HOW SECURE IS YOUR IPHONE?

An introduction to iOS security and forensics

Thursday, July 10th

ICTF 2014

AGENDA

- What to expect and why
- What is stored?
- Overview of iOS security features
- iOS structure and boot process
- Introduction to iOS forensics

MOTIVATION

- BYOD hype
- ICO
- Encryption
- Personal interest
- Sharing is caring!

CREDITS, CAVEATS AND DISCLAIMERS

- I use an iPhone
- Not a comparison with other mobile operating systems
- I'm not **the** expert (specific credits at the end)
- Not an endorsement of any tools or techniques used
    - be aware of legal responsibilities
    - potentially untrusted software/tools
- Not time to cover everything
- **Not** a forensics workshop

WHAT IS STORED?

- Obvious things like email, data in apps etc. that may contain personal or otherwise confidential data.
- Encrypted passwords to websites, wireless access points etc. which are stored in the keychain
- Nearly everything typed into the iPhone's keyboard is stored in a keyboard cache to which multiple copies can linger after deleted
- Pasteboard is persistent - reboot and try it!
    - Must be stored somewhere
- Screen shots are preserved of the last state of an application taken when the home button is pressed. Can include browser snapshots, SMS messages, contacts, maps and recent call lists
- Exhaustive call display, beyond that displayed generally available – approximately last 100 stored in call database and can be recovered using SQLlite.

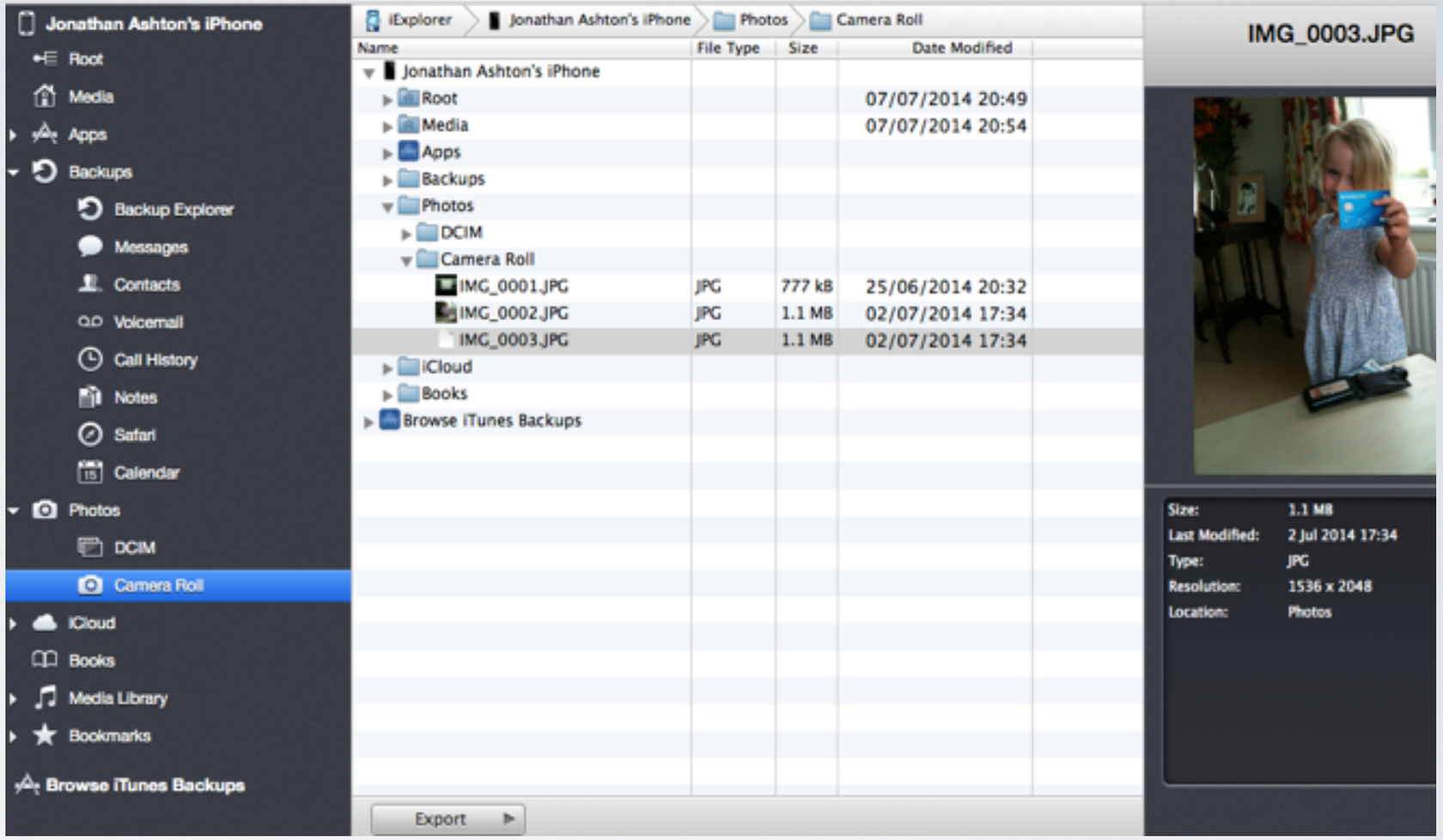
WHAT IS STORED?

- Map tile images, direction lookups, longtitude/latitude coordinates of map searches
- Cached location of towers,access points and coordinates of where devices has been seen
- Browser history and saved browser objects identifying websites visited
- Voicemail recordings
- Wi-fi pairing records
- Pairing records between devices and desktop computers

WHAT IS STORED - DELETED DATA?

- Deleting data from SSD is difficult!
- Cached and deleted emails, SMS messages and other forms of correspondence
- Deleted images from the photo library, camera roll and browsing and email store can be recovered using a data carving tool. Movies and music can also be recovered. Images taken with the device may be geo-tagged containing GPS coordinates
- Deleted address book, contacts, calendar events an other personal data can often be found in fragments on disk
- Deleted call lists

DATA ACQUISITION



- iTunes backup – will need some tool to correctly query the data e.g. iPhone Analyzer
- Acquisition via logical means – e.g. iPhone iExplorer
- Acquisition via physical means – i.e. bit by bit copy
- Acquisition by Jailbreaking

IOS SECURITY OVERVIEW

- System security
- Encryption and data protection
- App security
- Network security
- Internet Services
- Device controls

APP SECURITY

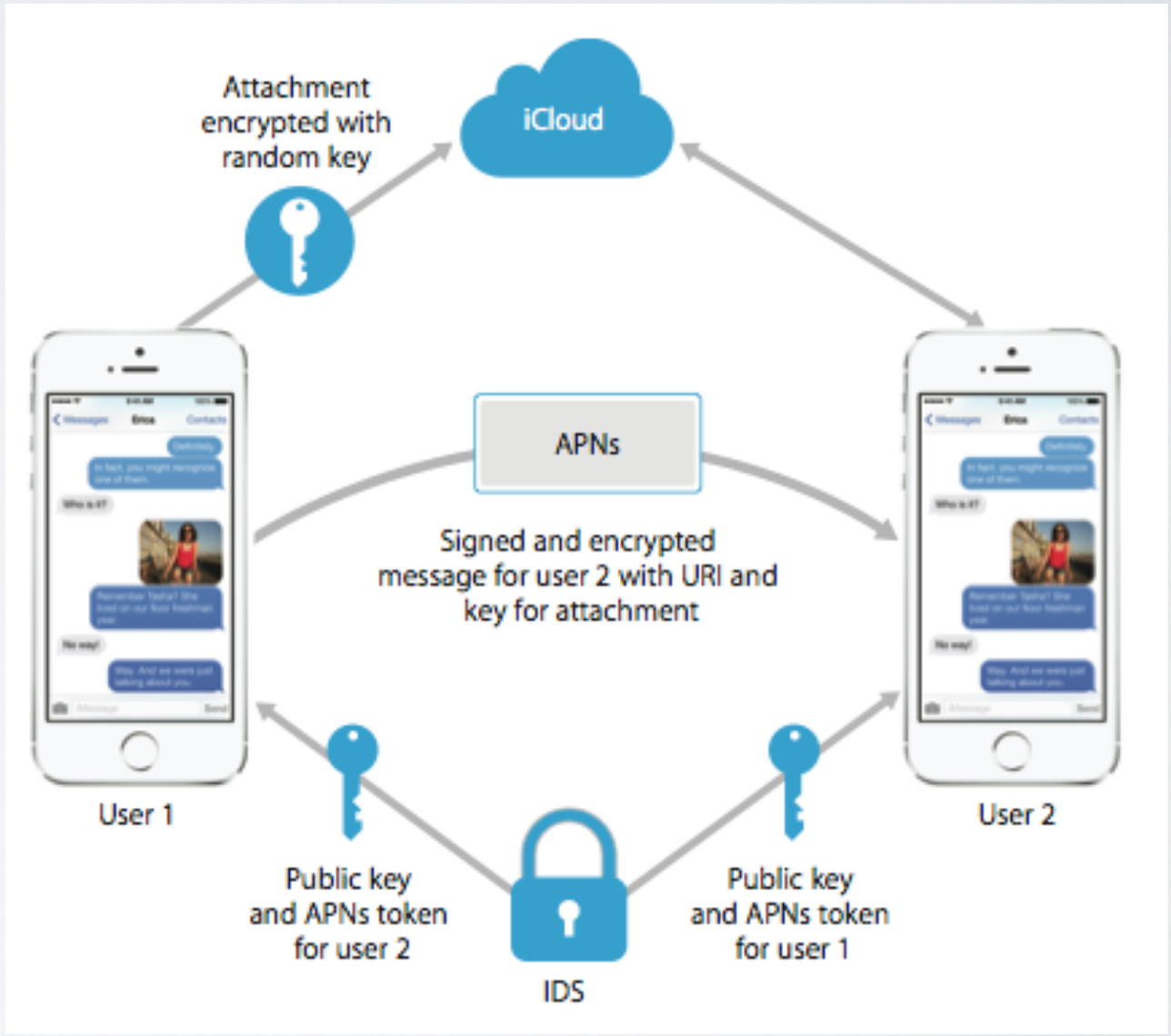
- Code signing
- Sandboxing
- Privilege separation
- Entitlements
- Address space layout randomisation
- Data execution prevention
- Data protection (covered later)

NETWORK SECURITY

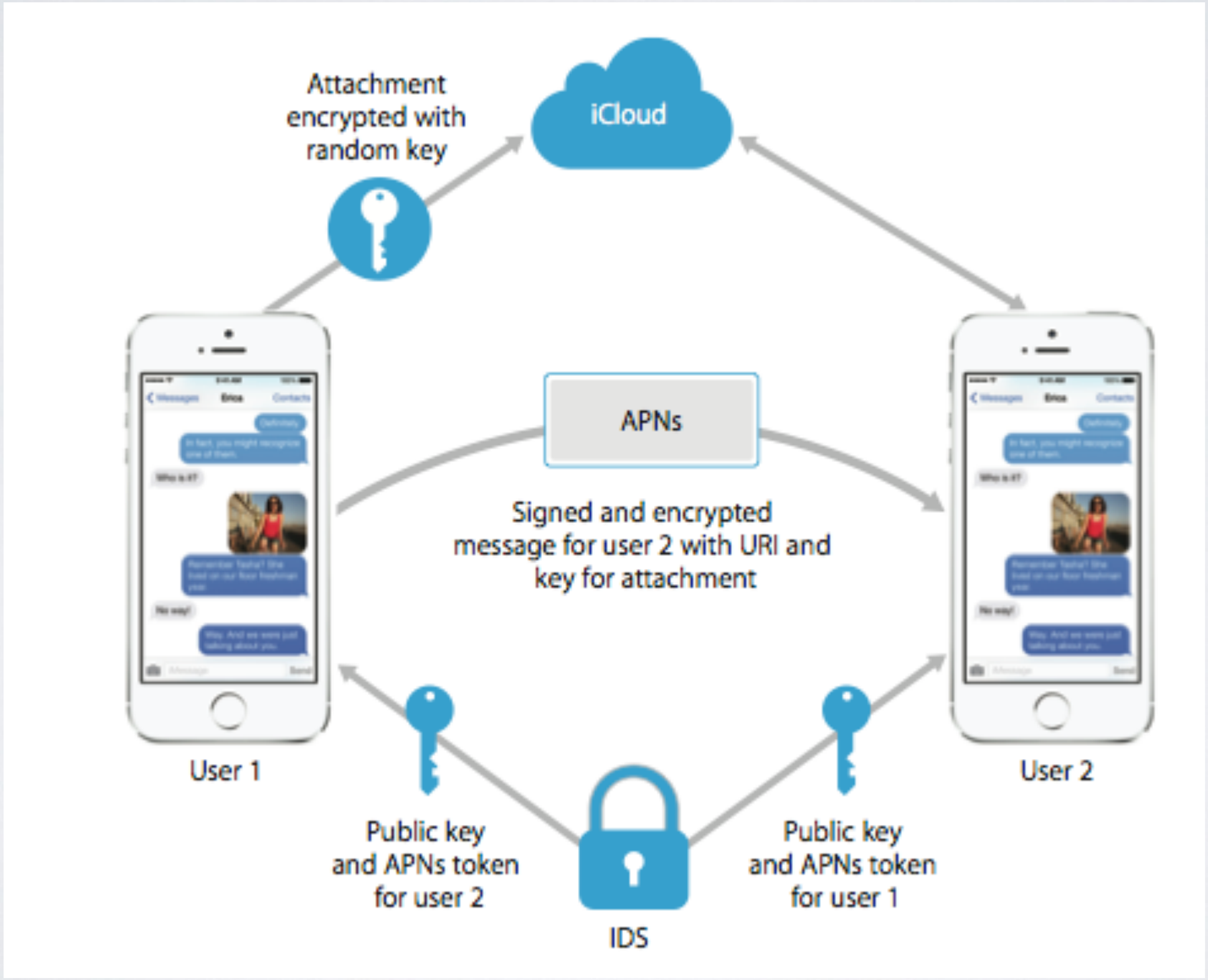
- No personal firewall
- Support for SSL v3 as well as Transport Layer Security (TLS v1.0, TLS v1.1, TLS v1.2) and DTLS
- Various options for VPN
- iOS 7 introduces notion of per-app VPN (MDM can specify connection for managed apps and/or specific domains in Safari)
- Beware of vulnerabilities
- How does your app handle secure communications?

INTERNET SERVICES

- iMessage
- Face Time
- Siri
- iCould
- iCloud Backup
- iCloud Keychain



EXAMPLE: IMESSAGE

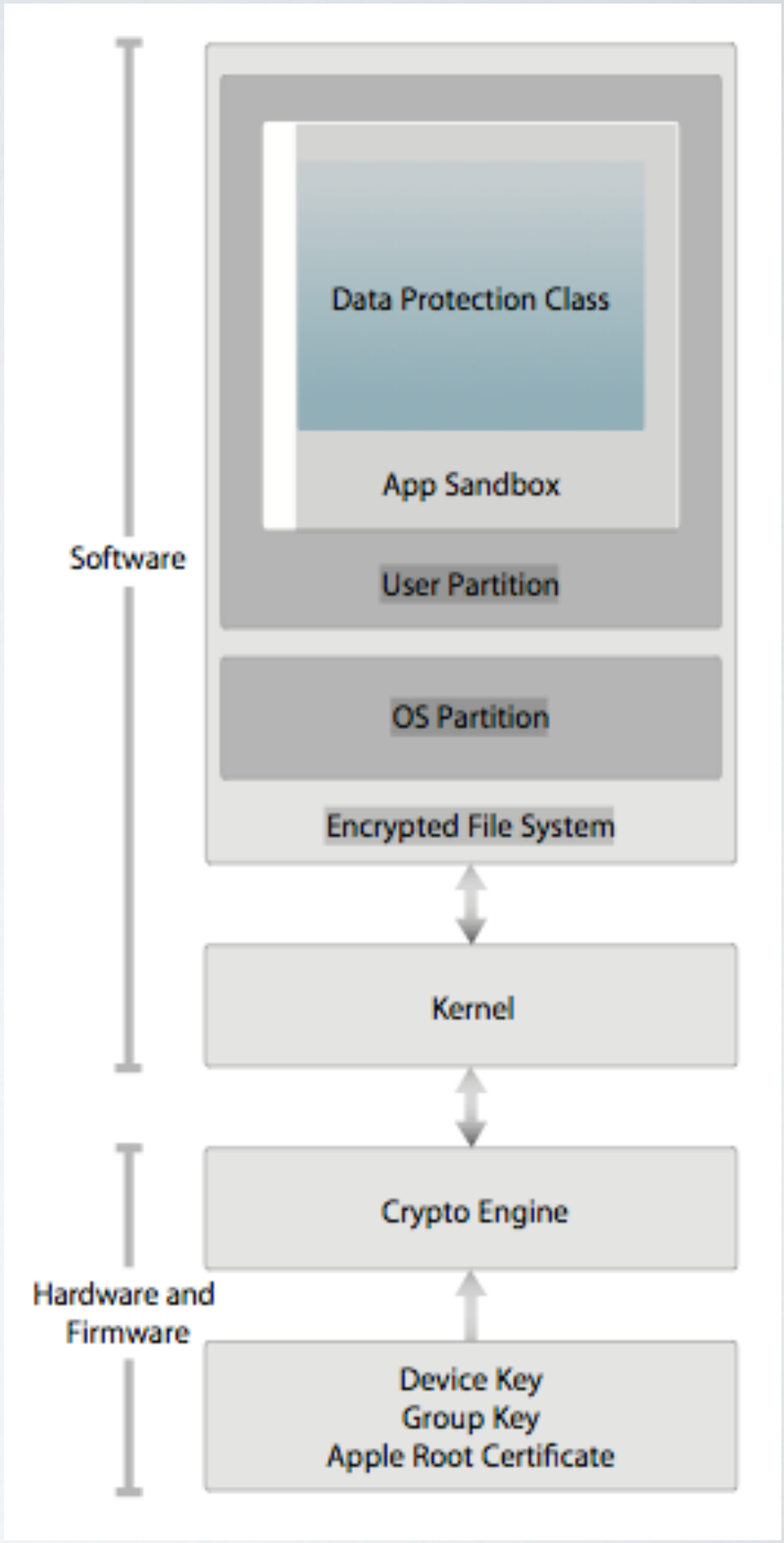


OTHER DEVICE CONTROLS

- Passcodes - we'll come to those
- Configuration Profiles
- Mobile Device Management
- Apple Configurator
- Restrictions
- Remote Wipe

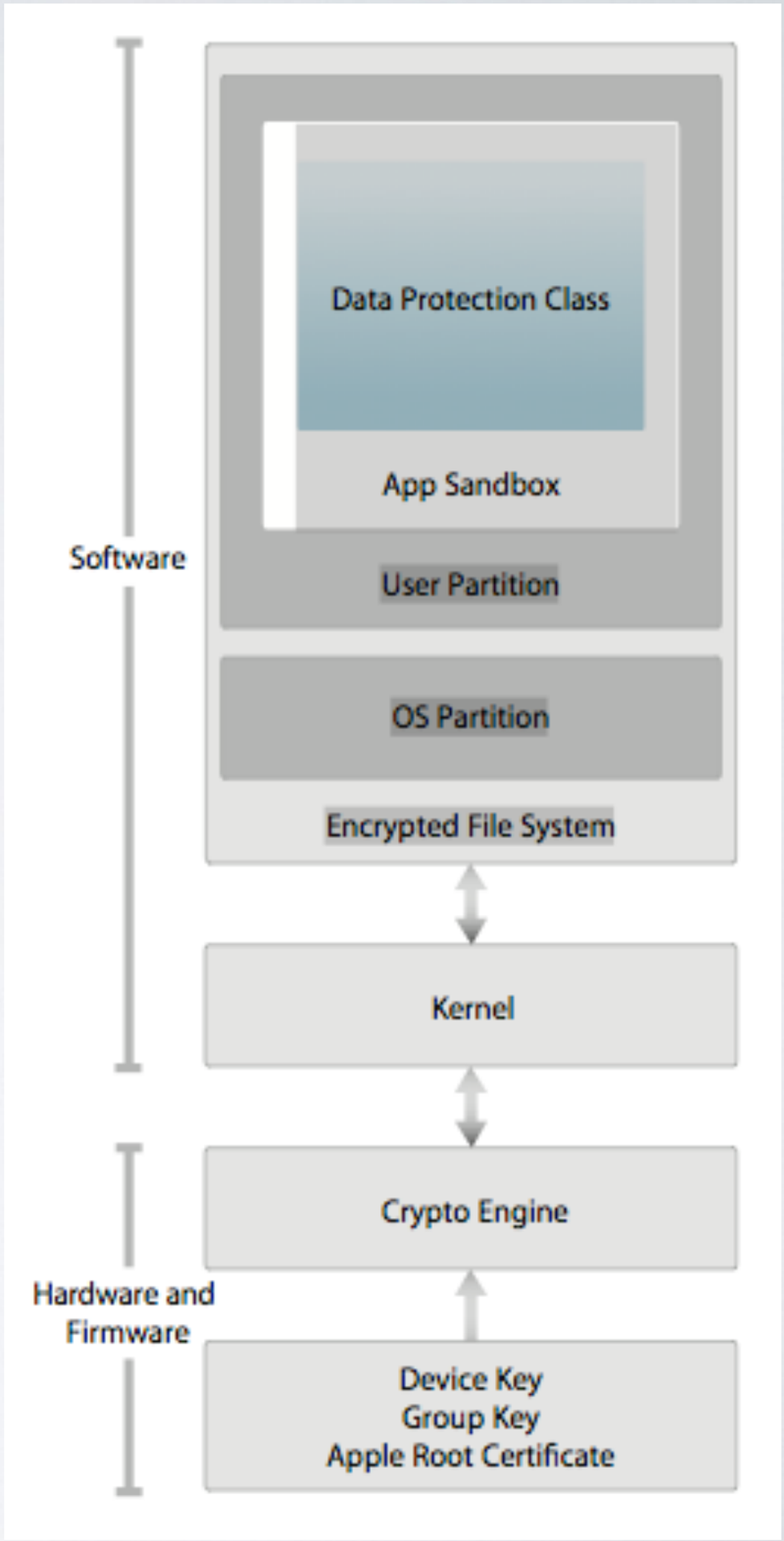
SYSTEM SECURITY

- iOS is basically stripped down version of regular OSX
- Same Unix backend, related frameworks and libraries
- By default does not contain many of the command-line tools found on the desktop version
- Number of hardening techniques (already mentioned)



SYSTEM SECURITY

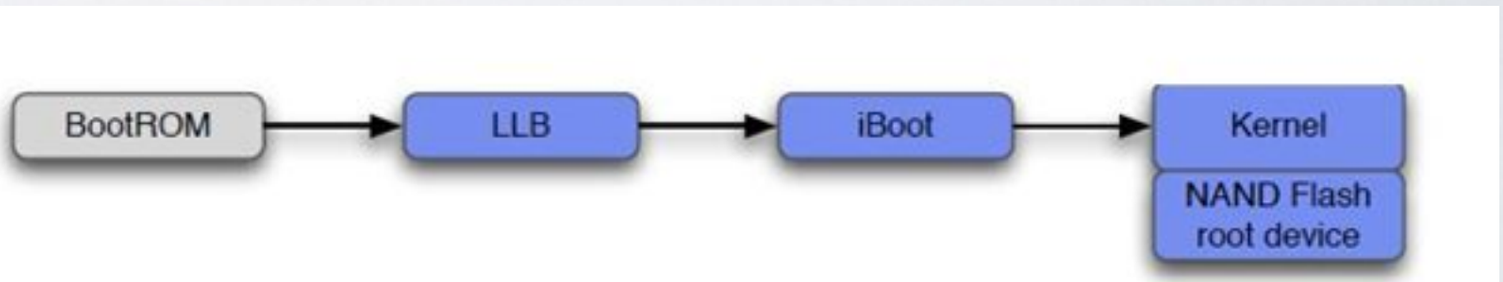
- Solid State NAND chips act as hard drive
- File system treated as two logical partitions
  - Firmware partition - read-only unless firmware upgrade taking place in which case overwritten by iTunes.
- Data partition
- Most convenient way to provide updates without affecting user data



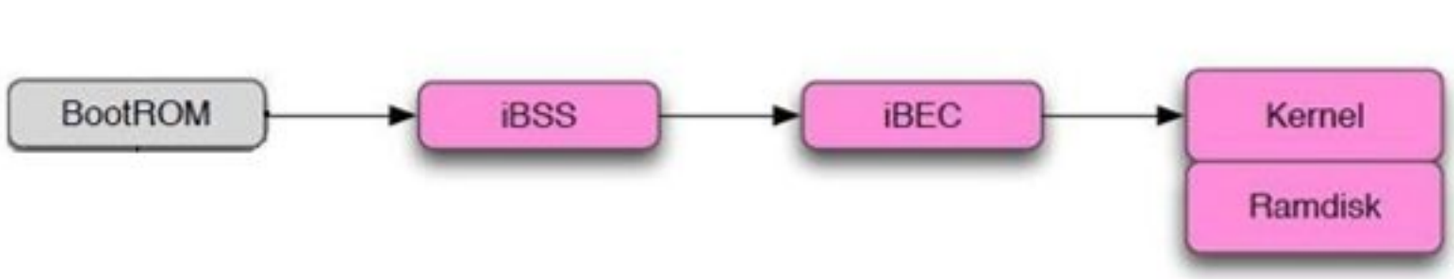
SYSTEM SECURITY: BOOT PROCESS

- When iOS device boots it runs through a chain of trust to ensure all components are cryptographically signed

- iOS devices operate in 3 modes
  - normal mode
  - recovery mode
  - DFU mode



(Figure 1) (copied from Sogeti presentation)



(Figure 2) (copied from Sogeti presentation)

BYPASSING SECURE BOOT PROCESS

- Create custom Ramdisk and load it into volatile memory
- Signature checks mean boot process doesn't allow custom Ramdisk to load
- To load it we have to bypass signature checks (compromise one link and all others are compromised)
- Vulnerabilities in BootROM exploited to flash own boot loader and patch all other signature checks
- Encryption keys for various stages can be gotten from jailbreaking tools

BYPASSING SECURE BOOT PROCESS

- Boot via patched kernel and custom ramdisk a bit like booting from a "live CD"
- Establish communication with the device via USBMUX
  - Protocol used by iTunes to talk to the booted iPhone and coordinate access to its iPhone services by other applications
  - USB multiplexing provides TCP-like connectivity over a USB port using SSL
  - Over this channel iTunes uses AFC service to transfer files but here we use the channel to establish an SSH session and get a shell on the device

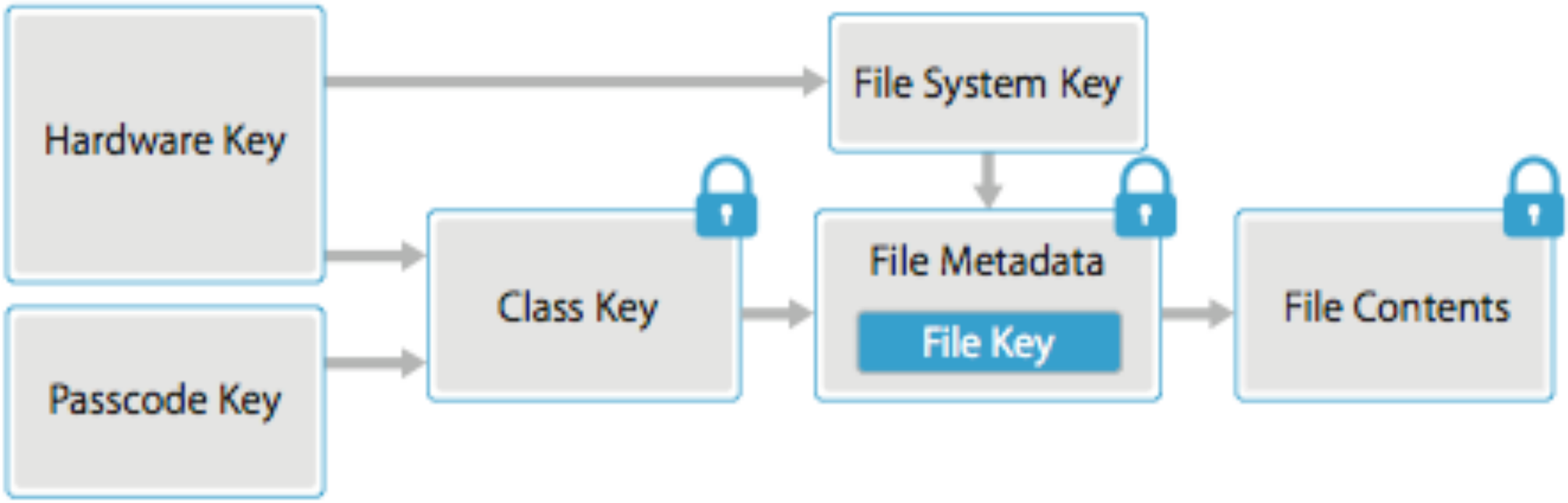
```
Jonathans-MacBook-Pro-3:~ jashton$ nc -vv localhost 2222
nc: connectx to localhost port 2222 (tcp) failed: Connection refused
found 0 associations
found 1 connections:
     1: flags=82<CONNECTED,PREFERRED>
        outif lo0
        src 127.0.0.1 port 49497
        dst 127.0.0.1 port 2222
        rank info not available
        TCP aux info available

Connection to localhost port 2222 [tcp/rockwell-csp2] succeeded!
SSH-2.0-OpenSSH_5.2
^C
Jonathans-MacBook-Pro-3:~ jashton$ ssh -p 2222 root@localhost
root@localhost's password:
-sh-4.0#
-sh-4.0#
```

DEMO

ENCRYPTION

- AES 256 crypto engine
- UID and GID fused into processor at manufacture
- Other keys generated by RNG
  - GID – Group ID key
  - UID – Unique per device key
  - Dkey – Default File Key
  - EMF – encrypts entire file system and HFS journal
  - Class keys – one per protection class
- Since 3GS and iOS4 data partition is hardware encrypted

DATA PROTECTION



Looks good?

KEY CLASSES

- Complete Protection
    - Data will be protected 10 seconds after device is locked and not accessible again without PIN.
- Protected Unless Open
    - For files that need to be written while the device is locked e.g. mail attachment downloading in the background.
- Protected Until First User Authentication
    - Similar to complete protection but key remains when device is locked. Bit like protection provided by WDE.
- No Protection
    - Protected only with UID – only affords benefit of quick remote wipe.

KEYCHAIN AND KEY BAGS

- iOS keychain provides secure way to store passwords etc. Implemented as an SQLlite database

- Protected in a similar way to file Data Protection classes using different keys

- System Keybag used to store wrapped class keys

- Unwrapped when the PIN is entered

- All very well but……

DEFEATING DATA PROTECTION

```
1110 of 10000 ETA:  0:52:23
1120 of 10000 ETA:  0:52:20
1130 of 10000 ETA:  0:52:17
1140 of 10000 ETA:  0:52:13
1150 of 10000 ETA:  0:52:10
1160 of 10000 ETA:  0:52:06
1170 of 10000 ETA:  0:52:03
1180 of 10000 ETA:  0:52:00
1190 of 10000 ETA:  0:51:56
1200 of 10000 ETA:  0:51:53
1210 of 10000 ETA:  0:51:50
1220 of 10000 ETA:  0:51:46
1230 of 10000 ETA:  0:51:43
10000 of 10000 Time: 0:07:17
100% |################################################################################|
BruteforceSystemKeyBag : 0:07:17.113783
{'passcode': '1234', 'passcodeKey': '6b1470a37cc032309cd06f438064c3d5bbe321fa3ec5cfcf07667d7
True
Keybag type : System keybag (0)
Keybag version : 3
Keybag UUID : 0e0fcece75e840e2a72eb14f1c77cf02
-------------------------------------------------------------------------------
Class                                                WRAP Type       Key
-------------------------------------------------------------------------------
NSFileProtectionComplete                             3    AES        752555a88697c30c9d526fb
NSFileProtectionCompleteUnlessOpen                   3    Curve25519 932ce5db7b75c9eb032ed9c
568d6df68fa0773f
NSFileProtectionCompleteUntilFirstUserAuthentication 3    AES        4a979cb2c40d283b9e08208
NSFileProtectionRecovery?                            3    AES        591d067b952c78d31614ea4
```

PIN BRUTE FORCE TIMES

- Approximate Brute Force Times (Worst case):
  - 4 digits – 18 minutes
  - 8 digits – 125 days
  - 4 alphanumeric – 51 hours
  - 6 alphanumeric – 8 years
  - 8 alphanumeric – 10000 years

CONCLUSIONS

- How secure is your iPhone?
  - It Depends!!
- iOS has security built in and is continually improving
- Security depends on usage and the user (as is nearly always the case)

CONCLUSIONS

- If you are writing apps make sure you are protecting data in storage and transit

- If you are using apps be aware that others may not follow suit

- Remote wipe is effective if you can contact the device!

- If you care about the data on your device use more than a 4 digit PIN

- Some data may be retrievable anyway!

CREDITS AND MORE INFO

- Ken Van Wyk http://www.krvw.com/
- http://www.zdziarski.com/blog/wp-content/uploads/2013/05/iOS-Forensic-Investigative-Methods.pdf
- http://www.apple.com/ipad/business/docs/iOS_Security_Feb14.pdf
- http://esec-lab.sogeti.com/post/Low-level-iOS-forensics
- http://www.securitylearn.net/tag/iphone-data-recovery-on-ios-5/
- https://code.google.com/p/iphone-dataprotection/