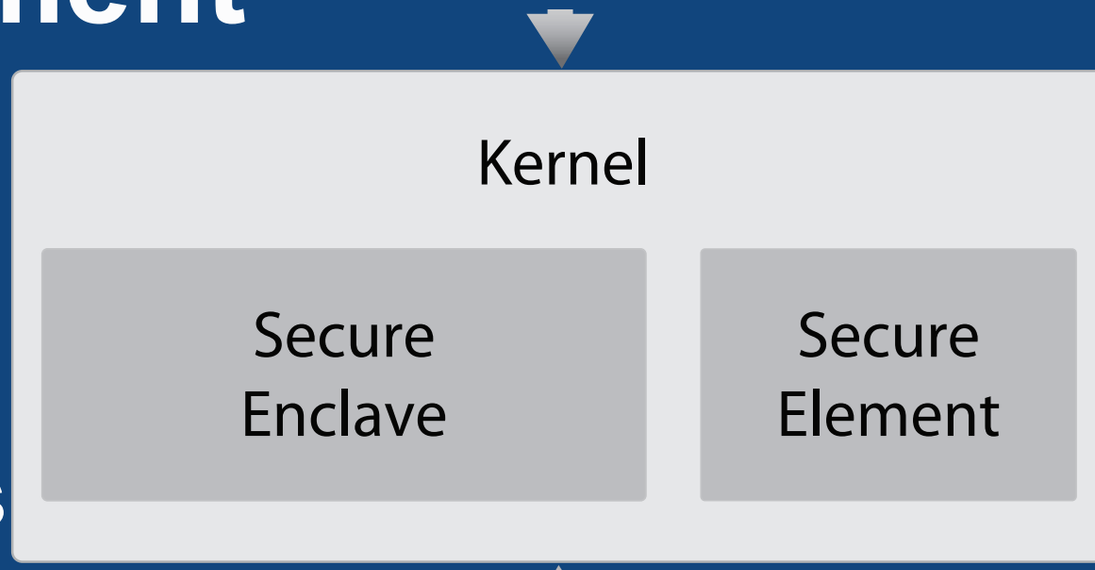
- iCloud data is accessible by default and can be turned off for individual applications

controls examples





secure element



- hosts a specially designed applet to manage Apple Pay.
- includes payment applets certified by the payment networks
- Credit or debit card data is sent from the payment network or card issuer encrypted to these payment applets using keys that are known only to the payment network and the payment applets' security domain.
- This data is stored within these payment applets and protected using the Secure Element's security features.
- POS terminal communicates directly with the Secure Element through the NFC controller over a dedicated hardware bus.

credits

- I. Cherapau, I. Muslukhov, N. Asanka, and K. Beznosov, “On the Impact of Touch ID on iPhone Passcodes,” presentation at the Symposium On Usable Privacy and Security (SOUPS), Ottawa, Canada, July 24, 2015.
- “iOS Security, iOS 9.0 or later” Apple, September 2015.
  - https://www.apple.com/business/docs/iOS_Security_Guide.pdf
- “iOS Security” presentation slides by Chun Zhang.

- image credits
  - https://support.apple.com/library/content/dam/edam/applecare/images/en_US/iphone/iphone5s/connect_to_itunes_screen.png