OVERCOMING iOS DATA PROTECTION TO RE-ENABLE iPHONE FORENSICS

ANDREY BELENKO
ELCOMSOFT

black hat®
BRIEFINGS & TRAINING

USA + 2011

EMBEDDING SECURITY

AGENDA

- iPhone Forensics 101
- Pre-iOS 4 Forensics
- iOS 4 Data Protection
- iOS 4 Forensics

iOS FORENSICS 101

GOAL: provided physical access to the device extract as much information as practical

» iTunes Backups

- Amount of information varies by firmware
- Requires passcode or escrow file
- Backup can be encrypted by the device

» Filesystem/raw filesystem acquisition

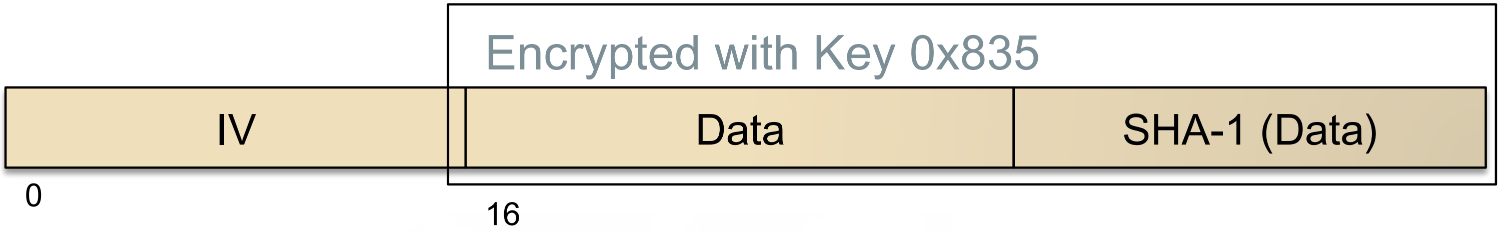
- Can get all information from the device
- Passcode and escrow file may be not necessary
- Requires exploit to boot unsigned ramdisk and kernel
- Acquired raw image can be encrypted

iOS 3 DISK ENCRYPTION

- No encryption before iPhone 3GS
- No data confidentiality protections
  - Encryption is to provide fast wipe, not to protect data
- Device automatically decrypts data
- Filesystem/raw filesystem acquisition is not affected

iOS 3 KEYCHAIN

- » All items are encrypted with the same key
  - Key 0x835 = AES_encrypt (uid-key, 0101..01)
- » Key is unique per device and is fixed for the lifetime of the device
- » Key 0x835 can be ‘extracted’ from the device for offline use
- » All past and future keychain items from the same device can be decrypted with the key



NEW IN iOS 4

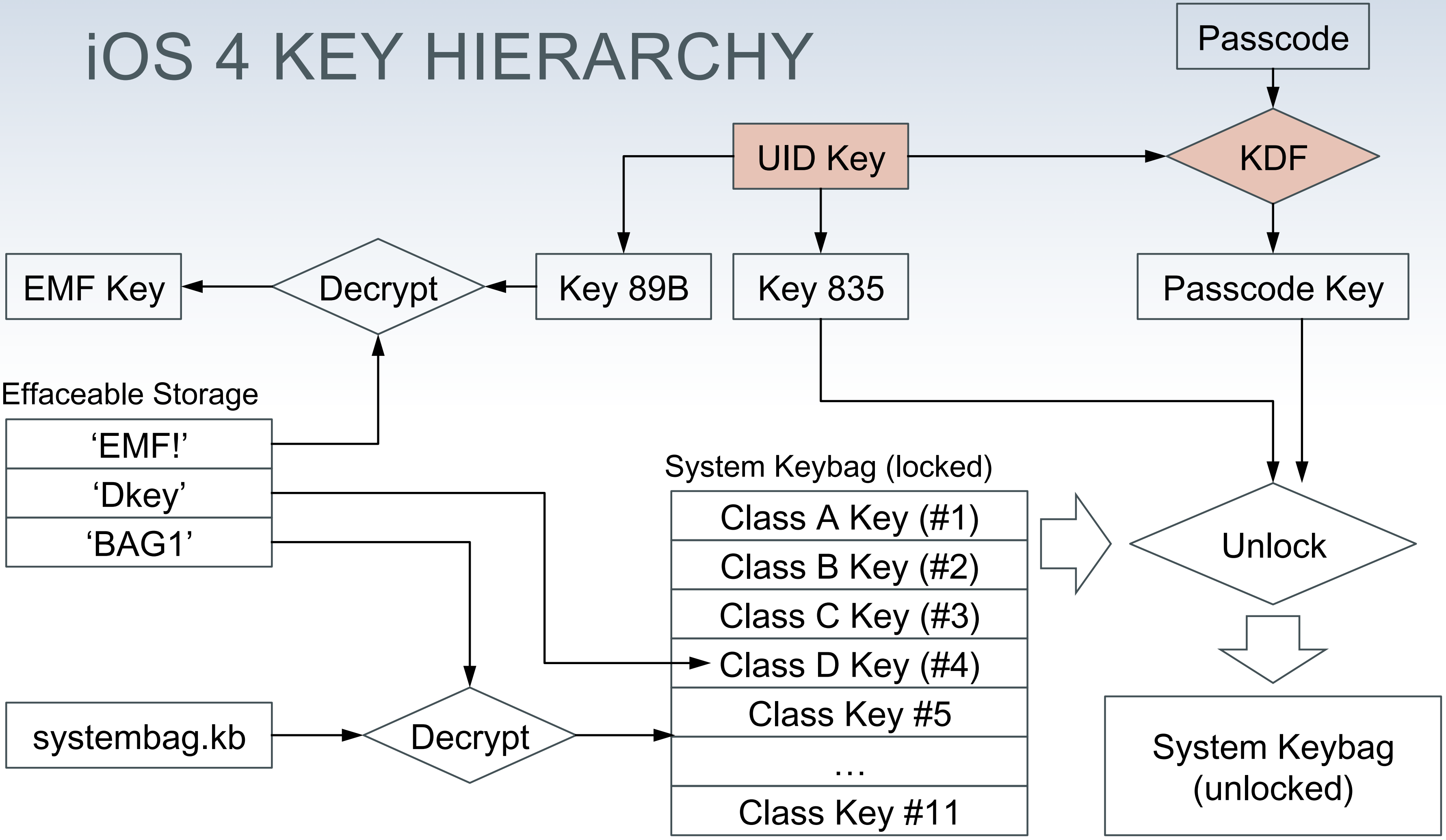
- Filesystem images are partially encrypted
  - Filesystem metadata is not encrypted — file names and properties are accessible
  - Contents of (almost all) files are encrypted
- New iTunes Backup format
  - Less of a problem — proprietary tools were available since day 0
- Keychain data is encrypted differently

All these are part of iOS 4 Data Protection

iOS 4 DATA PROTECTION

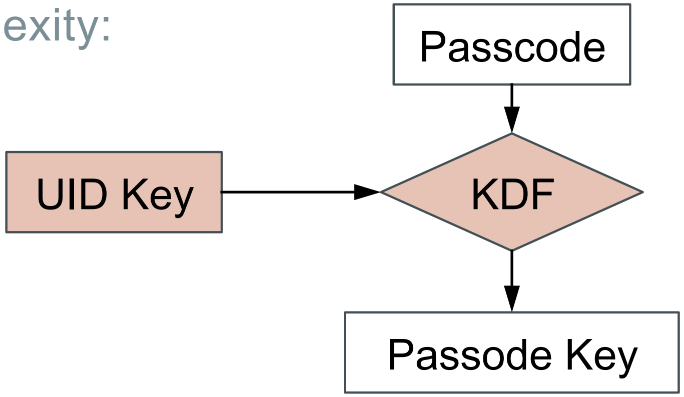
- » Content is grouped into protection classes based on availability requirements:
  - Available only when device is unlocked
  - Available after first device unlock
  - Always available
- » Separate protection classes for files and keychain items
- » Each protection class uses own master key
- » Class master keys are protected with device key and/or user passcode key
- » Encrypted protection class master keys are stored in system keybag
  - Keys are re-created during device restore

iOS 4 KEY HIERARCHY



iOS 4 PASSCODE

- Passcode key is required to unlock all but 3 keys in system keybag
  - Most files can be decrypted without it, most keychain items can't
- Passcode key computed from user passcode
  - Computation is tied to UID device key => must be computed on the device
- On-device bruteforce is slow
  - 2.1 p/s on iPhone 3G, 7 p/s on iPad
- System keybag contain hint on password complexity:
  - 0 = simple passcode, exactly 4 digits
  - 1 = digits-only passcode, length != 4
  - 2 = contains non-digits, any length

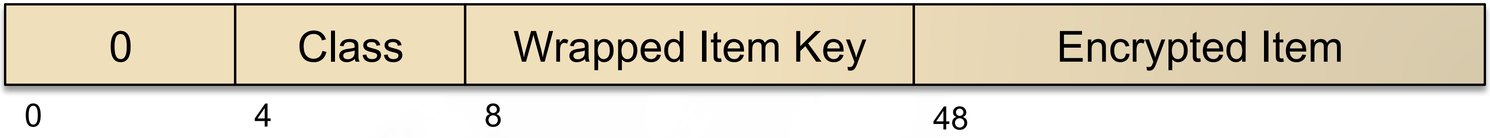


iOS 4 ESCROW KEYBAG

- Usability feature
  - Allows iTunes to unlock the device
- Contains same keys as system keybag
- Created when unlocked device is connected to the iTunes
- Stored on the computer
- Protected by 256-bit random “passcode”
  - Device stores “passcodes” for all paired computers
- Having escrow keybag gives same encryption keys as knowing the passcode

iOS 4 KEYCHAIN

- Available protection classes:
  - kSecAttrAccessibleWhenUnlocked
  - kSecAttrAccessibleAfterFirstUnlock
  - kSecAttrAccessibleAlways
  - …ThisDeviceOnly — do not include in the backup
- Random key for each item
  - Key wrapped with protection class master key is stored with the item



0	Class	Wrapped Item Key	Encrypted Item
0	4	8	48

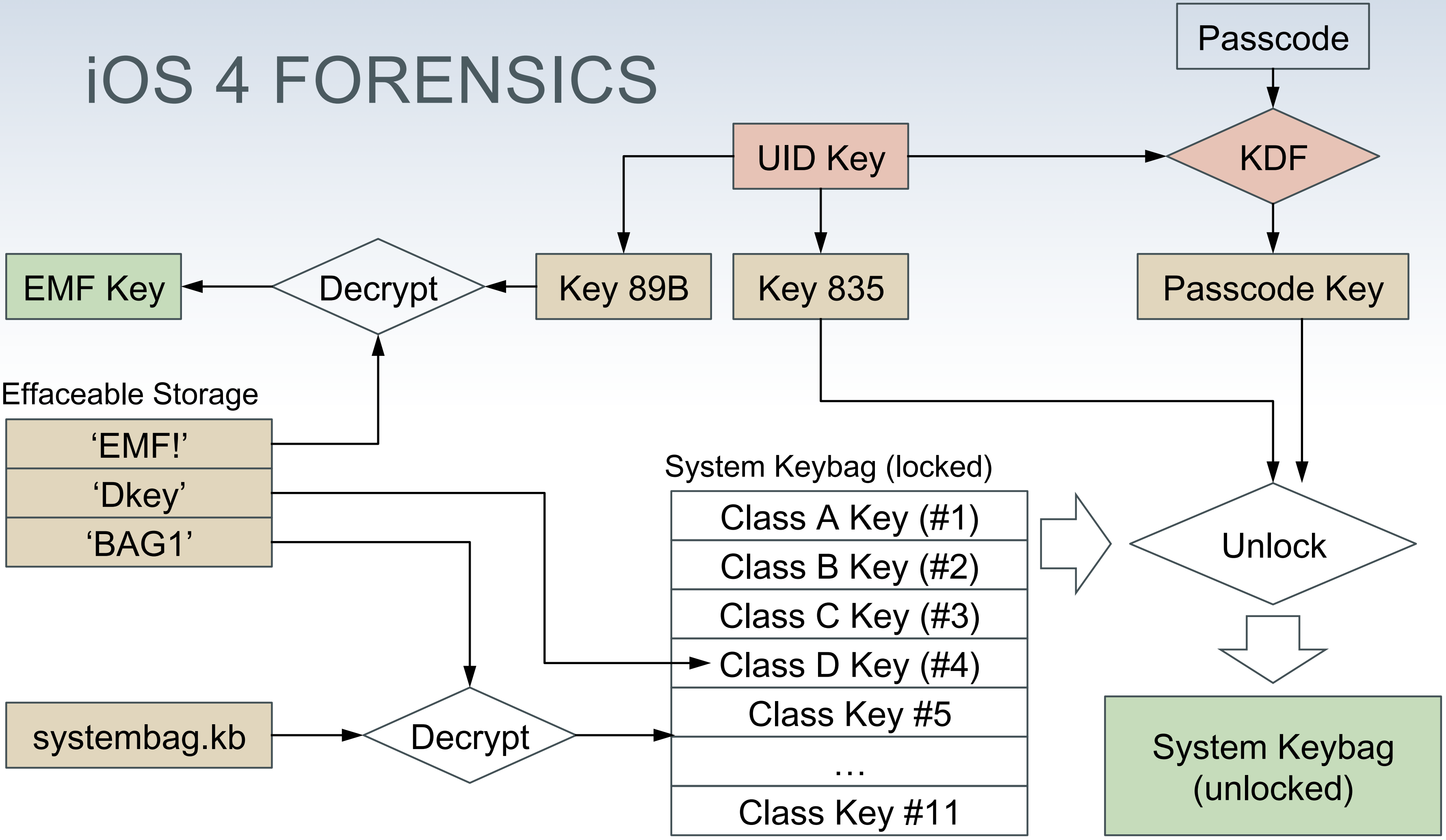
iOS 4 DISK ENCRYPTION

- Available protection classes:
  - NSProtectionNone
  - NSProtectionComplete
- Filesystem metadata is encrypted with EMF key (similar to the iOS 3)
  - Transparently decrypted by the device
- File contents are encrypted with per-file random key instead of EMF key
  - Key wrapped with protection class master key is stored in files' extended attribute `com.apple.system.cprotect`
- During **`dd`**-style imaging iOS decrypts file data using EMF key => garbage
  - To recover file data: encrypt with EMF key, then decrypt with file key

iOS 4 FORENSICS

- Acquiring disk image is not enough for iOS 4
  - Content protection keys must also be extracted from the device
  - EMF key is also needed to decrypt `dd` images
- Passcode or escrow keybag is needed for a complete set of keys
- In real world it might be better to extract source data and compute protection keys offline

iOS 4 FORENSICS



Passcode

UID Key

KDF

EMF Key

Decrypt

Key 89B

Key 835

Passcode Key

Effaceable Storage

‘EMF!’

‘Dkey’

‘BAG1’

System Keybag (locked)

Class A Key (#1)

Class B Key (#2)

Class C Key (#3)

Class D Key (#4)

Class Key #5

…

Class Key #11

Unlock

systembag.kb

Decrypt

System Keybag (unlocked)

SUMMARY

- iPhone physical analysis is possible again
- Physical acquisition requires bootrom/iBoot exploit
- Passcode is *usually* not a problem
- Proprietary and open-source tools for iOS 4 forensics available

THANK YOU

QUESTIONS?

blackhat®

BRIEFINGS & TRAINING

USA + 2011

EMBEDDING SECURITY

PLEASE TURN IN YOUR COMPLETED FEEDBACK FORM AT THE REGISTRATION DESK

blackhat®

BRIEFINGS & TRAINING

USA + 2011

EMBEDDING SECURITY