EC440 Operating Systems – Mobile OSes & Security

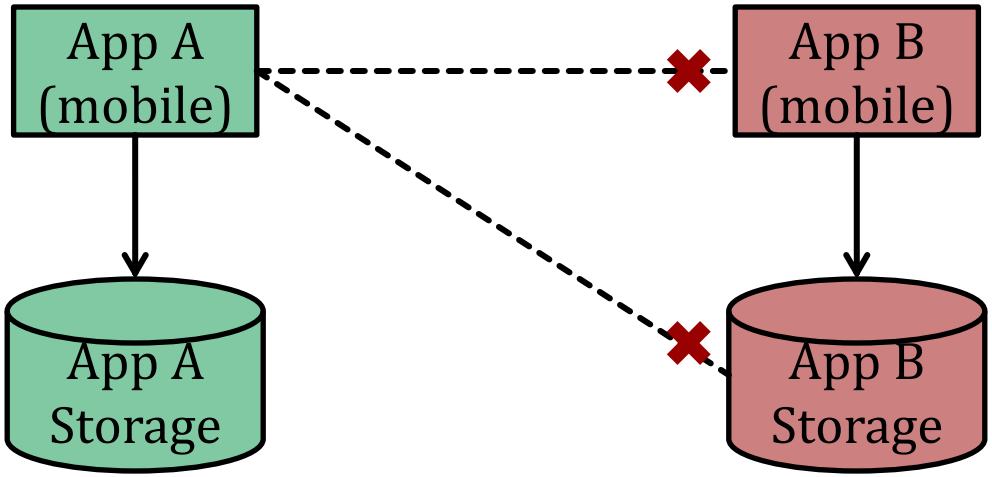
Manuel Egele
PHO 337
megele@bu.edu
Boston University

Slides from William Robertson @ NEU

Apple iOS

- Operating system on iPhone, iPad, iPod
  - UNIX-like (XNU/Darwin)
- Apps written in Objective-C/Swift
  - Native code
- Apps have restricted API and access to the underlying system
  - No Java, Flash
  - Common system utilities not present

iOS Security Architecture



- Privilege separation
  - Built on UNIX security model
  - Most processes run as the `mobile` user
  - System daemons run under other least-privilege protection domains
- Apps are siloed
  - Little to no sharing outside of system apps

iOS Security Architecture

- DEP, ASLR, sandboxing
- Code signing framework
  - All binaries, libraries must be signed by Apple
  - All apps must originate from the App Store (exceptions: provisioning, jailbreaking)
- Curated app store model
  - Apps placed in the App Store are trusted
  - "Inspected" by Apple prior to publishing

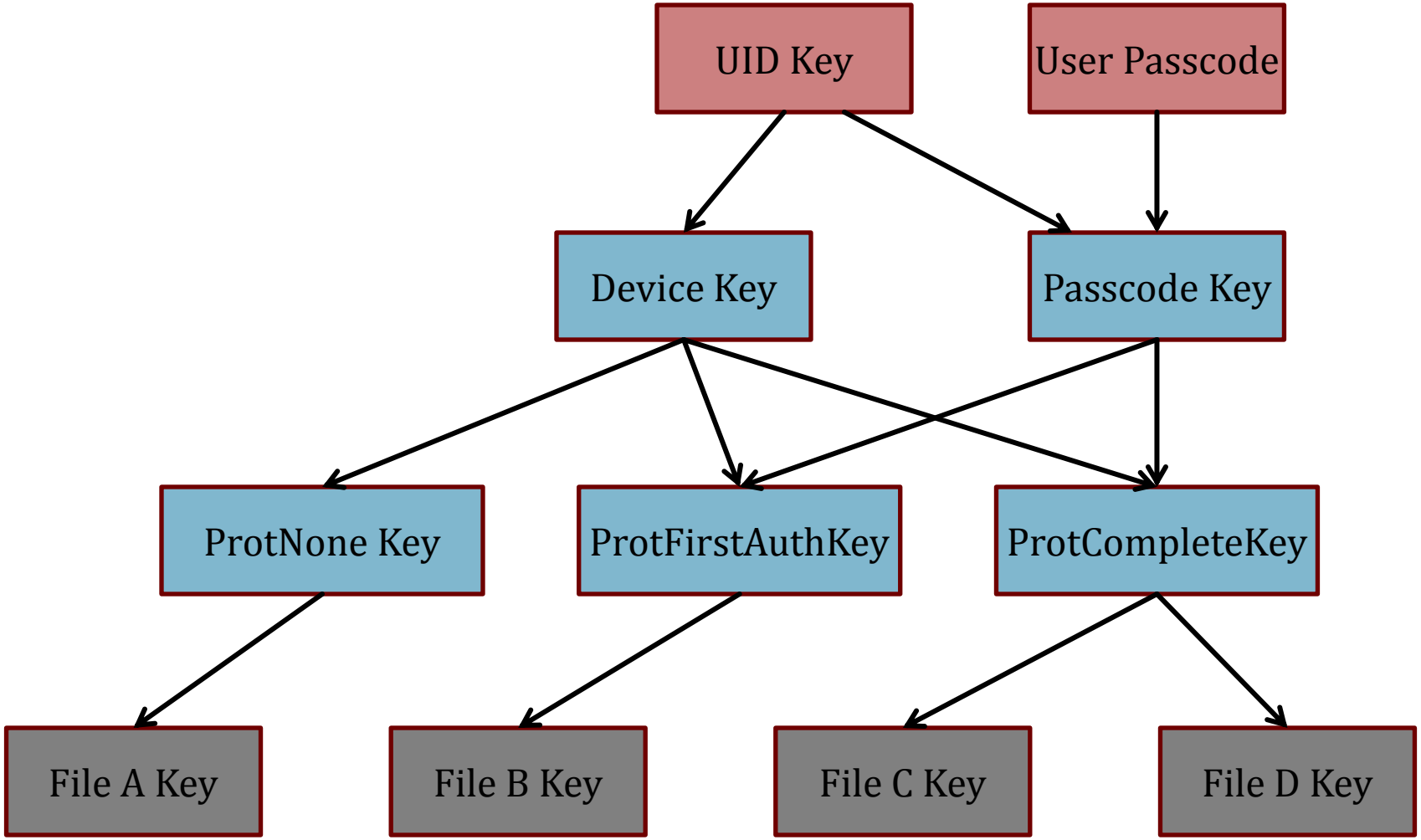
Storage Encryption

- Mobile devices are at a higher risk of confidentiality breaches
  - Easier to lose or steal a phone than a desktop
- Mobile devices present unique challenges
  - Difficult to perform pre-boot authentication
  - Limited user input capabilities

Data Protection API

- iOS provides an API for encrypting stored files
- Provides a number of protection classes
  1. File is protected, only accessible when device unlocked
  2. File is protected, accessible after device unlocked
  3. File is protected until user passcode entered
  4. File is not protected
- Keys derived from device UID key and user passcode using (modified) PBKDF2
  - UID key resides in device crypto accelerator, never released

Data Protection Keys



Data Protection API

- Passcode key
  - Derived from user passcode
- UID key can only be used (not read) while the phone is running
  - Prevents offline attacks
  - Ensures passcode key is unique for different devices even if passcode is the same
- Brute-force mitigations
  - Incorrect guesses exponentially increase delay between checks
  - Devices can be configured to erase user data after too many incorrect guesses
  - **These defenses are implemented in the UI!**

Brute-Forcing User Passcodes

Length	Complexity	Time
4	Numeric	18 minutes
4	Alphanumeric	19 days
6	Alphanumeric	196 years
8	Alphanumeric	755k years
8	Alphanumeric (complex)	27M years

On-device expected passcode brute-forcing times for the iPhone 4

Permission Dialogs

- iOS apps must obtain permission to access potentially sensitive user data
- iOS uses a *time-of-use* permissions model
  - System dialog prompts user to grant or deny access at runtime
  - App must be able to handle access denials

Code Signing

```
#if CONFIG_MACF
error = mac_vnode_check_exec(
    imgp->ip_vfs_context, vp, imgp);
if (error)
    return (error);
#endif
```

- All iOS apps must pass signature verification before they can run
  - Implemented as an extension of TrustedBSD MAC framework
  - AppleMobileFileIntegrity (AMFI) policy
- AMFI hooks inserted on key kernel code paths
  - Allows kernel to verify app signatures before execution

Provisioning

- All system apps must be signed by Apple's private key to execute
  - But, enterprises and universities often want to distribute their own apps outside of the App Store
- Provisioning provides these capabilities
  1. Apple signs certificates provided by developers
  2. Apple signs a provisioning profile that references developer certs
  3. Users install provisioning profile
  4. Device allows apps signed by developer's key to run according to the installed profile
- Similar process used for self-signed developer certs used during development

Code Signing Enforcement

```
#if CONFIG_MACF
error = mac_vnode_check_signature(
    vp, blob->csb_sha1, (void *) addr, size);
if (error)
    goto out;
#endif
```

- Enforcement implemented within the kernel virtual memory subsystem
  - At executable load time, the image as a whole is checked against the embedded signature
  - System checks whether individual memory pages originated from signed code at runtime

Page Signature Checks