## How iOS Security Really Works

Session 705

Ivan Krstić Head of Security Engineering & Architecture









Criminals

Criminals

Business competitors

Criminals

Business competitors

Service providers

Criminals

Business competitors

Service providers

Nation states

Criminals

Business competitors

Service providers

Nation states

Romantic partners, family, friends

Criminals

Business competitors

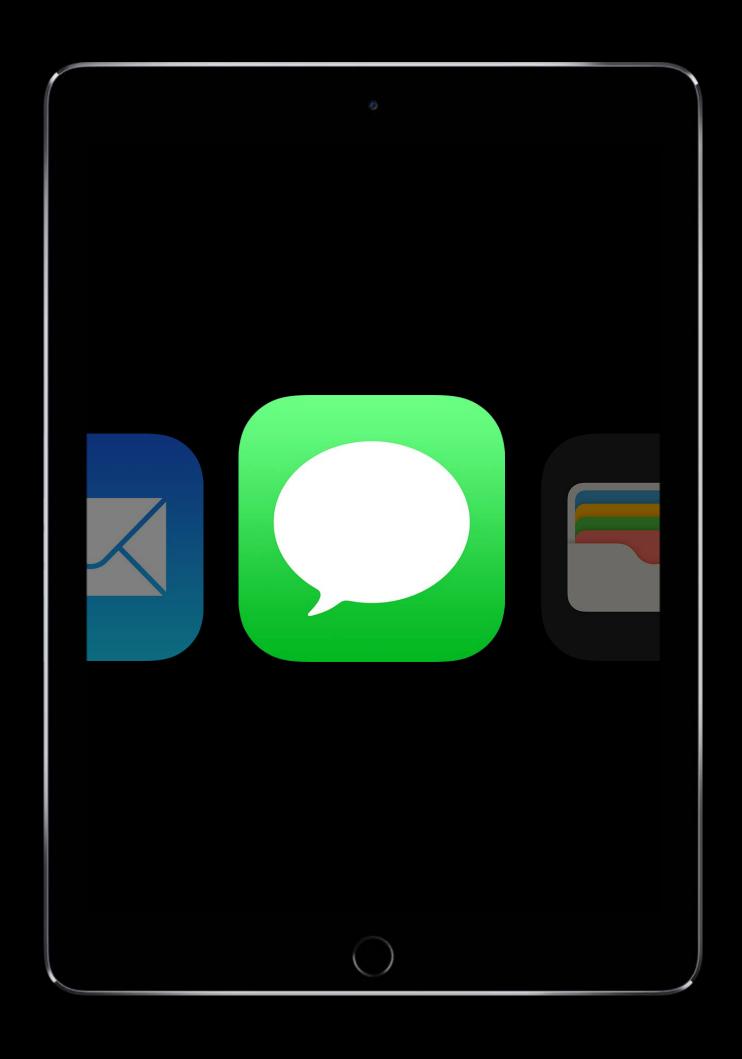
Service providers

Nation states

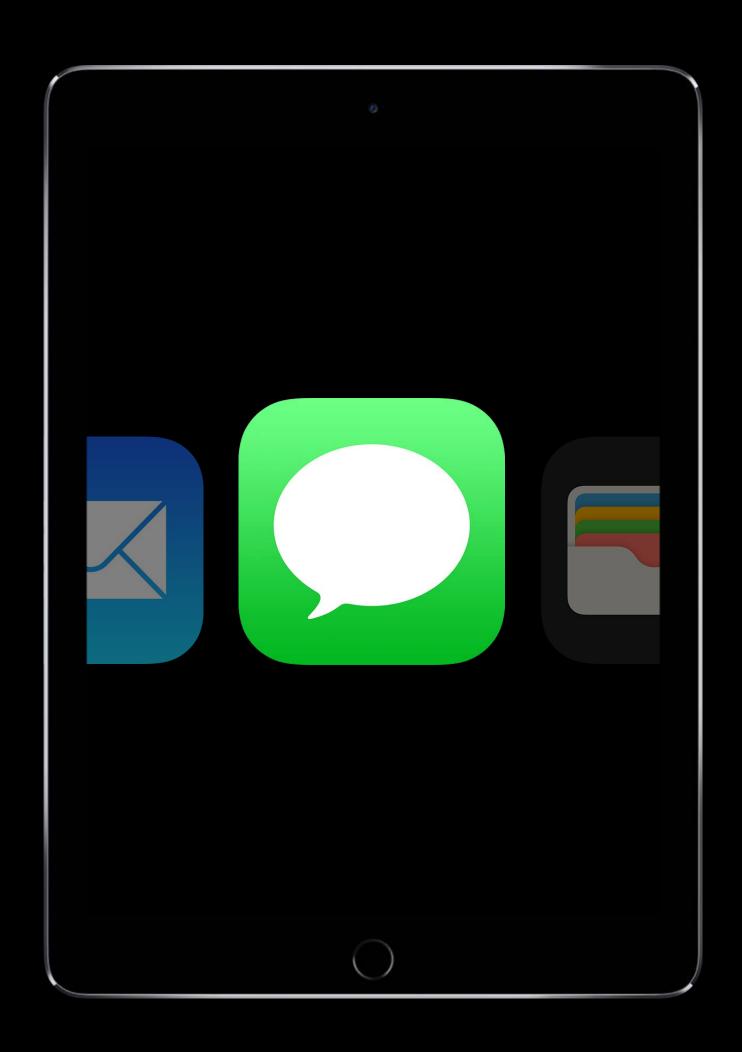
Romantic partners, family, friends

Cats



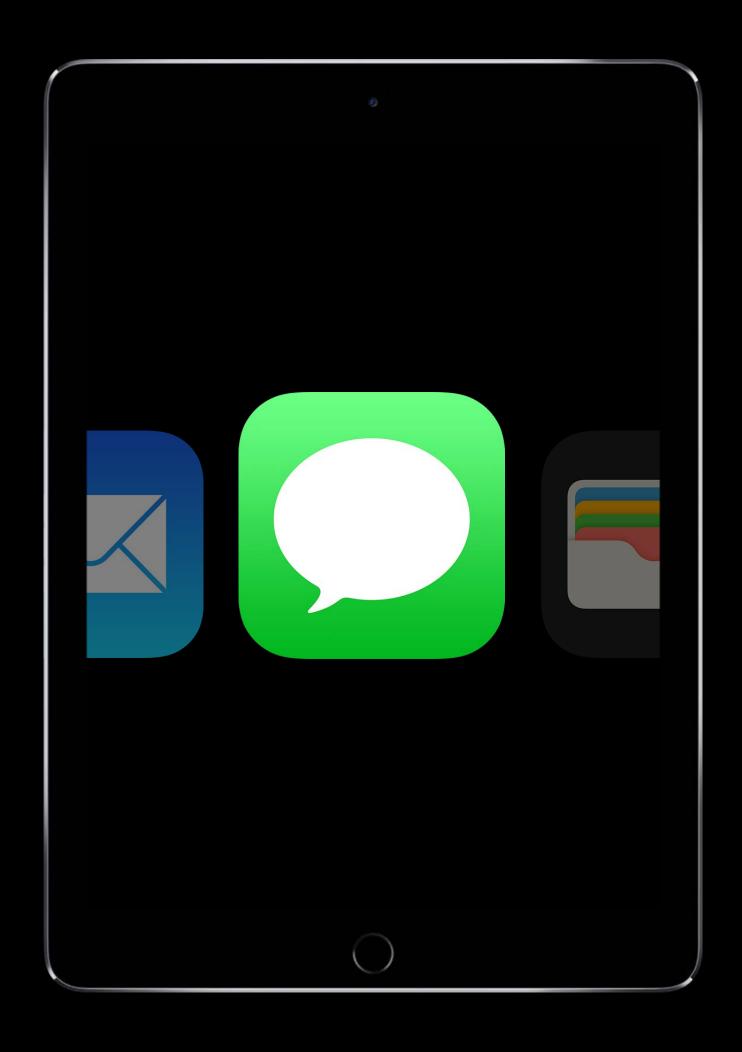


Personal stalking and surveillance



Personal stalking and surveillance

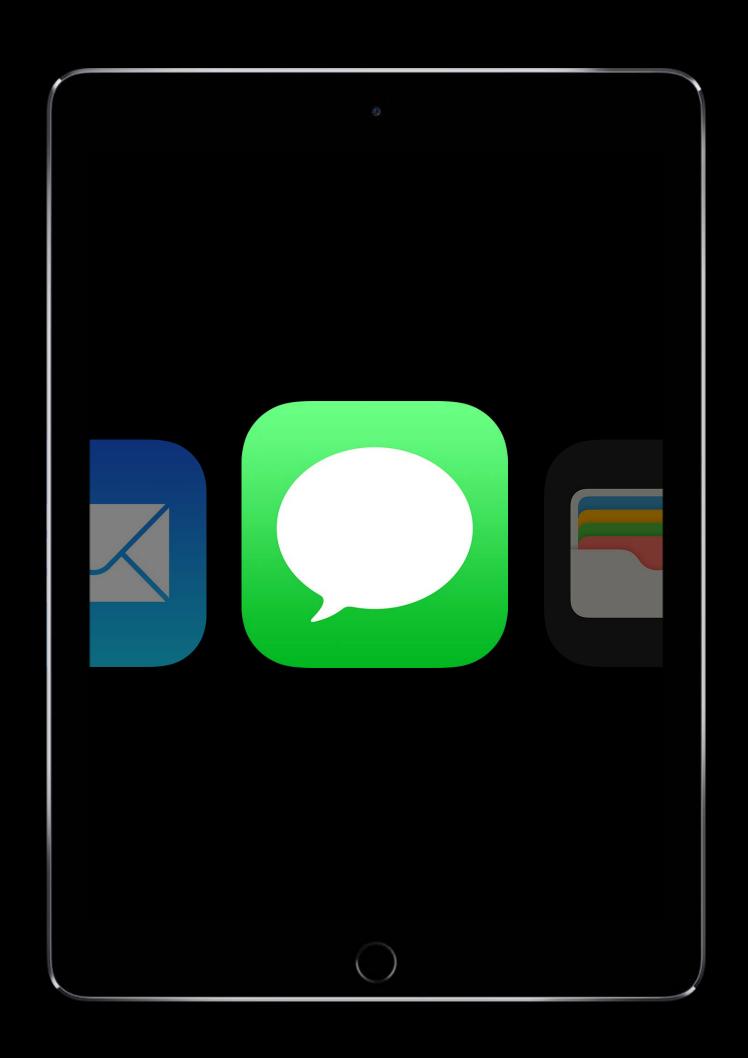
Corporate espionage



Personal stalking and surveillance

Corporate espionage

Direct financial benefit



#### How Do We Know This?

We see it on other platforms



#### But Not on iOS

No malware has affected iOS devices at scale



Decade-long effort to protect customers from security problems

Decade-long effort to protect customers from security problems Incredible scale

Decade-long effort to protect customers from security problems

Incredible scale

Every single iOS security feature is designed to thwart a real security threat

# iOS Security Pillars

#### iOS Platform Security

Users Upgrading their Software

Developers Building Secure Apps

#### iOS Platform Security

Users Upgrading their Software

Developers Building Secure Apps

#### Traditional Security

physical security

secure configuration

installing latest patches

password policy

vetted apps

mandated policies

| Traditional Security      | iOS Security                   |
|---------------------------|--------------------------------|
| physical security         | security built from silicon up |
| secure configuration      | secure default settings        |
| installing latest patches | easy updates                   |
| password policy           | Touch ID                       |
| vetted apps               | App Store                      |
| mandated policies         | ease of use                    |



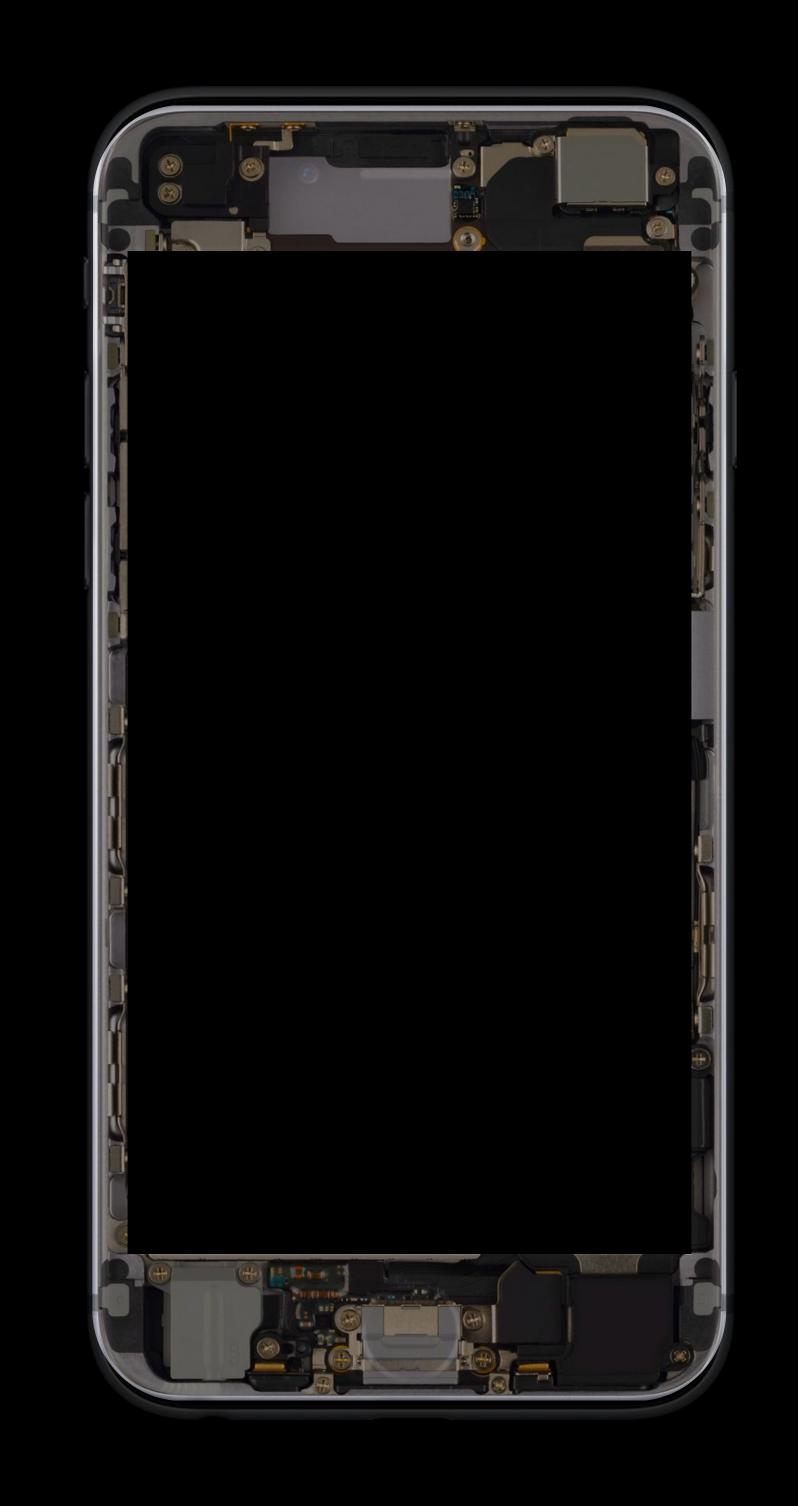
- (1) Secure Boot
- (2) Data Protection
- (3) Sandboxing
- (4) Code Signing
- (5) Touch ID



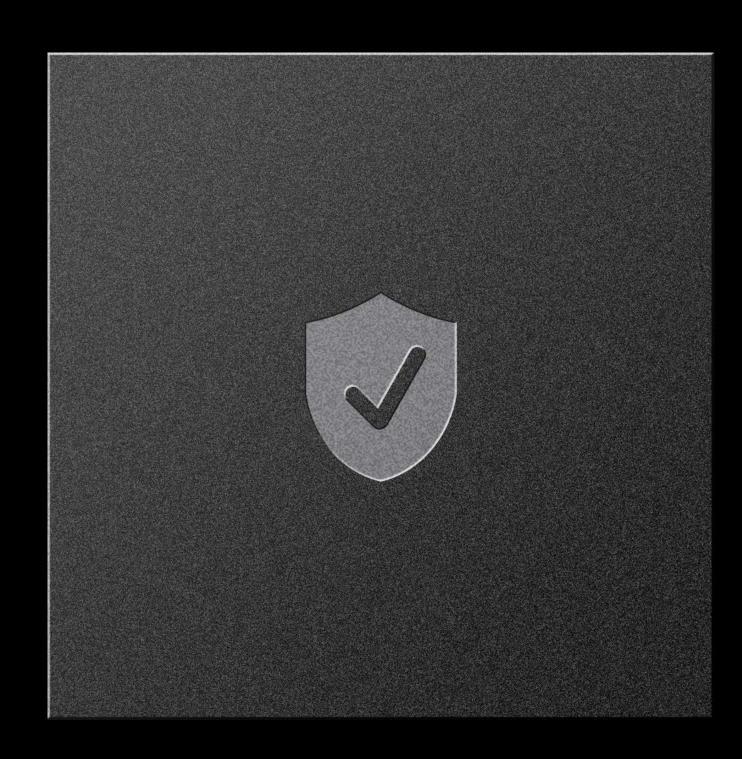
- (1) Secure Boot
- (2) Data Protection
- (3) Sandboxing
- (4) Code Signing
- (5) Touch ID







Trust built from silicon up



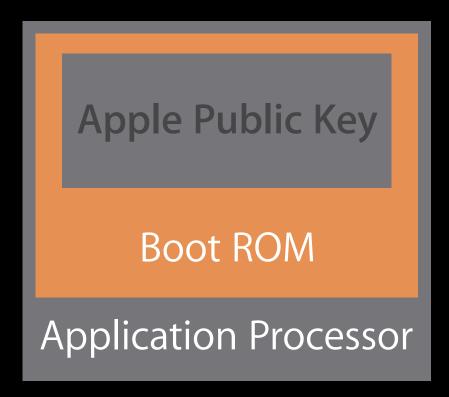
Application Processor

Boot ROM

Application Processor

Boot ROM

Application Processor





Boot ROM

Application Processor

Low-Level Bootloader (LLB)



Boot ROM

Application Processor

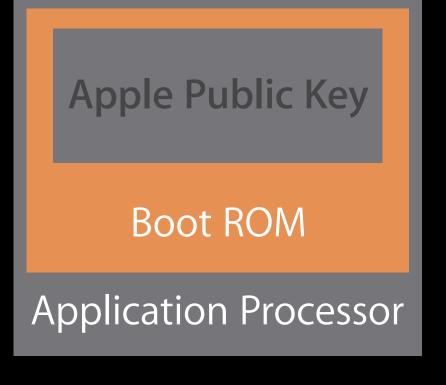
Low-Level Bootloader (LLB)



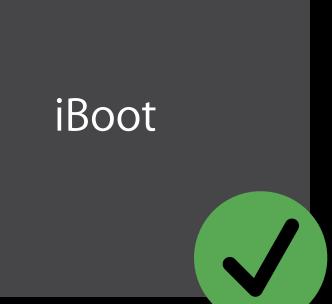
Low-Level Bootloader (LLB)

iBoot













iBoot





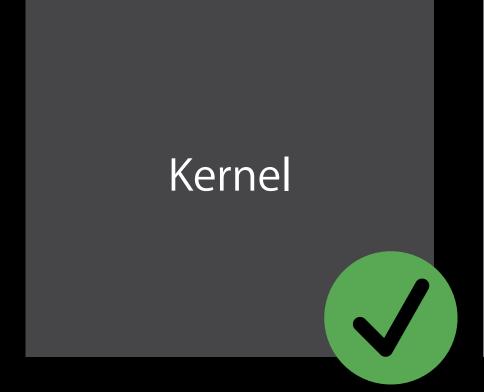
Low-Level Bootloader (LLB) iBoot

Kernel



Low-Level Bootloader (LLB)



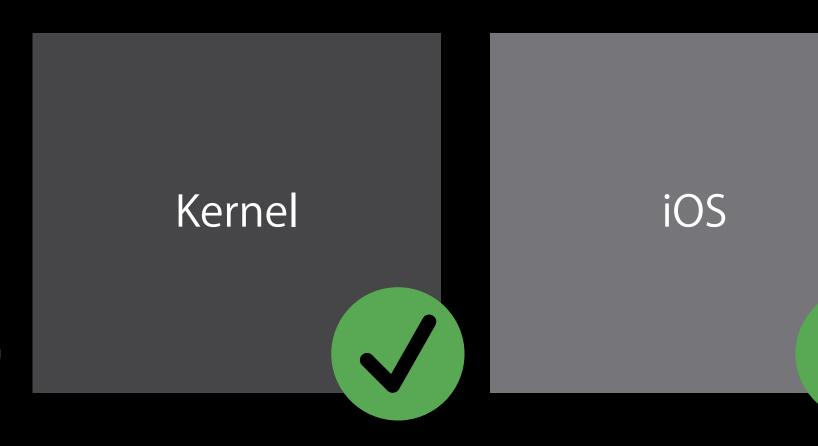


iOS











# Trusting Secure Boot

## Trusting Secure Boot

Keys are securely provisioned and managed by Apple

### Trusting Secure Boot

Keys are securely provisioned and managed by Apple Software updates are authorized individually for each device

- (1) Secure Boot
- (2) Data Protection
- (3) Sandboxing
- (4) Code Signing
- (5) Touch ID



- (1) Secure Boot
- (2) Data Protection
- (3) Sandboxing
- (4) Code Signing
- (5) Touch ID





User data is encrypted at rest with keys derived from the passcode



User data is encrypted at rest with keys derived from the passcode

Entangled with hardware key in SEP



SEP refuses to unlock after more than 10 incorrect passcode attempts

SEP refuses to unlock after more than 10 incorrect passcode attempts

'Erase Data' only controls erasure, not ability to unlock

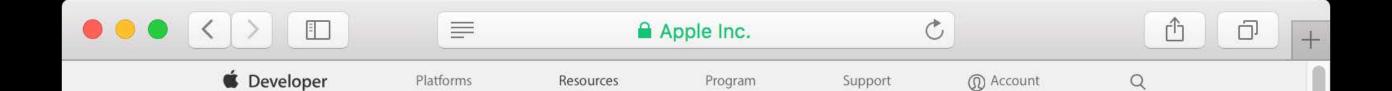
ALG2p0AnBLcatk4Y3UxF3Bk3UghOwx4uzAPmULKIbt51qIja ZUrmz8WtsBsmocaTdr3J9mzAzCXCvQPQ4N9E2kdbc0n94hYVkwq8dq. koFZyNzgA3PJb5KurLy+1r7CyodZYVP/gR+HbFszkAPvzhokUunPY6oUNa nfxhjTjDf7nAAI2c0r1H0wyU7ZAl<u>l</u>kSoNwZGgF1ocROPdHvG/YKti7zNcOJ KZZIL77oF6oRbB77I0n4EEYSBz KTLBY+w9E5oM7Dcu0ItEJNci4gF JgTEFURVNUIEtFWSkgPGRjaGV0c o24gQ2hldHNvbiAoQUxXQVl1 QGdtYWlsLmNvbT6JAkAEEwEK GH4AFCwkIBwMFFQoJCAsFFgID/ HgECF4AFAlVgGq0CGQEACg hAAyKKky4b4MMwc6QupuK9sF ZPWXk1UeUscW3sdsosJdX EWrkD1i+rqheywtYykP7mg2 l+qCLb6h/EQHGW9VY2two BreOKjc/VhHetufYPXt+Trk aw0TXQ/Qw5dtk4ZW3YoKt x/vK7cT5uZyk3LjswtSzTEy g4I8YWnBIYOKBEW9byz3Gi AUZNbmEuQ2ZNY3WmJPwXnLt xGq/M7mCR5/wLxx+5uzuTI7 EcRaSER69KK5d0jzGZWvBdyN /qAX4Al7Z0yjGUuW401TBw4mlc V+U3+rCF5Bzy1hJweP0GhM92N YGrChnaJto8vZ6NuCE/78CIiN ngNeIscT292TqB3jpl7PcHSYGm Ymns/nRxsq1ddm0a/YxiWH400 yUdWF8JZe9v2lHVTbZBeF15wbd+ wxGBpXZ51uV7EkAdS0orAGgZ7l 0iJ6rn65WHMJ2odMqdLs2gpmJaG2 NNdtrx45mttP/3+vDYEkXbUrpm PN7q7EYKd31M9fh9oA6GiLdHN7+uc QEQAAEBAAAAAAAAAAAAAAAA/9j/4 py7zN6L3ia5kc5x7V0DR/wAAMp SkZJRgABAQEASABIAAD/4QBYRX FU0AKgAAAAgAAgESAAMAAAABAAEA*A* AAEAAKACAAQAAAABAAAAgaADAAQAA AAQAAAABAAAAJgAAAAAAAA6ABAA AAAAlwAAAAD/7QA4UGhvdG9za MAA4QklNBAQAAAAAAAA4QklNBCUA*A* Q19QUk9GSUxFAAEBAAAMSExpbm8CE ABDUHYzZjwCyBOmACZjs+EJ+/bW50clJHQiBYWVogB84AAgAJAA AYWNzcE1TRlQAAAAASUVDIHNSR0IA*A* AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA AAAAAAAAAAAAPbWAAEAAAAA0 AAAAAAAAAAAAAAAAAAAAAAAAAA RY3BydAAAAVAAAAAzZGVzYwAAAYQAA d3RwdAAAAfAAAAAUYmtwdAAAAg( ₩AUclhZWgAAAhgAAAAUZ1hZWgAAAiwA/ 'lhZWgAAAkAAAAUZG1uZAAAAlQAAABwZG1kZAAAAsQAAACIdnVlZAAAA0wA nlldwAAA9QAAAAkbHVtaQAAA/gAAAAUbWVhcwAABAwAAAAkdGVjaAAABDA `SQwAABDwAAAgMZ1RSQwAABDwAAAgMYlRSQwAABDwAAAgMdGV4dAAAA/

"OgKGMpIDE50TggSGV3bGV0dC1QYWNrYXJkIENvbXBhbnkAAGP"

Standard algorithms
Internal security audits
3rd-party code review

gTEFURVNUIEtFWSkgPGRjaGV0c GdtYWlsLmNvbT6JAkAEEwEK H4AFCwkIBwMFFQoJCAsFFgIDA lgECF4AFAlVgGq0CGQEACg hAAyKKky4b4MMwc6QupuK9sF EWrkD1i+rqheywtYykP7mg2 ZPWXk1UeUscW3sdsosJdX BreOKjc/VhHetufYPXt+Trk l+qCLb6h/EQHGW9VY2two k/vK7cT5uZyk3LjswtSzTEy aw0TXQ/Qw5dtk4ZW3YoKt AUZNbmEuQ2ZNY3WmJPwXnLt g4I8YWnBIYOKBEW9byz3G EcRaSER69KK5d0jzGZWvBdyN kGq/M7mCR5/wLxx+5uzuTI7 /+U3+rCF5Bzy1hJweP0GhM92N /qAX4Al7Z0yjGUuW401TBw4mlc ngNeIscT292TqB3jpl7PcHSYGm yUdWF8JZe9v2lHVTbZBeF15wbd+ Ymns/nRxsq1ddm0a/YxiWH400 wxGBpXZ51uV7EkAdSOorAGgZ7N ðiJ6rn65WHMJ2odMqdLs2gpmJaG2 NNdtrx45mttP/3+vDYEkXbUrpm PN7q7EYKd31M9fh9oA6GiLdHN7+uc py7zN6L3ia5kc5x7V0DR/wAAMp QEQAAEBAAAAAAAAAAAAAAAA/9j/4 SkZJRgABAQEASABIAAD/4QBYRX FU0AKgAAAAgAAgESAAMAAAABAAEA*A* AAQAAAABAAAAJqAAAAAAAAABAA AAAA lwAAAAD/7QA4UGhvdG9zaG MAA4Qk\NBAQAAAAAAA4Qk\NBCUAA Q19QUk9GSUxFAAEBAAAMSExpbm8CE ABDUHYzZjwCyBOmACZjs+EJ+/+ bW50clJHQiBYWVogB84AAgAJAA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA Y3BydAAAAVAAAAAzZGVzYwAAAYQA*A* UclhZWgAAAhgAAAAUZ1hZWgAAAiwA/

``lldwAAA9QAAAAkbHVtaQAAA/gAAAAUbWVhcwAABAwAAAAkdGVjaAAABDA ``SQwAABDwAAAgMZ1RSQwAABDwAAAgMYlRSQwAABDwAAAgMdGV4dAAAA/ ``OgKGMpIDE50TggSGV3bGV0dC1QYWNrYXJkIENvbXBhbnkAAGP'





#### Cryptographic Libraries

The same libraries that secure iOS and OS X are available to third-party developers to help them build advanced security features.

#### Security Framework

Security Framework provides interfaces for managing certificates, public and private keys, and trust policies. It supports the generation of cryptographically secure pseudorandom numbers. It also supports the storage of certificates and cryptographic keys in the keychain, which is a secure repository for sensitive user data.

iOS Security Framework Reference

OS X Security Framework Reference

Security Framework on Apple Open Source

Apple Developer Forums: Security

#### Common Crypto

The Common Crypto library provides additional support for operations like symmetric encryption, hash-based message authentication codes, and digests.

Cryptographic Services Guide

Common Crypto on Apple Open Source

#### corecrypto

Both Security Framework and Common Crypto rely on the corecrypto library to provide implementations of low level cryptographic primitives. This is also the library submitted for validation of compliance with U.S. Federal Information Processing Standards (FIPS) 140-2 Level 1. Although corecrypto does not directly provide programming interfaces for developers and should not be used by iOS or OS X apps, the source code is available to allow for verification of its security characteristics and correct functioning.

Download corecrypto source

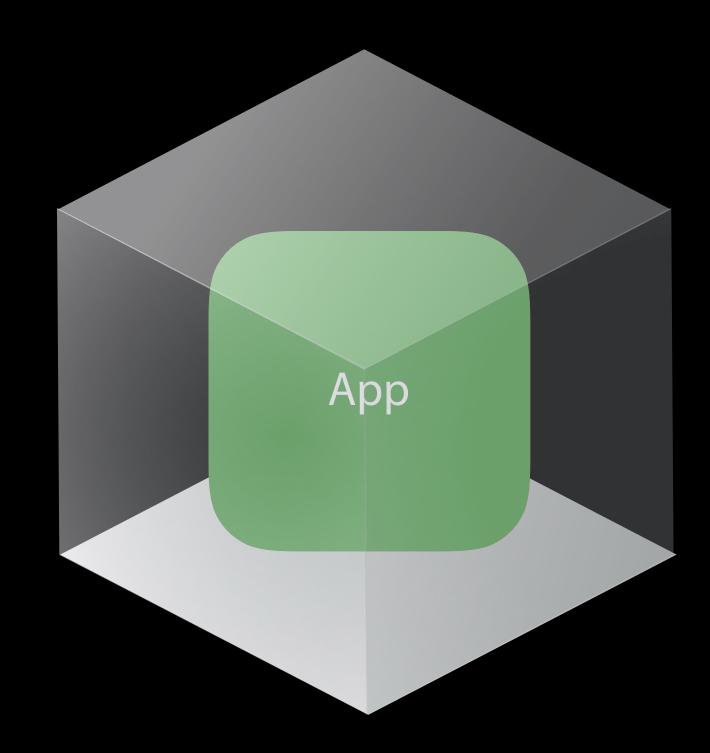
- (1) Secure Boot
- (2) Data Protection
- (3) Sandboxing
- (4) Code Signing
- (5) Touch ID

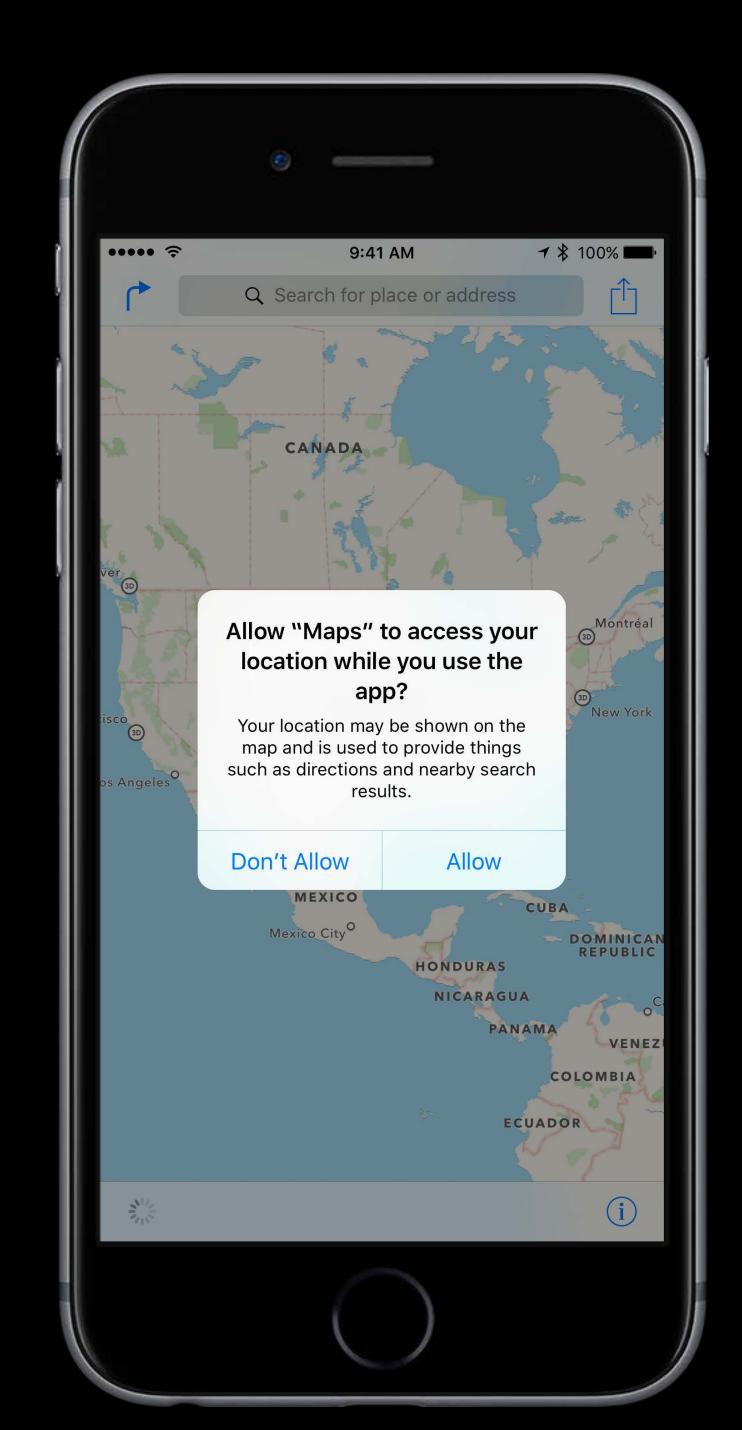


- (1) Secure Boot
- (2) Data Protection
- (3) Sandboxing
- (4) Code Signing
- (5) Touch ID

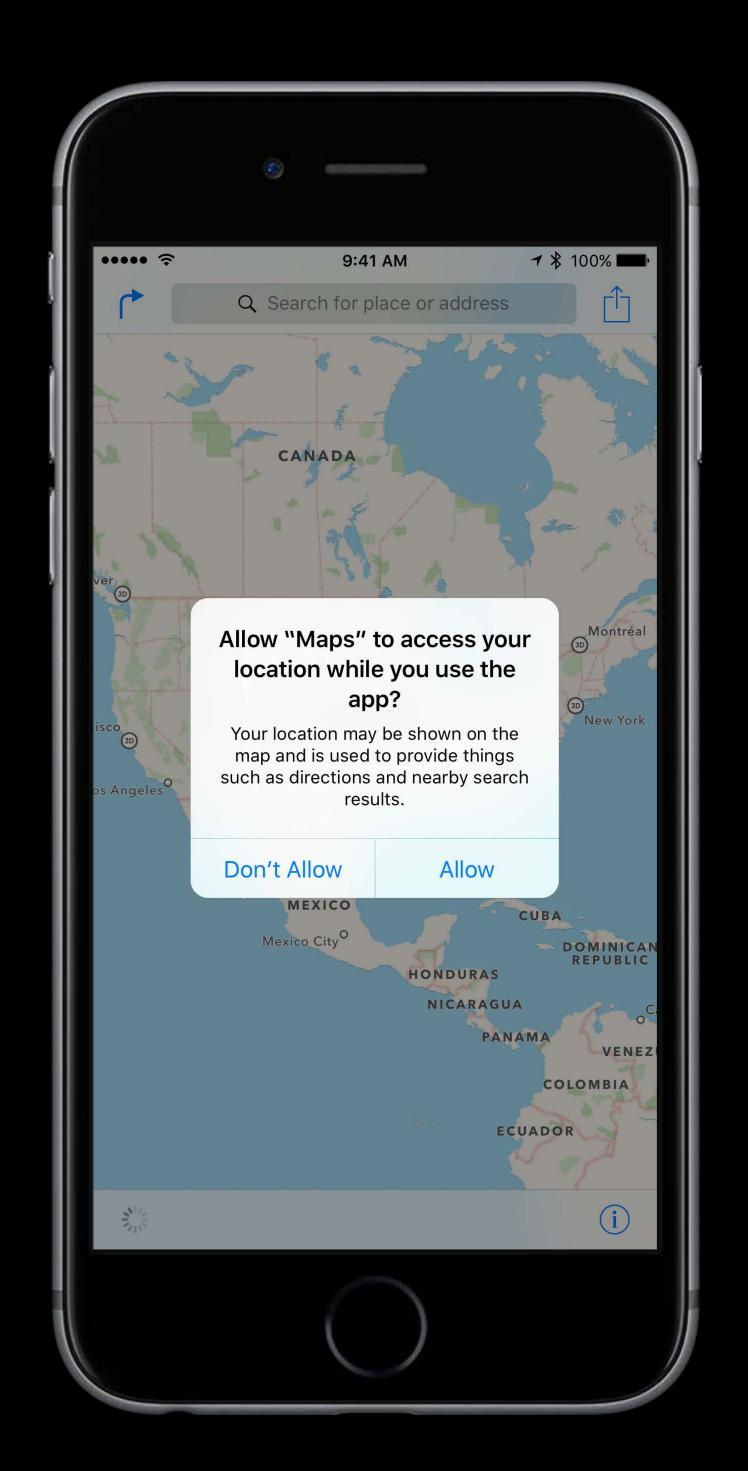


Isolating data between apps





Transparency
Consent
Control



- (1) Secure Boot
- (2) Data Protection
- (3) Sandboxing
- (4) Code Signing
- (5) Touch ID



- (1) Secure Boot
- (2) Data Protection
- (3) Sandboxing
- (4) Code Signing
- (5) Touch ID



# Code Signing

## Code Signing

Attacker's first step: code execution iOS code signing covers not just the OS, but every app that runs

- (1) Secure Boot
- (2) Data Protection
- (3) Sandboxing
- (4) Code Signing
- (5) Touch ID



- (1) Secure Boot
- (2) Data Protection
- (3) Sandboxing
- (4) Code Signing





Average user unlocks per day



Easy Fast Protects user data

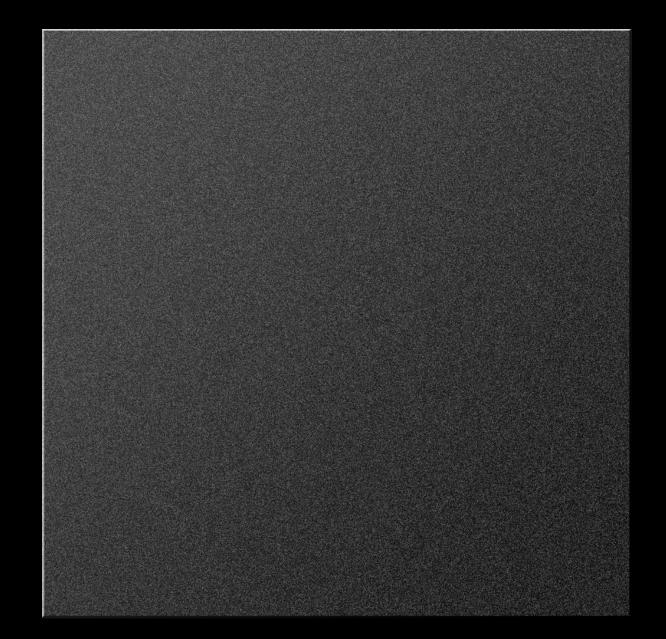


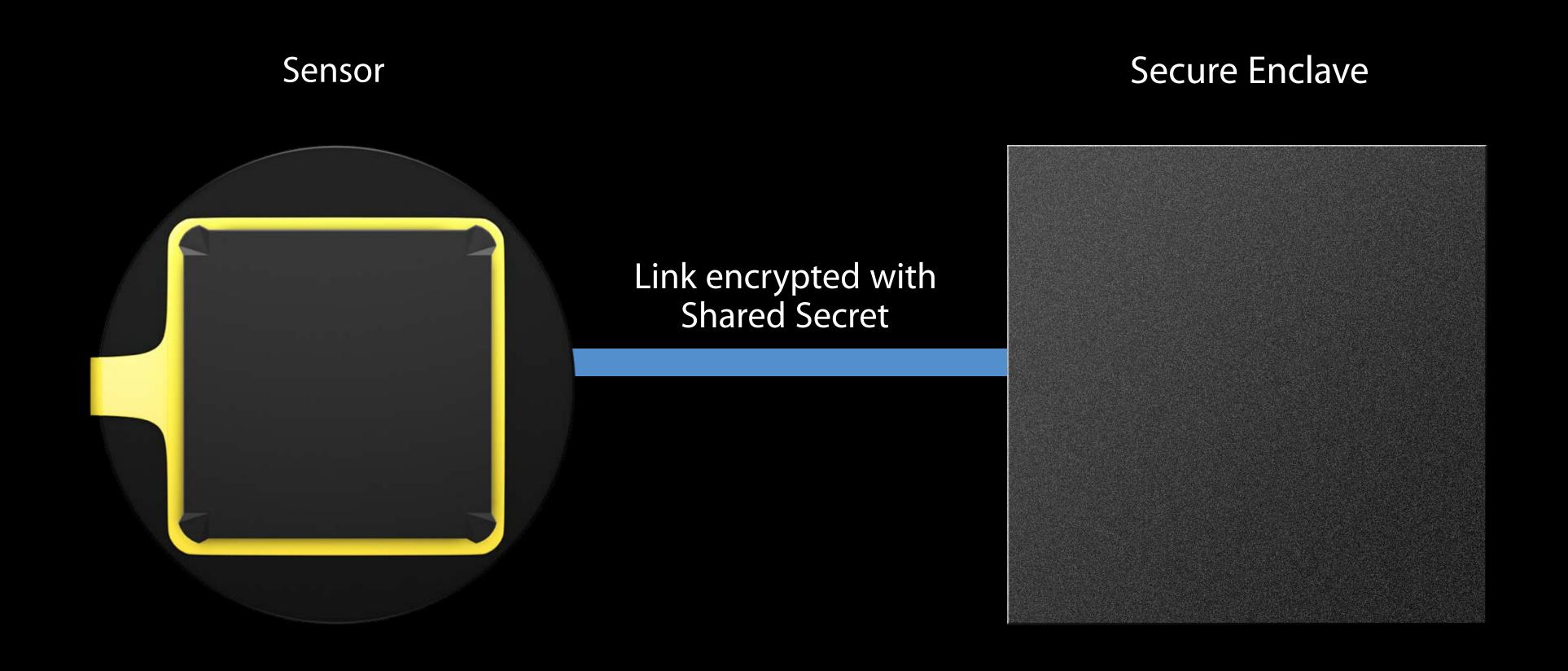


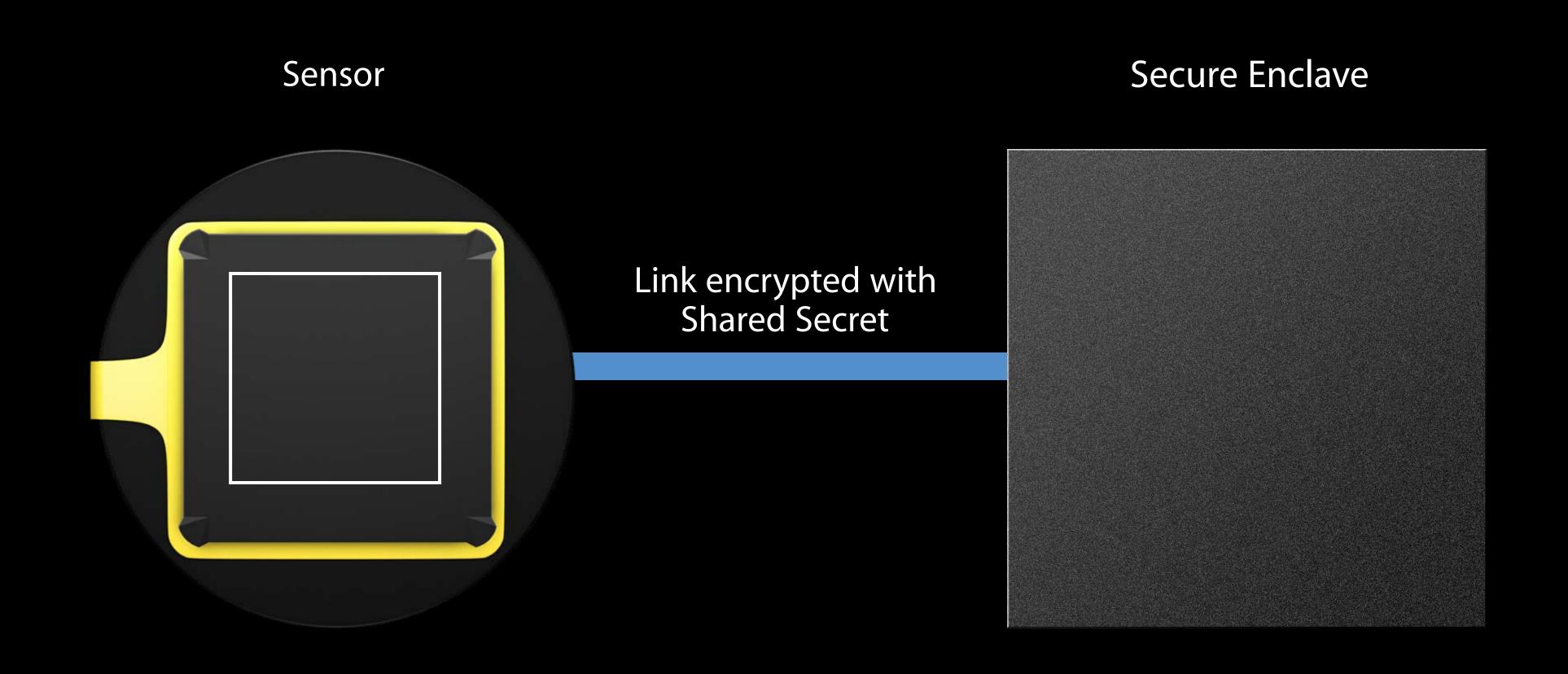
Sensor

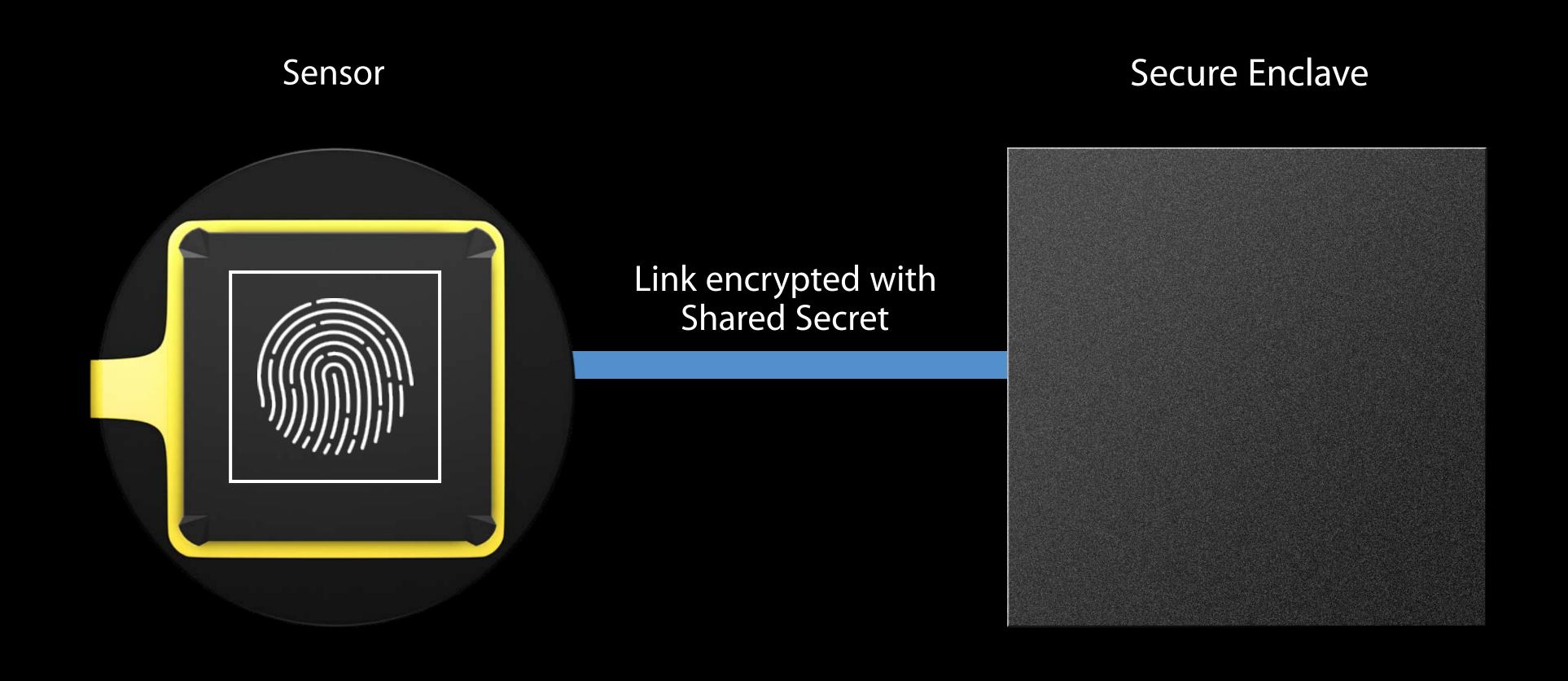


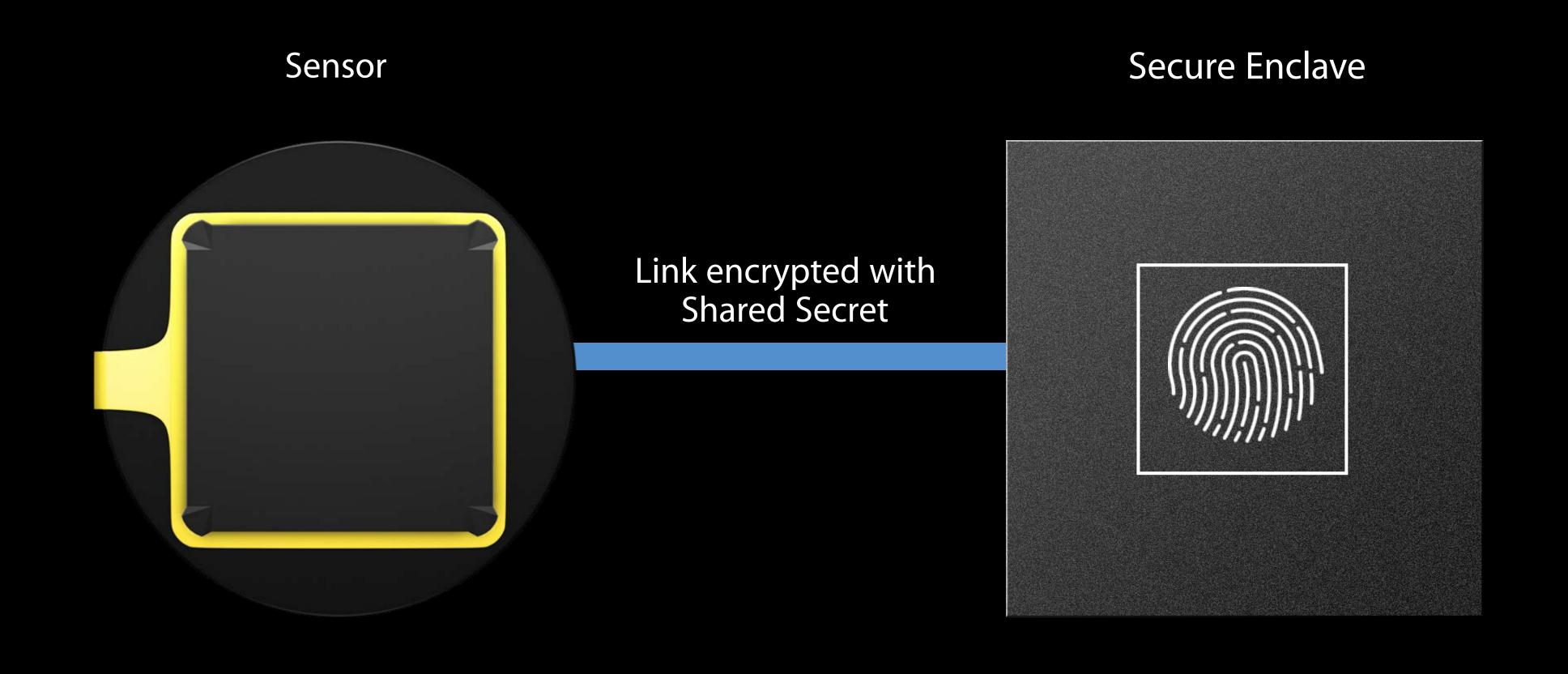
Secure Enclave

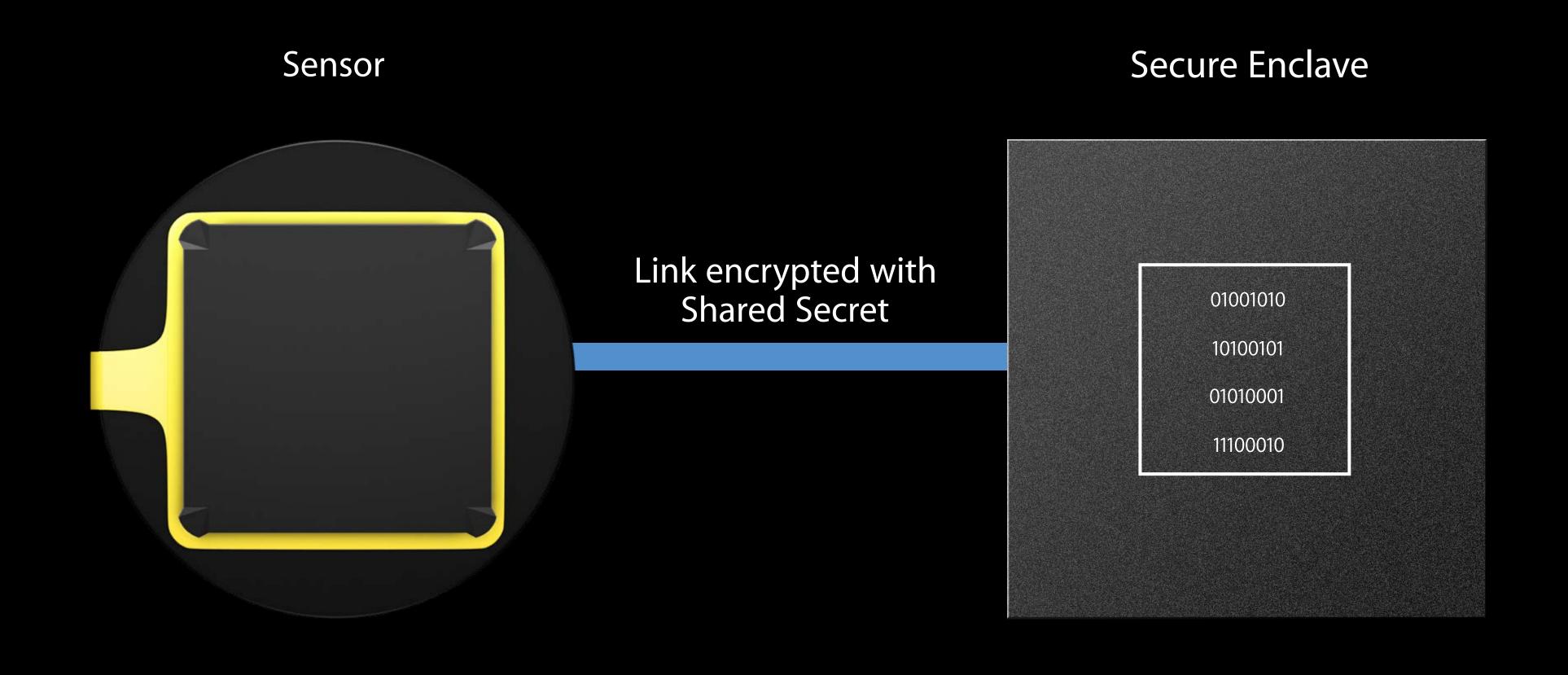


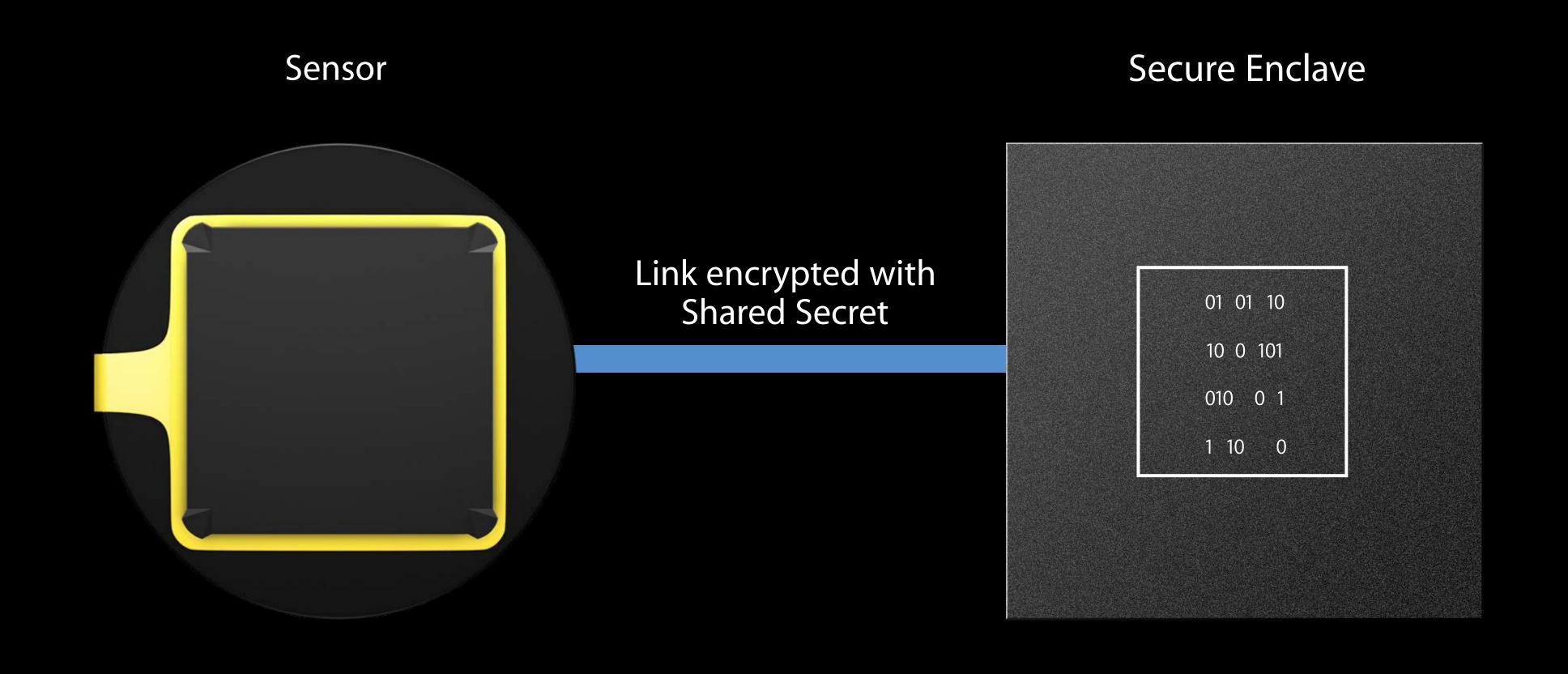




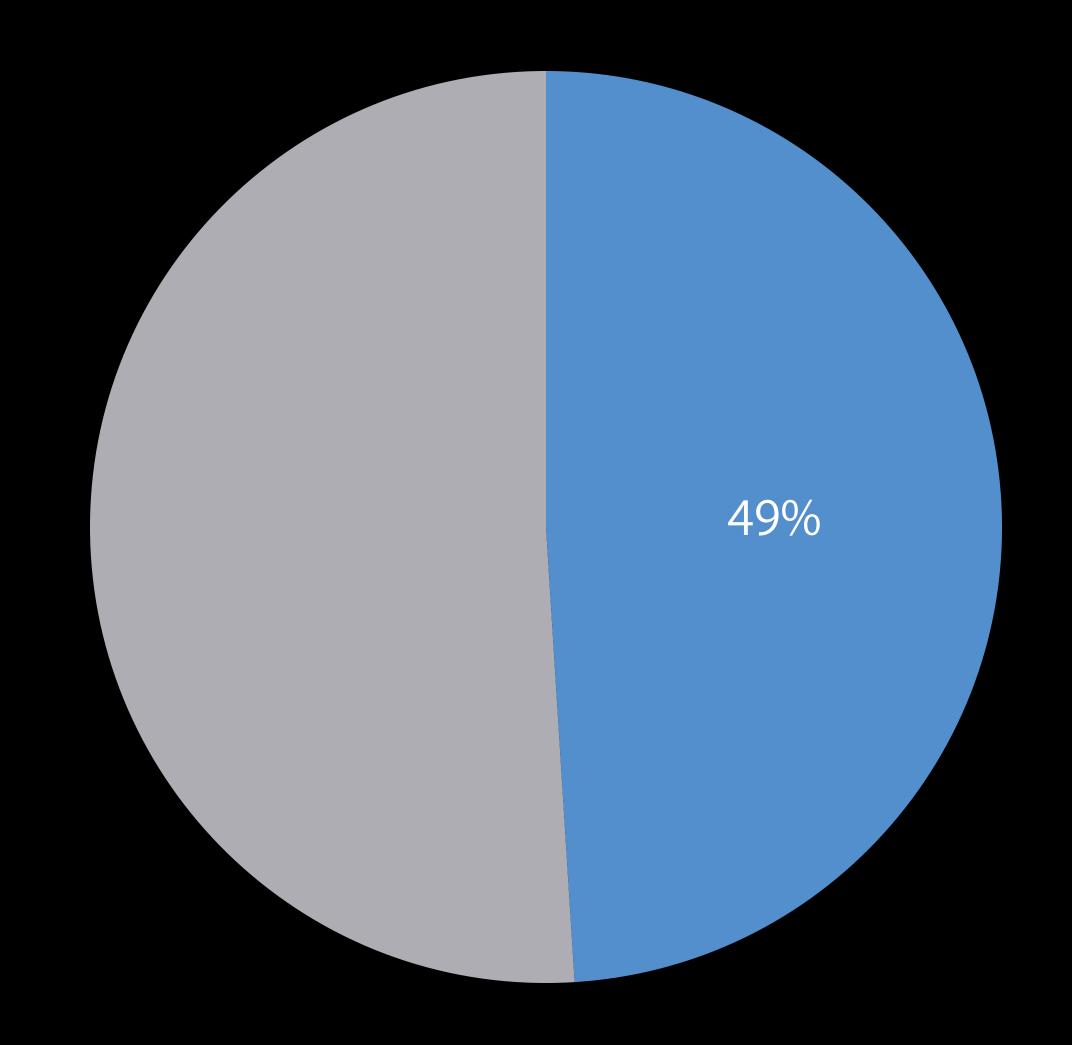




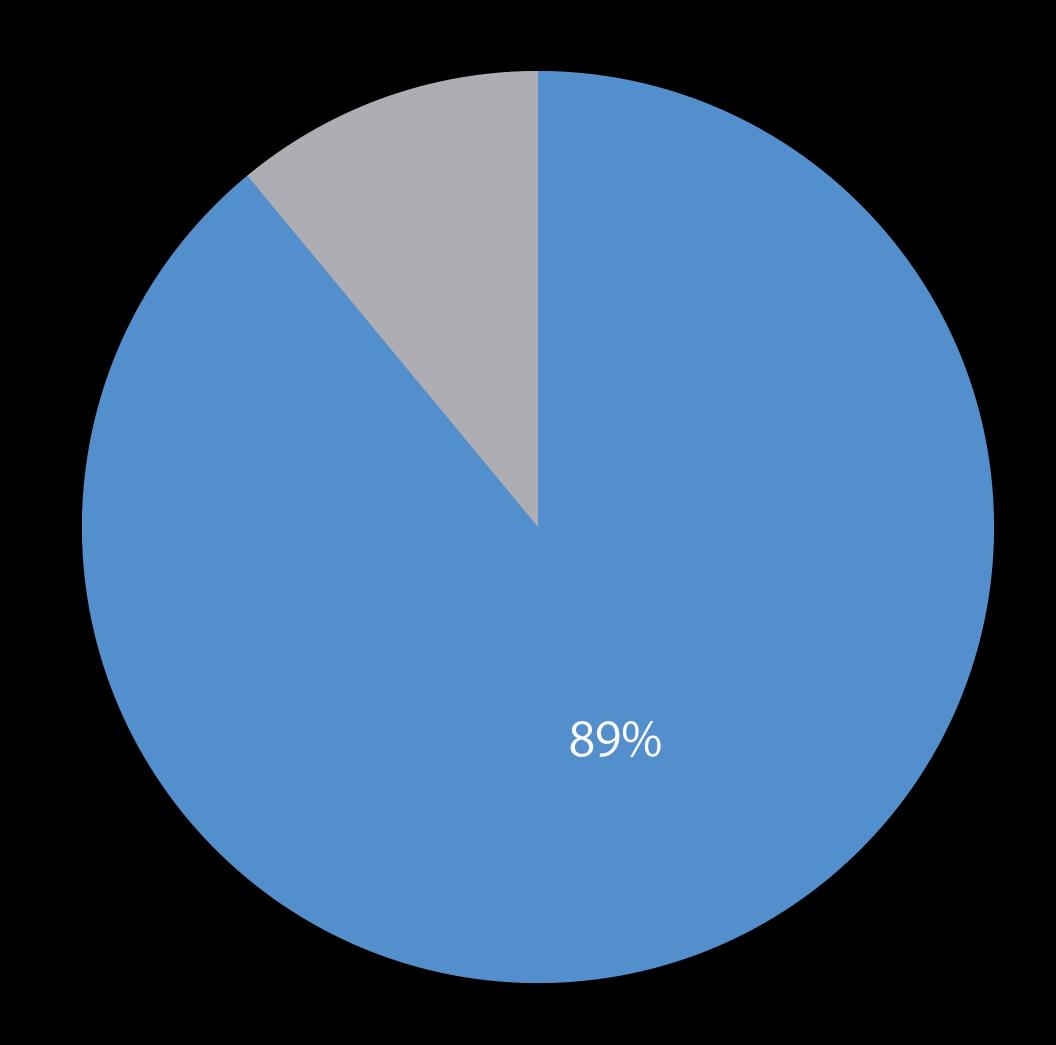




# Passcode usage before Touch ID



# Passcode usage after Touch ID



#### iOS Platform Security

Users Upgrading their Software

Developers Building Secure Apps

#### iOS Platform Security

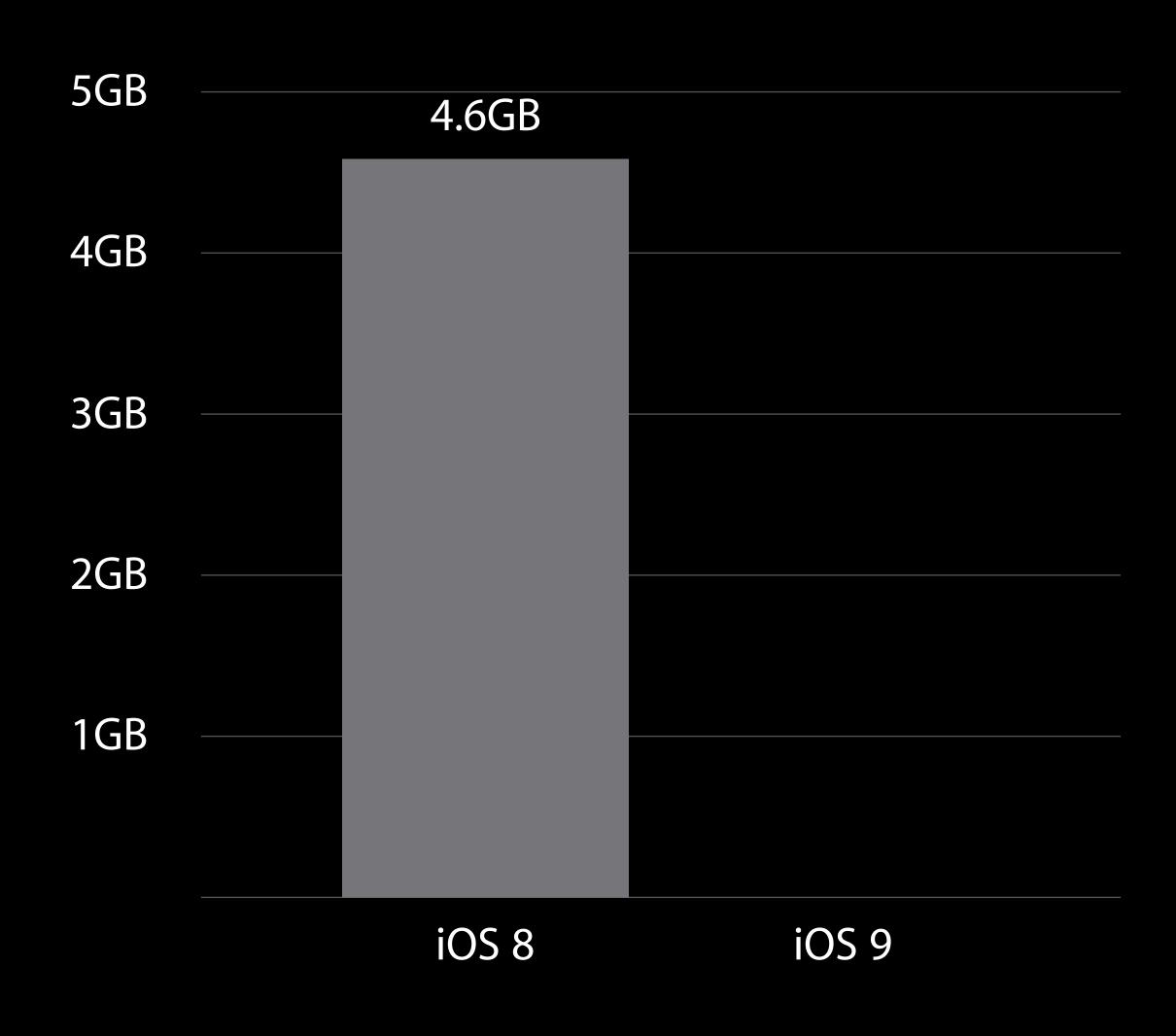
Users Upgrading their Software

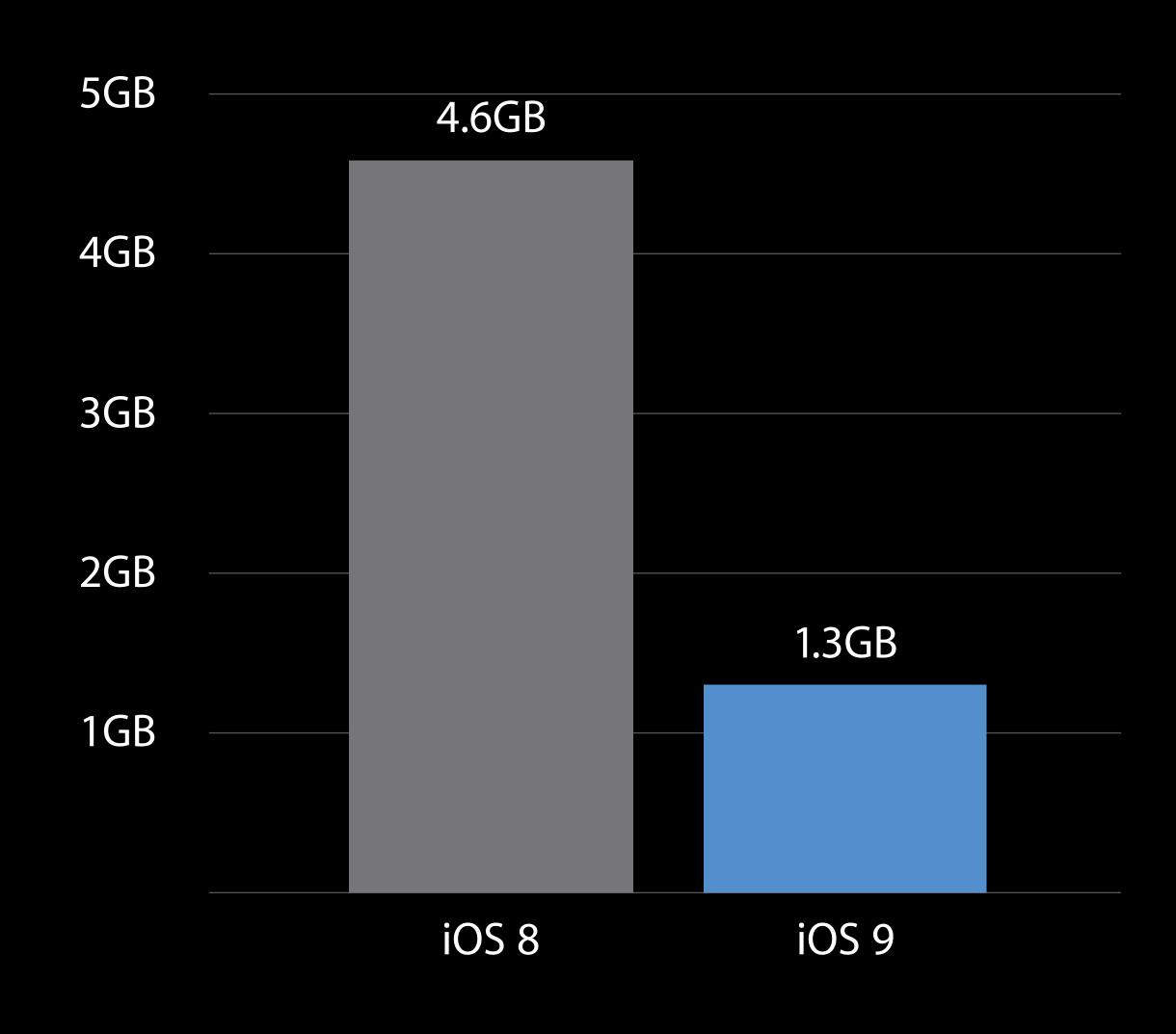
Developers Building Secure Apps

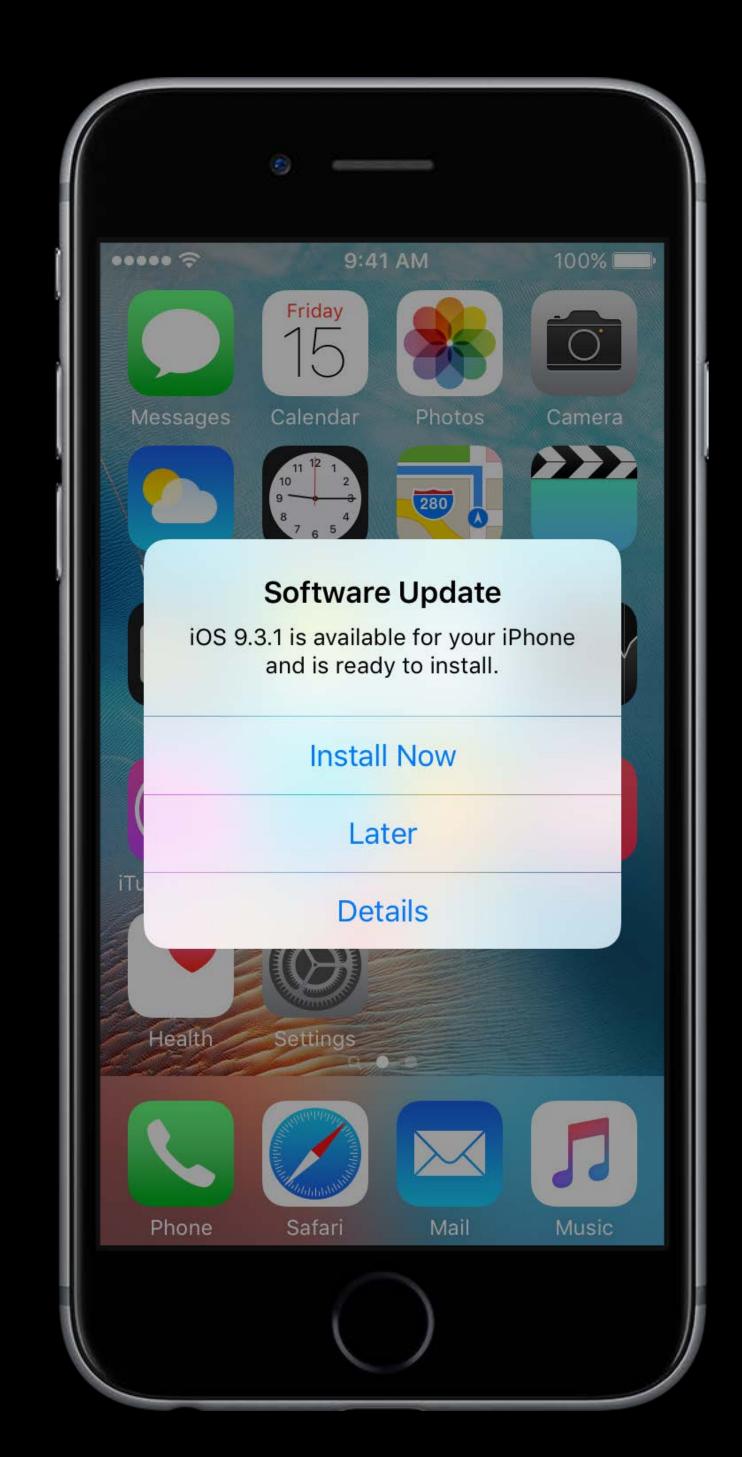


| 5GB |  |
|-----|--|
| 4GB |  |
| 3GB |  |
| 2GB |  |
| 1GB |  |
|     |  |

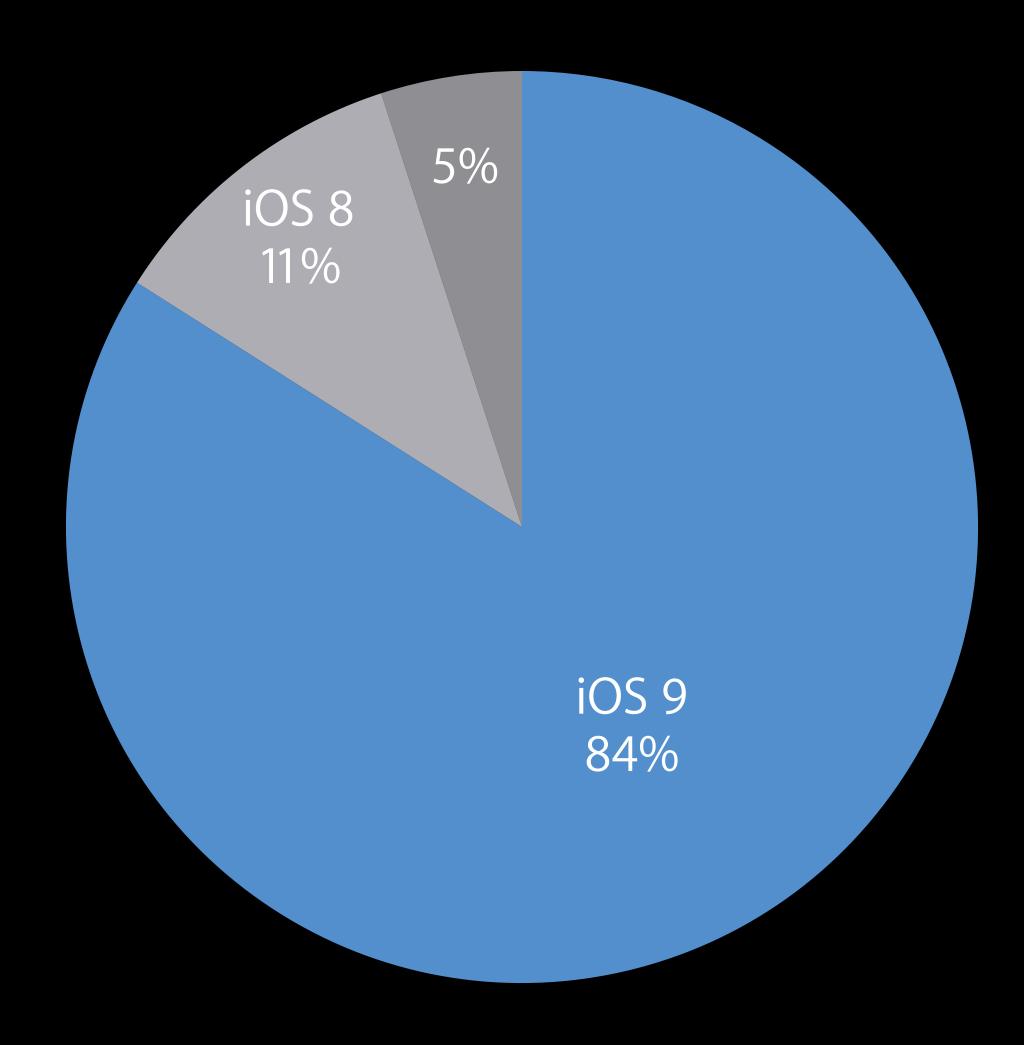
| 5GB |       |       |
|-----|-------|-------|
| 4GB |       |       |
| 3GB |       |       |
| 2GB |       |       |
| 1GB |       |       |
|     |       |       |
|     | iOS 8 | iOS 9 |



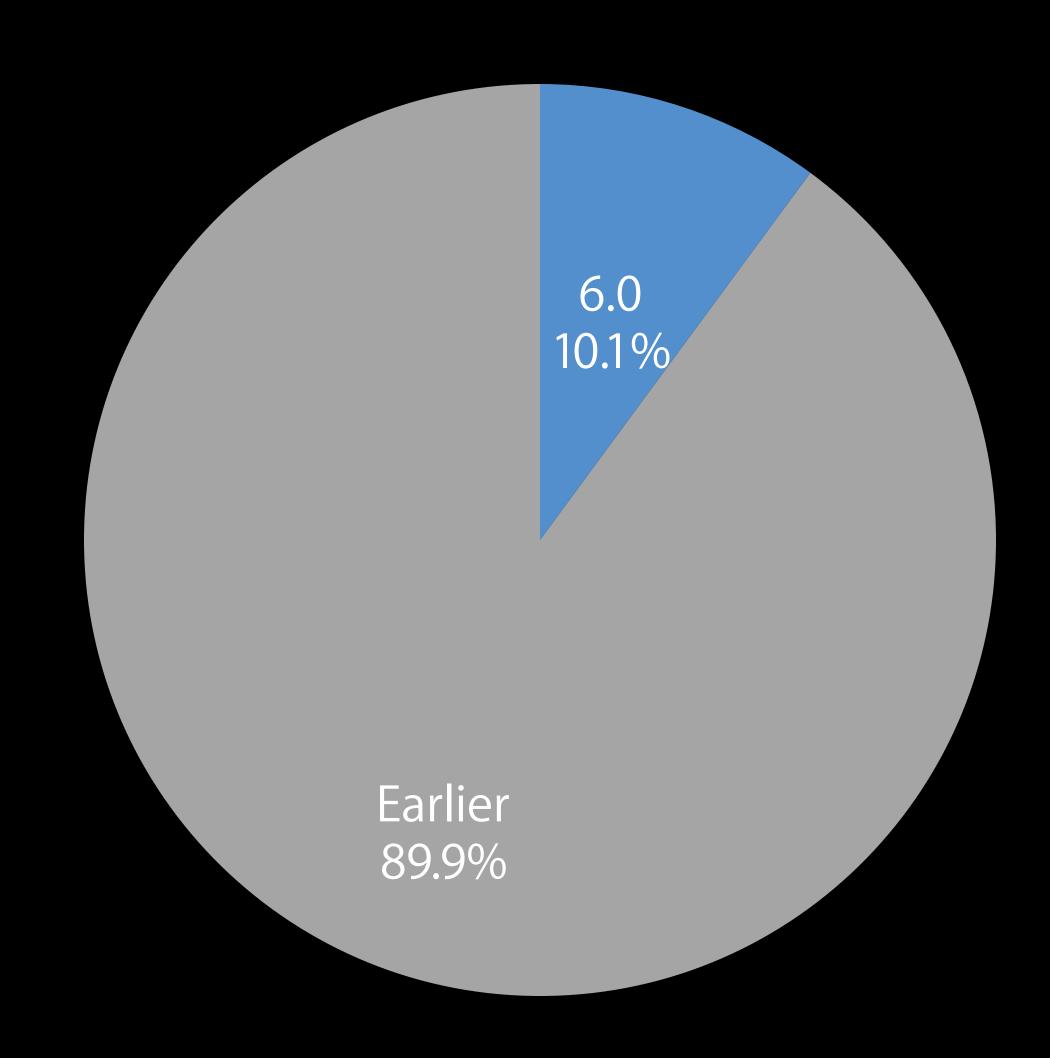




#### iOS Installed Base



#### Android Installed Base



#### iOS Platform Security

Users Upgrading their Software

Developers Building Secure Apps

#### iOS Platform Security

Users Upgrading their Software

Developers Building Secure Apps



Touch ID

App Transport Security





#### Touch ID

App Transport Security

Required by App Store at end of 2016





#### Touch ID

#### App Transport Security

- Required by App Store at end of 2016
- TLS v1.2, with exceptions for alreadyencrypted bulk data like media streaming





You are responsible for third-party code you include in your apps



You are responsible for third-party code you include in your apps

Libraries you use may undermine app security



You are responsible for third-party code you include in your apps

Libraries you use may undermine app security

Keep them current!



#### iOS Platform Security

Users Upgrading their Software

Developers Building Secure Apps

#### iOS Platform Security

Users Upgrading their Software

Developers Building Secure Apps

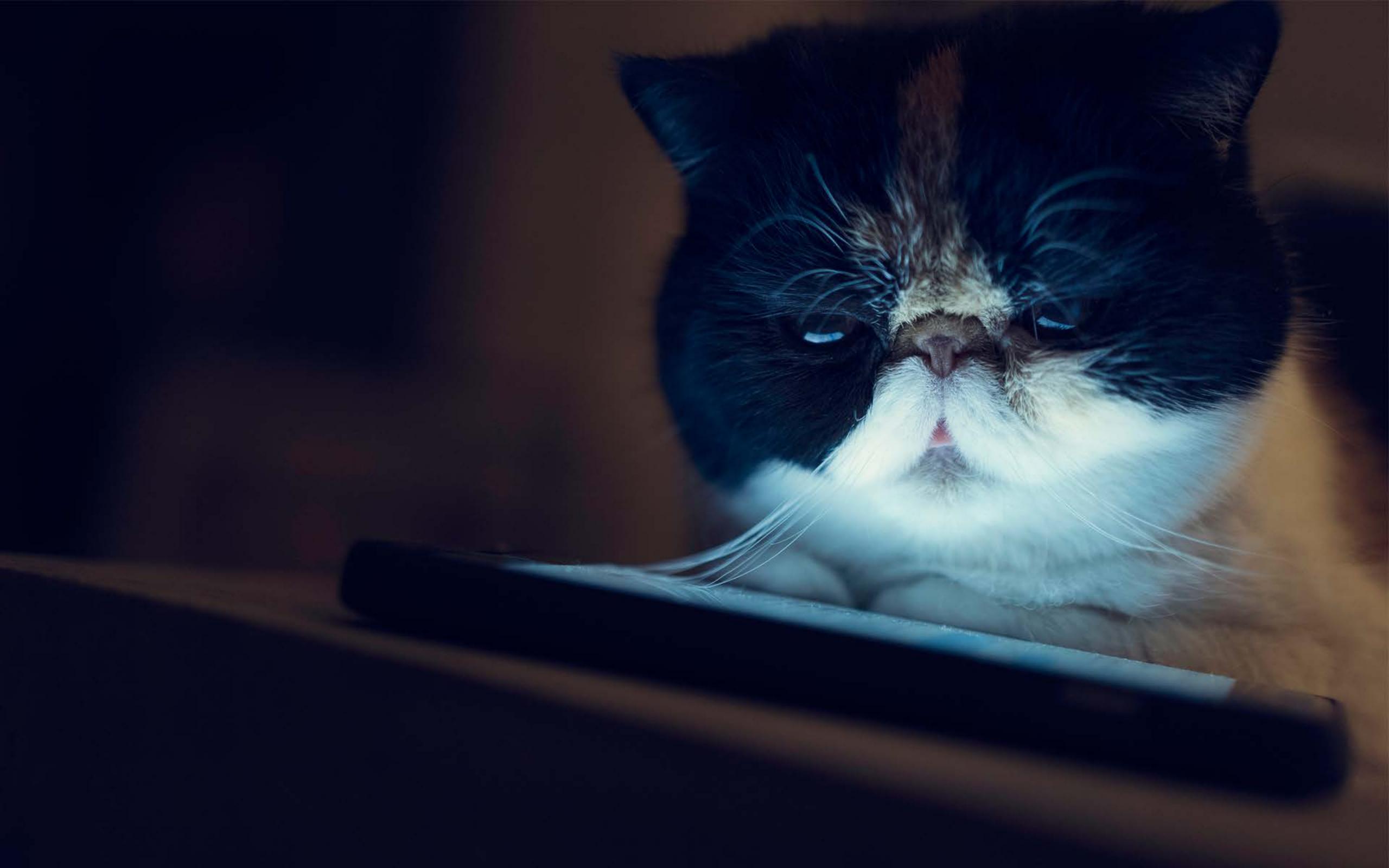
# How well are we doing?

# No iOS malware at scale

# 5-10 vulnerabilities

# \$1 million

# Security is a process, not a destination



More Information

https://developer.apple.com/wwdc16/705

#### Related Sessions

| What's New in Security             | Nob Hill        | Tuesday 5:00PM   |
|------------------------------------|-----------------|------------------|
| Engineering Privacy for Your Users | Pacific Heights | Wednesday 4:00PM |

# Labs

| Security & Privacy Lab 1 | Frameworks Lab C | Wednesday 9:00AM |
|--------------------------|------------------|------------------|
| Security & Privacy Lab 2 | Frameworks Lab B | Thursday 9:00AM  |

# ÓWWDC16