Securing Application Data

Protecting user data

John Wright
Director, Platform Technology

Agenda

- Data Protection and what it means for you
- Design of Data Protection in iOS 4
- How to enable Data Protection
- Adopting Data Protection in your application

Harpreet Singh
Of course you have the number ;-)
I practically live there!
I'll make the reservation under your name then
Can't wait to catch up
Yeah, and I want to hear about your trip
How about 7 tonight?
Sounds great
See you there!
Send
Inbox (3)
HOMETOWN BANK
Menu
Sign In
Enter Online ID
mylogin
Enter Password
Sign In
Work Info
Mega donuts
Pay Periods
Bi-weekly
Time Worked Recently
Add time worked
June 11, 2010
June 10, 2010
Remove Time Worked Thru Date
SPACEBOOK
Me
Friends
Chat
Updates
My Health - Family
Penelope Heaps
Medical Conditions
Medical History
Edit Info
Perscriptions
Appointments
Edit

Current State

- On iPhone 3GS: All data is encrypted in place
    - No performance impact
    - Fast Remote Wipe
    - Protects “data at rest”
        - If device is stolen it is vulnerable to local attack
- Also in iPhone OS 3—encrypted backups
    - Data protected when “off device”

Data Protection

- Goal: to keep data safe even if the device itself is compromised
- For iOS 4: the ability to tie the encrypted data to the passcode
    - Data is only available when the device is unlocked
    - Able to use choose data availability
    - Adopted by Mail, APIs available for adoption in your apps
    - No performance impact
- Still enables encrypted backups
    - Allows keychain migration between devices
    - No excuse to not use the keychain!

Passcodes

- Reminder: Passcodes can be brute-forced
- Mitigations:
  - Backoff between failed attempts
    - Sixth failed attempt delays 10 minutes
  - Force complex passcodes
    - User-configurable in Settings
  - Force erase after number of attempts
    - User-configurable in Settings
- For iOS 4: Data protected with combination of passcode and device keys
  - Prevents attacks off device as device limits speed of attacks

Technology	Security Mitigation
Passcode	Prevents casual device access
Privilege Separation and Sandboxing	Limits access to system or other app data if local app compromised
Code Signing	Only code of known origins can execute
Remote Wipe	Erase all data if phone is lost
Encrypted Storage	Fast Remote Wipe
Encrypted Backups	Protects data off the device
Data Protection	Protects user’s data when the device is locked

Ultimate Goal

All apps adopt Data Protection

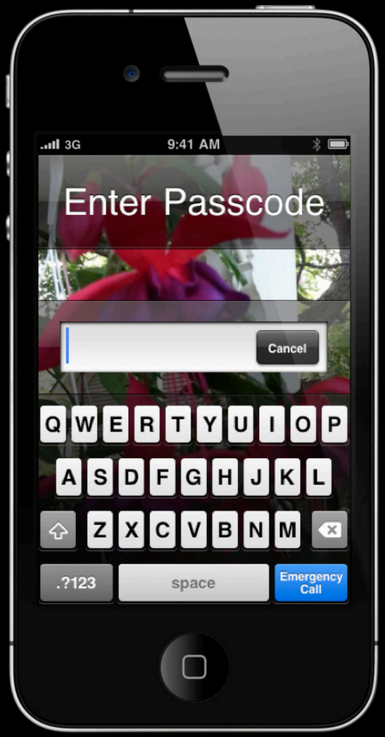
- We started with Mail
- We want you to choose how your data is protected
- Goal: Users know that their data is protected

Mechanism

How and why it works

Mitch Adler
Architect, Embedded Security

Protected When Locked



Data Protection Classes

Availability	File System	Keychain
When unlocked	`…ProtectionComplete`	`…WhenUnlocked`
After first unlock		`…AfterFirstUnlock`
Always	`…ProtectionNone`	`…Always`

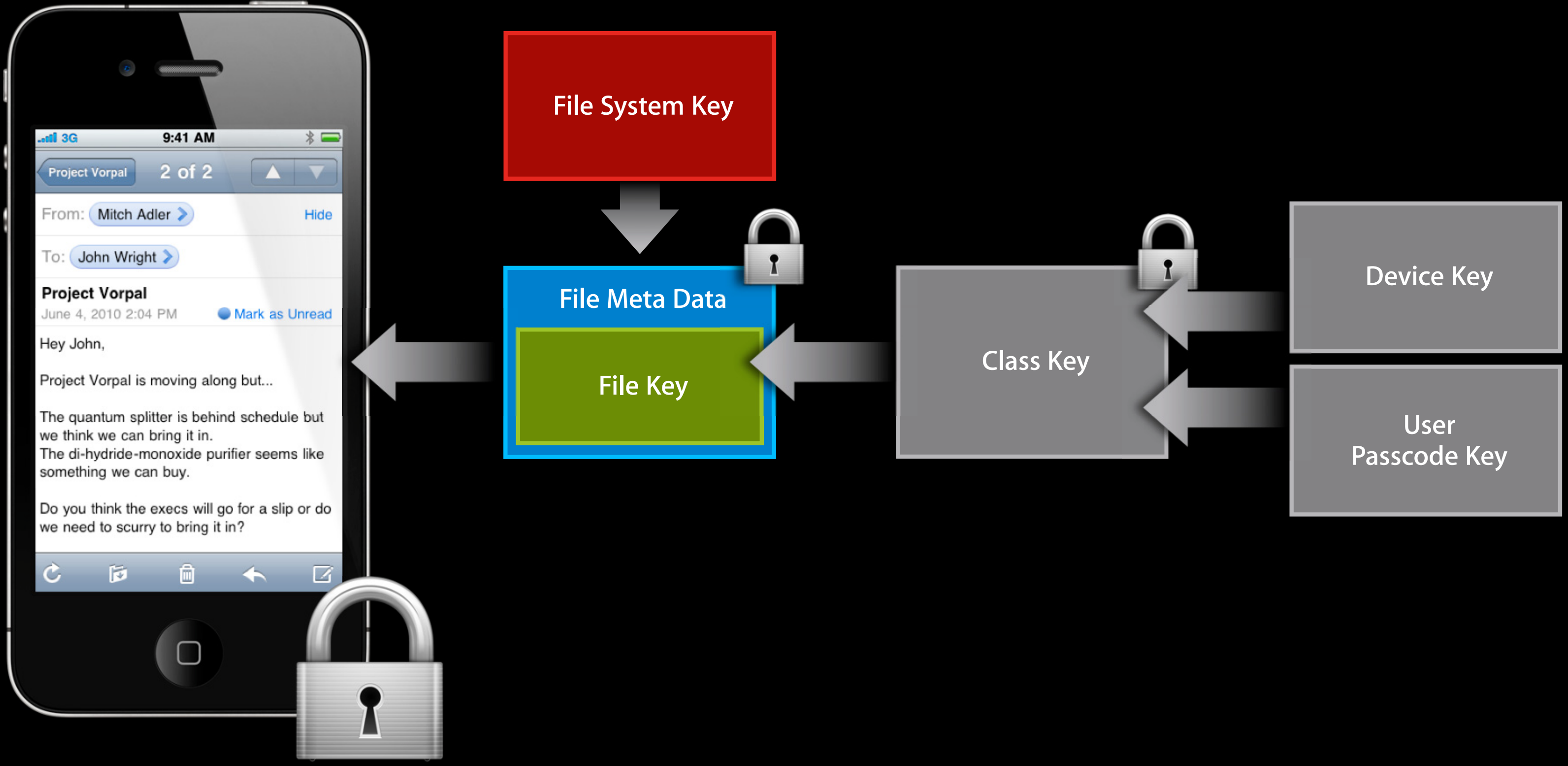
Filesystem and Keychain

Independent clients

- Intentionally similar semantics
- Separate class keys
- Isolates creation and accessing data

Protection Hierarchy Example

Protected file



3G 9:41 AM
Project Vorpal 2 of 2
From: Mitch Adler Hide
To: John Wright
Project Vorpal
June 4, 2010 2:04 PM Mark as Unread
Hey John,

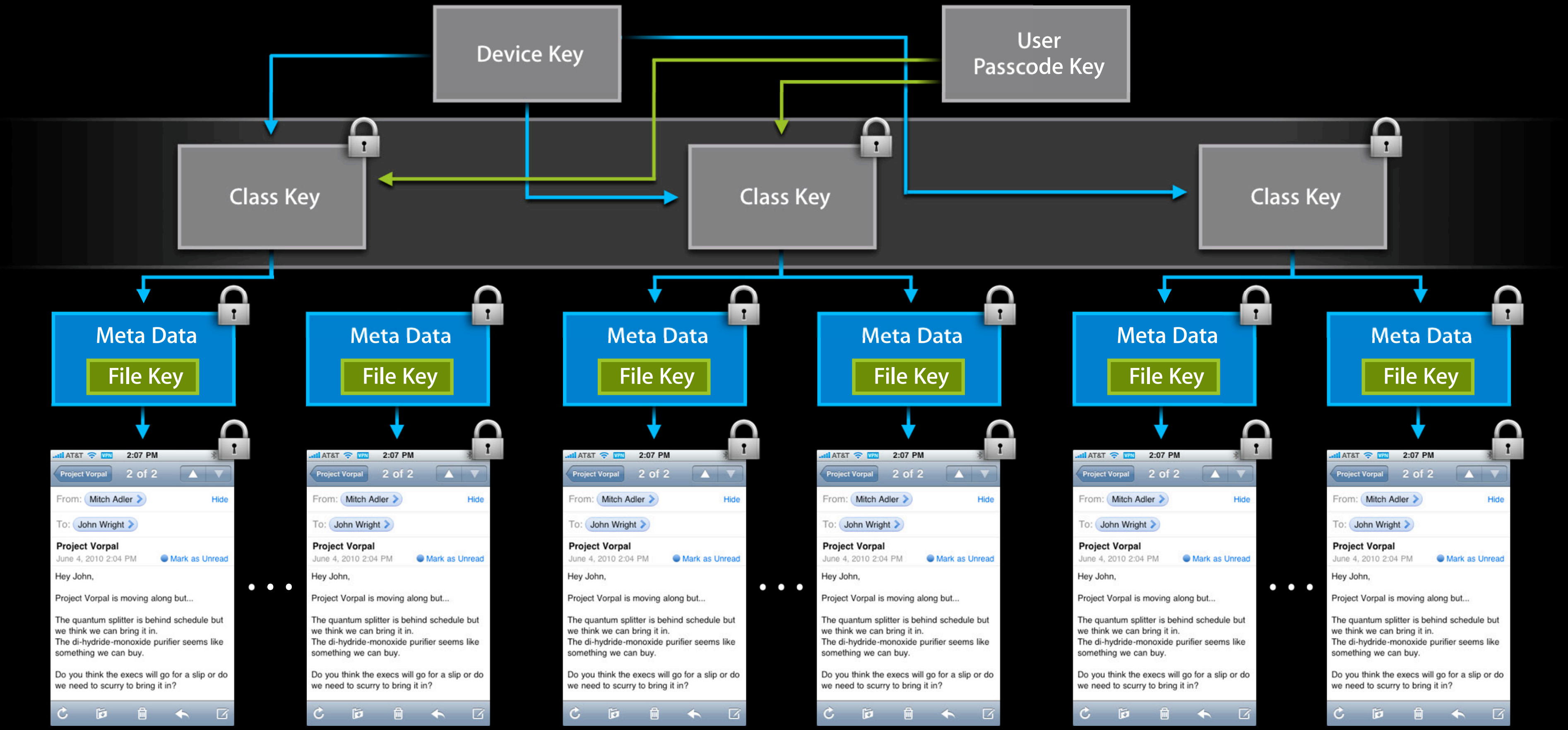
Project Vorpal is moving along but...

The quantum splitter is behind schedule but we think we can bring it in.
The di-hydride-monoxide purifier seems like something we can buy.

Do you think the execs will go for a slip or do we need to scurry to bring it in?

Protection Hierarchy

File system



From: Mitch Adler

To: John Wright

Project Vorpal

June 4, 2010 2:04 PM

Hey John,

Project Vorpal is moving along but...

The quantum splitter is behind schedule but we think we can bring it in.
The di-hydride-monoxide purifier seems like something we can buy.

Do you think the execs will go for a slip or do we need to scurry to bring it in?

Keybags

- System
- Backup
- Escrow

System Keybag

- Holds class keys for all protected data on the device
- Contained class keys are protected with the device key
- Keybag protected with a disposable key on the file system
  - Wiped on every passcode change
    - Avoid former passcodes compromising class keys
- Lock and unlock using the users passphrase
  - Making class keys available and unavailable

Backup Keybag

- Created for each backup (new keys)
- Holds the Class Keys for data in the backup
- Class Keys protected with backup password, if you have one
  - Uses PBKDF2 and backup password to derive key
  - If not, keychain keys are protected with device-key and cannot migrate

Escrow Keybag

- Created for external systems
  - Improve user experience for syncing
  - Reduced device protection if synced systems are compromised
- Contains copies of system keybag class keys
- Class Keys protected with the device-key and generated key
  - Only useful with a particular device
  - Generated key can be forgotten to invalidate the keybag

Filesystem Protection

- All Filesytem data is encrypted
- Metadata uses File System Key
  - Destroyed on remote wipe
- Files encrypted to file key
  - Stored in Metadata protected by Class Key
  - default is …ProtectionNone
- No performance impact

Keychain Protection

- Data protected by classes just for keychain
  - Keychain adds migratable and non-migratable classes
  - By default data is …Always
    - Which is migratable

Data Protection Classes

With migration attributes

Availability	File System	Keychain
When unlocked	`…ProtectionComplete`	`…WhenUnlocked` `…WhenUnlockedThisDeviceOnly`
After first unlock		`…AfterFirstUnlock` `…AfterFirstUnlockThisDeviceOnly`
Always	`…ProtectionNone`	`…Always` `…AlwaysThisDeviceOnly`

User Passcode Handling

- Passcode transformed into key by mixing with hardware key
  - Strengthens passcode against brute force
  - Example: iPhone 4 uses 50,000 hardware operations to derive, ~50mS
- Passcode and User Passcode Key erased after unlock
  - Only exposed class keys are retained during unlock

Data Protection Classes

Device boot

Availability	File System	Keychain
When unlocked	`…ProtectionComplete`	`…WhenUnlocked` `…WhenUnlockedThisDeviceOnly`
After first unlock		`…AfterFirstUnlock` `…AfterFirstUnlockThisDeviceOnly`
Always	`…ProtectionNone`	`…Always` `…AlwaysThisDeviceOnly`

Data Protection Classes

Passcode entered

Availability	File System	Keychain
When unlocked	`…ProtectionComplete`	`…WhenUnlocked` `…WhenUnlockedThisDeviceOnly`
After first unlock		`…AfterFirstUnlock` `…AfterFirstUnlockThisDeviceOnly`
Always	`…ProtectionNone`	`…Always` `…AlwaysThisDeviceOnly`

Data Protection Classes

Device locked

Availability	File System	Keychain
When unlocked	…ProtectionComplete	…WhenUnlocked …WhenUnlockedThisDeviceOnly
After first unlock		…AfterFirstUnlock …AfterFirstUnlockThisDeviceOnly
Always	…ProtectionNone	…Always …AlwaysThisDeviceOnly

Summary

- Protection extends through the keybag
  - From user secrets and device keys
  - To Filesystem and Keychain
- Passcode key derivation resistant to brute-force attacks
- Keychain contents migratable via protected backups

Configuration for Data Protection

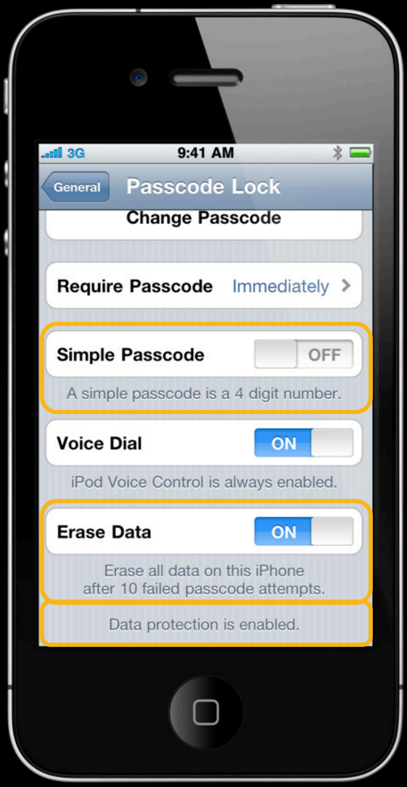
Which knobs to turn

John Wright
Director, Platform Technology

Enabling Data Protection

By the end user

- To enable on any iPhone 4:
  - Simply set the user passcode on the device
- To enable on any iPhone 3GS or iPod touch (3rd gen):
  - Must do an Erase Install first
- Coming to the iPad with iOS 4
- To check if Data Protection is enabled:
  - Go to Settings -> General -> Passcode Lock



Change Passcode

Require Passcode Immediately

Simple Passcode OFF

A simple passcode is a 4 digit number.

Voice Dial ON

iPod Voice Control is always enabled.

Erase Data ON

Erase all data on this iPhone after 10 failed passcode attempts.

Data protection is enabled.

Enabling Data Protection

For managed devices

- Configuration Profiles can enforce security requirements:
  - Require Data Protection is enabled
  - Require passcode length and complexity
  - Require maximum passcode grace period
  - Require encrypted backups
- Mobile Device Managment
  - Initial device configuration and automatic updates
  - Remote Wipe capabilities

Configuration

- Simple for users
- Fully manageable by enterprises

Adoption

Adding data protection to your application

Mitch Adler
Architect, Embedded Security

Protect Your Customer's Data

- Assume the data needs protection
- Put data into the right class
  - The most protective class you can
- Purge user data when the device locks

Analyzing Data for Protection

- What data of the user’s do I have?
- Do I need this when the device is locked?
- Does it belong on the keychain?
- Should it move from device to device?

What Data of the User’s Do I have?

- Everything they create
- And credentials they enter
- Credentials you create on their behalf

Do I Need This When the Device Is Locked?

- No, for most
- Some background-aware applications need credentials
  - Use `kSecAttrAccessibleAfterFirstUnlock`

Does This Belong on the Keychain?

- Credentials—particularly those that migrate
    - Passwords
    - Keys
    - Identities
- Large bulk data: no

Should It Move from Device to Device?

- Yes
- Unless you have a compelling reason not to
  - e.g., device identities for VPN

Protection Choices Example

Game

	When locked	Migrate	Keychain	Protection
Online Play Credentials	No	Yes	Yes	`kSecAttrAccessibleWhenUnlocked`
Saved Games	No	Yes	No	`NSFileProtectionComplete`

Protection Choices Example

Online Service

	When locked	Migrate	Keychain	Protection
Service Credentials	Background Task Completion	Yes	Yes	`kSecAttrAccessibleAfterFirstUnlock`
Data being uploaded	Yes	No	No	`NSFileProtectionNone`
Cached Data	No	No	No	`NSFileProtectionComplete`

Protection Choices Example

Diary

	When locked	Migrate	Keychain	Protection
Diary Entries	No	Yes	No	`NSFileProtectionComplete`

Device Locking

What your application needs to do

- Handle Notification
  - `UIApplicationProtectedDataWillBecomeUnavailable`
- In response to notification
  - Purge data in memory not necessary while locked
    - Close files, delete memory copies
  - 10s limit to complete handling
- Protected data access after handling will fail
  - Be resilient to these failures—don’t crash

Filesystem

Interfaces

- `-[NSData writeToFile:options:error:]`
  - Creates and associates with protection class
  - Options:
    - `NSDataWritingFileProtectionComplete,` `NSDataWritingFileProtectionNone`
- `-[NSFileManager setAttributes:ofItemAtPath:error:]`
  - Change the class of a file
  - Attribute: NSFileProtectionKey
    - `NSFileProtectionComplete, NSFileProtectionNone`
- Protection class persists and is not disturbed by other access
- Update protection class for existing files

Filesystem

Protection selection

- Use
  - `...ProtectionComplete`
- Until you find a good reason to not protect your user’s data

Keychain

Interfaces

- Select protection via SecItem API attributes
    - SecItemAdd, SecItemUpdate
    - Using attributes:
        - `kSecAttrAccessibleWhenUnlocked`
        - `kSecAttrAccessibleAfterFirstUnlock`
        - `kSecAttrAccessibleAlways`
        - `kSecAttrAccessibleWhenUnlockedThisDeviceOnly`
        - `kSecAttrAccessibleAfterFirstUnlockThisDeviceOnly`
        - `kSecAttrAccessibleAlwaysThisDeviceOnly`
- Update existing data to appropriate class
    - Existing data defaults to kSecAttrAccessibleAlways

Keychain

Protection selection

- Start with most restrictive but migratable
  - Use …`AccessibleWhenUnlocked`
- If you find reasons to access while locked
  - Use …`AccessibleAfterFirstUnlock`
- If you create data for this device only
  - Be sure it's really only for this device
  - Use …`ThisDeviceOnly` variants

Example

Adding protection attributes

```
CFMutableDictionaryRef query = CFDictionaryCreateMutable(NULL, 0, NULL, NULL);

CFDictionarySetValue(query, kSecClass, kSecClassGenericPassword);
CFDictionarySetValue(query, kSecAttrAccount, account);
CFDictionarySetValue(query, kSecAttrAccessible, kSecAttrAccessibleWhenUnlocked);
CFDictionarySetValue(query, kSecValueData, pwdata);

SecItemAdd(query, NULL);

CFRelease(query);
```

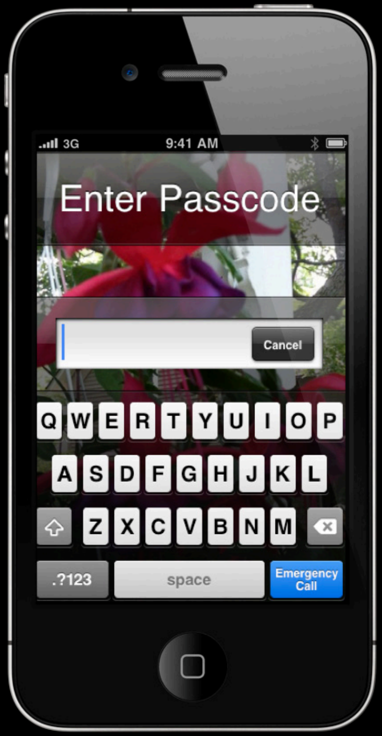
Adoption

- Protect your user's data
  - Choose the strongest protection class you can
  - Update existing data's protection
    - Filesystem and Keychain
- Use the keychain
  - Supports migration now—last missing feature

Wrap-up

Why

- Devices are more mobile
- Mobile devices are more active
- Data is more personal
  - More data, higher value
- Tie data to user secret



Data Protection

- Protect data with passcode
  - Strengthen against compromise
- Passcode key tied to hardware
  - Strengthen against brute force
- Always on when passcode enabled
- No performance impact

Adoption

Protect your customer’s data

- In all applications
  - User information is everywhere
- We’re adopting it—Mail first
  - More to come!
- Put data into appropriate classes
  - Update your existing data
- Use the Keychain for credentials!

More Information

Paul Danbold
Evangelist
danbold@apple.com

Apple Developer Forums
http://devforums.apple.com

Related Sessions

Adopting Multitasking on iPhone OS, Part 1	Presidio Tuesday 11:30AM
Adopting Multitasking on iPhone OS, Part 1 (R)	Marina Friday 9:00AM
Adopting Multitasking on iPhone OS, Part 2	Mission Tuesday 3:15PM
Adopting Multitasking on iPhone OS, Part 2 (R)	Marina Friday 10:15AM
Managing Mobile Devices	Nob Hill Tuesday 3:15PM

Labs

iPhone OS and Mac OS X Security Lab	Core OS Lab A Thursday 2:00PM

WWDC10