SSTIC 2012

Jean-Baptiste Bédrune
Jean Sigwald

Forensics iOS

SOGETI

Introduction

- Analyses forensiques de terminaux iOS Apple
- Travaux précédents
  - iPhone data protection in depth, HITB, mai 2011
  - Outils open-source
- Document Apple « iOS Security », mai 2012
- Acquisition et (dé)chiffrement

Forensics iOS - Plan

- Techniques d'acquisition
  - Acquisition physique
- Mécanismes de chiffrement d'iOS
  - Clé CPU et clés dérivées
  - iOS 3
  - iOS 4
  - iOS 5
  - Backup iTunes
- Démo

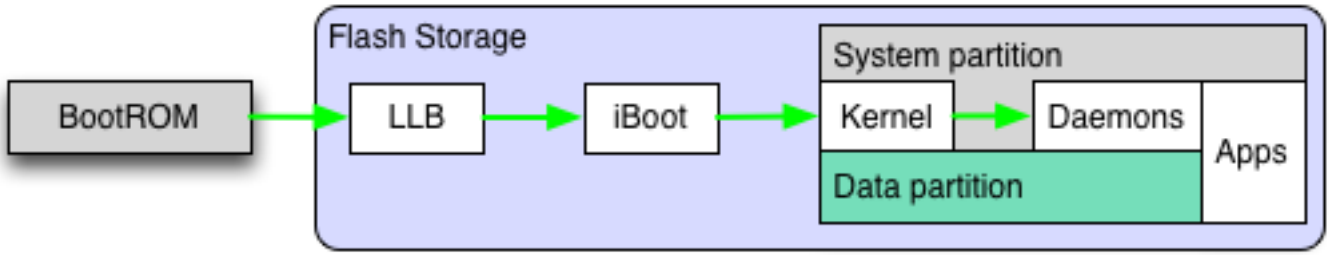
Techniques d'acquisition

- Extraction logique : backup iTunes
  - Code PIN téléphone requis (appairage)
  - Mot de passe du backup (si utilisé)
- Extraction physique : image dd des partitions
  - 2 partitions : système/données (HFS+)
  - Accès root nécessaire
  - Code PIN téléphone pour déchiffrer certains fichiers (iOS 4+)

Acquisition physique – accès root

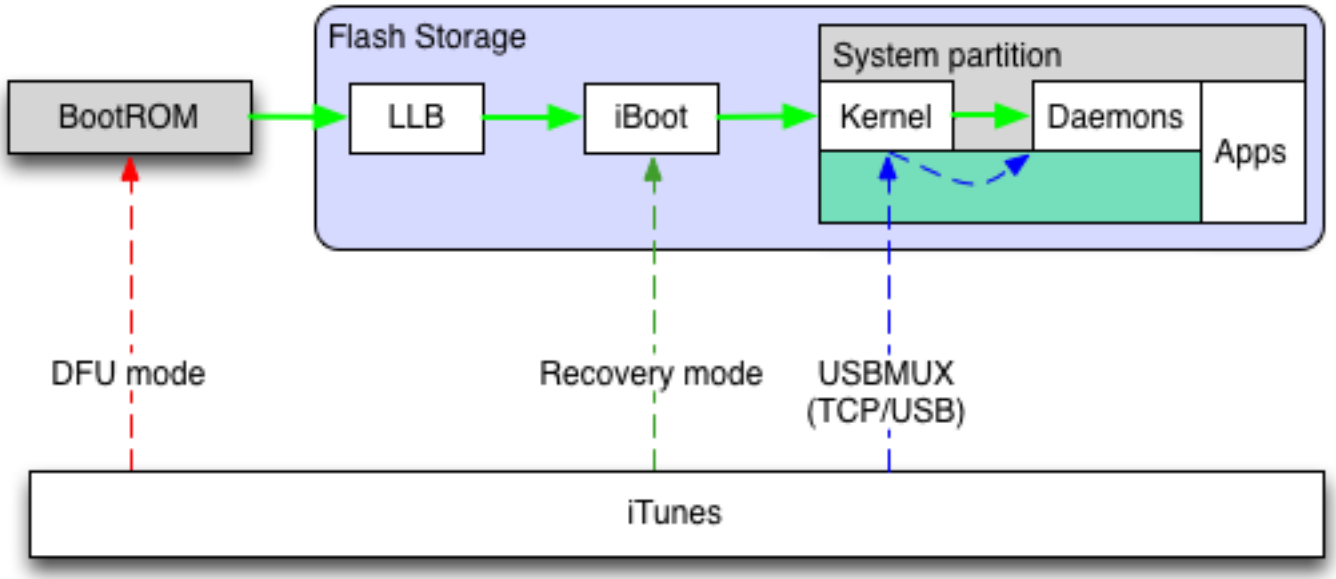
- Jailbreak
  - Code PIN téléphone souvent requis
    - interaction avec la GUI ou services iTunes pour exploiter des vulnérabilités
  - Modifie les partitions, laisse des traces
- Exploits bootloaders issus des outils de jailbreak
  - Exécution de code avant le démarrage du système
  - Environnement d'acquisition en mémoire (ramdisk)

Boot sécurisé iOS



- Bootloaders et binaires chiffrés/signés
  - BootROM en lecture seule
- Plateforme contrôlée par Apple

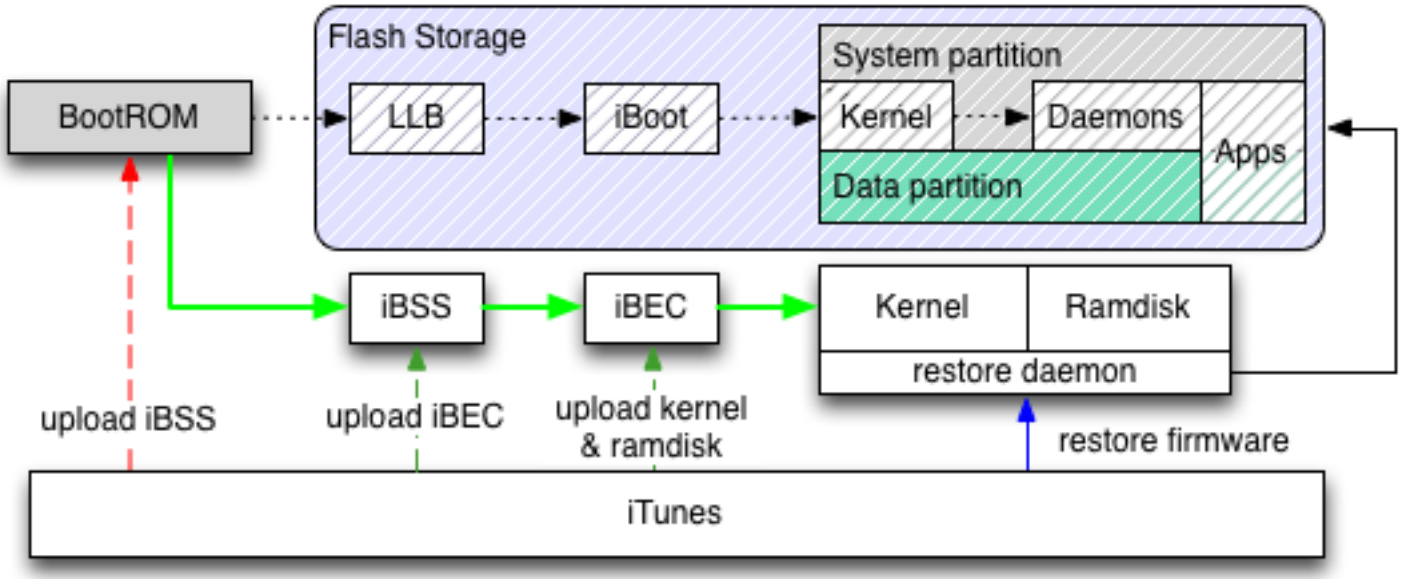
Interfaces USB



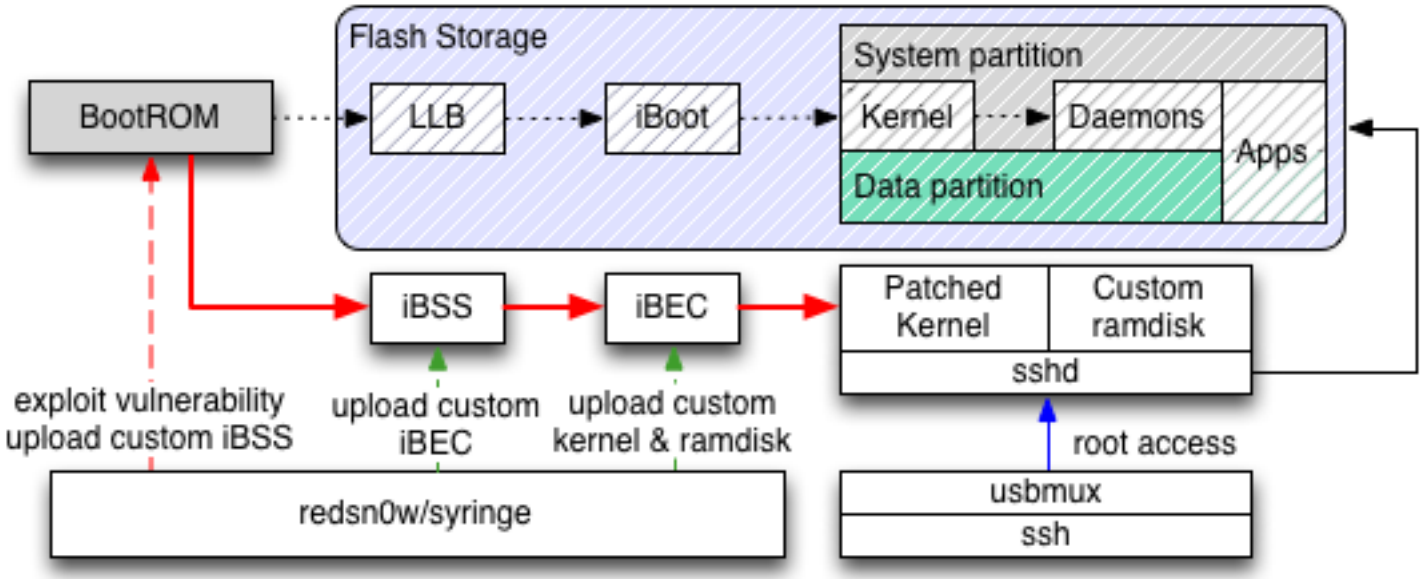
Exploits BootROM

- Exécution de code via le mode DFU (Device Firmware Upgrade)
  - Le BootROM est en lecture seule et ne peut pas être mis à jour
  - Fonctionne quelque soit la version d'iOS installée
  - Permet de charger noyau/ramdisk non signés
- Exploits publics pour tous les modèles jusqu'à l'iPhone 4
  - Pwnage2 (iphone dev team) : iPhone 2G, iPhone 3G
  - Steaks4uce (pod2g) : iPod Touch 2G
  - Limera1n (geohot) : iPhone 3GS, iPad 1, iPhone 4
- Pas de vulnérabilité BootROM connue à ce jour pour les iPad2/iPhone 4S/iPad 3 (CPU A5)
  - Acquisition physique possible si un jailbreak existe pour la version d'iOS installée

Flashage de firmware en mode DFU



Ramdisk SSH/forensics



Acquisition physique

Modèles	Jailbreak nécessaire	Code PIN requis
iPad 1 ≤ iPhone 4	Non : exploits BootROM	Non
iPad 2/3 iPhone 4S	Oui	Oui (en général)

- A partir d'iOS 4
  - Montage de la partition de données depuis un ramdisk : certains fichiers ne peuvent pas être lus
  - Tous les fichiers sont illisibles via un dump dd

Mécanismes de chiffrement d'iOS

- Clé UID et clés dérivées
- iOS 3
  - Chiffrement de la Flash NAND
  - Keychain
- iOS 4
  - Data protection
  - Keybag système
  - Zone effaçable
  - Bruteforce du code PIN
  - Escrow keybag
- iOS 5
- Backup iTunes

Clé UID et clés dérivées

- Clé UID: clé AES unique enfouie dans le CPU
  - Utilisable par les bootloaders et le noyau
  - Ne peut pas être extraite sans attaque matérielle
- Utilisée pour dériver des clés uniques au démarrage
  - Chiffrement de blocs fixes de 128 bits avec la clé UID
    - Résultat unique pour chaque appareil
- Clé 0x835 = AES_UID(01010101010101010101010101010101)
- Clé 0x89B = AES_UID(183E99676BB03C546FA468F51C0CBD49)

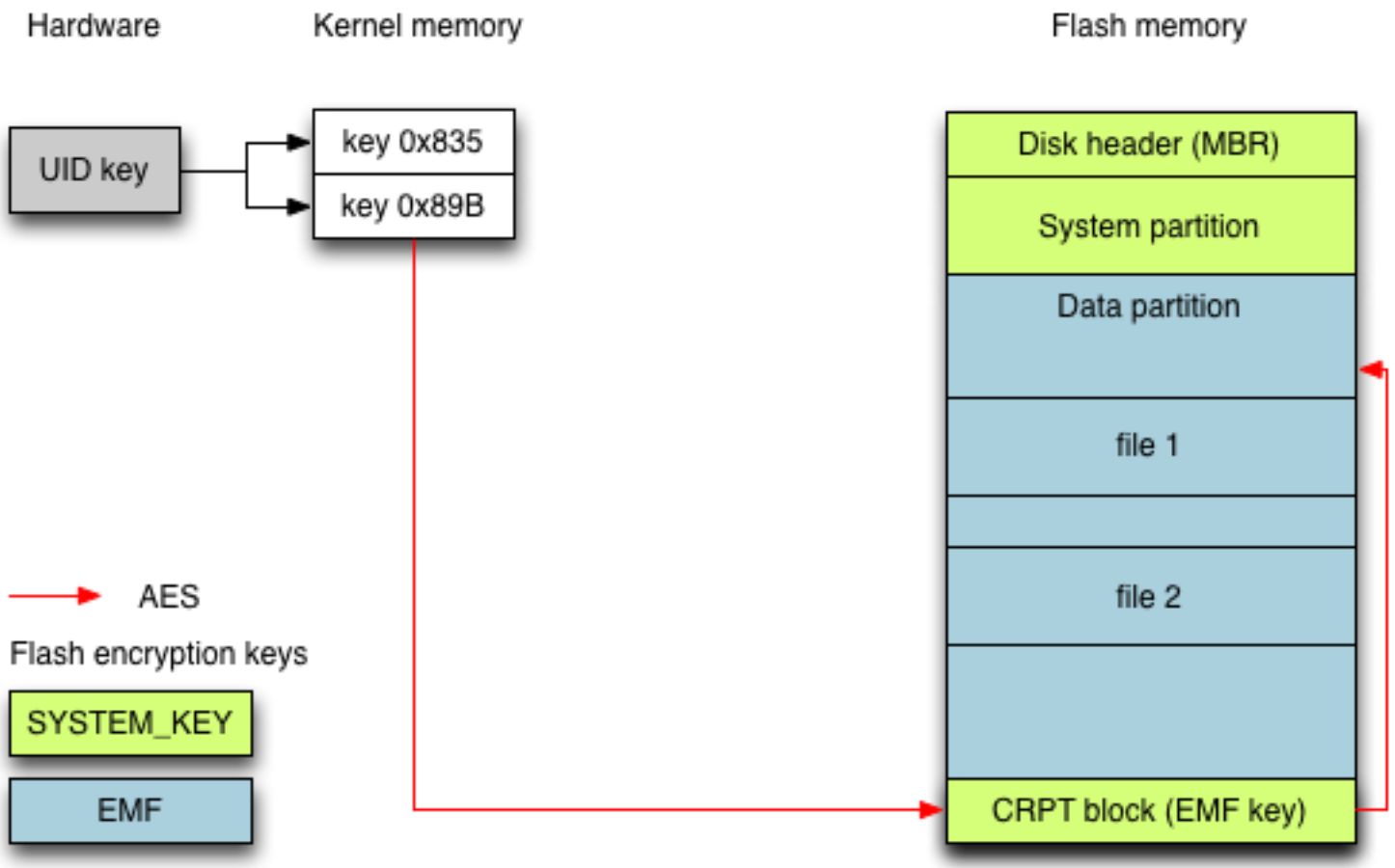
Clé UID et clés dérivées

- *« A 256-bit AES key that's burned into each processor at manufacture. It cannot be read by firmware or software, and is used only by the processor's hardware AES engine. To obtain the actual key, an attacker would have to mount a highly sophisticated and expensive physical attack against the processor's silicon. »*
- *« The UID is unique to each device and is not recorded by Apple or any of its suppliers »*
  - Source : iOS Security, Apple, mai 2012

Chiffrement iOS 3

- iPhone 3GS, juin 2009
- Chiffrement de la Flash NAND
  - AES-CBC via le contrôleur DMA (CDMA)
- Partition système : clé fixe (AES128)
  - Définie en dur dans les bootloaders et le noyau
- Partition de données : clé EMF (AES256)
  - Clé de volume, générée aléatoirement lors du formatage
  - Effacée lors du wipe
  - Stockée dans le dernier *bloc logique* de la partition
    - Bloc chiffré par la clé « système »
    - Contenu du bloc chiffré par la clé 0x89B

Chiffrement iOS 3

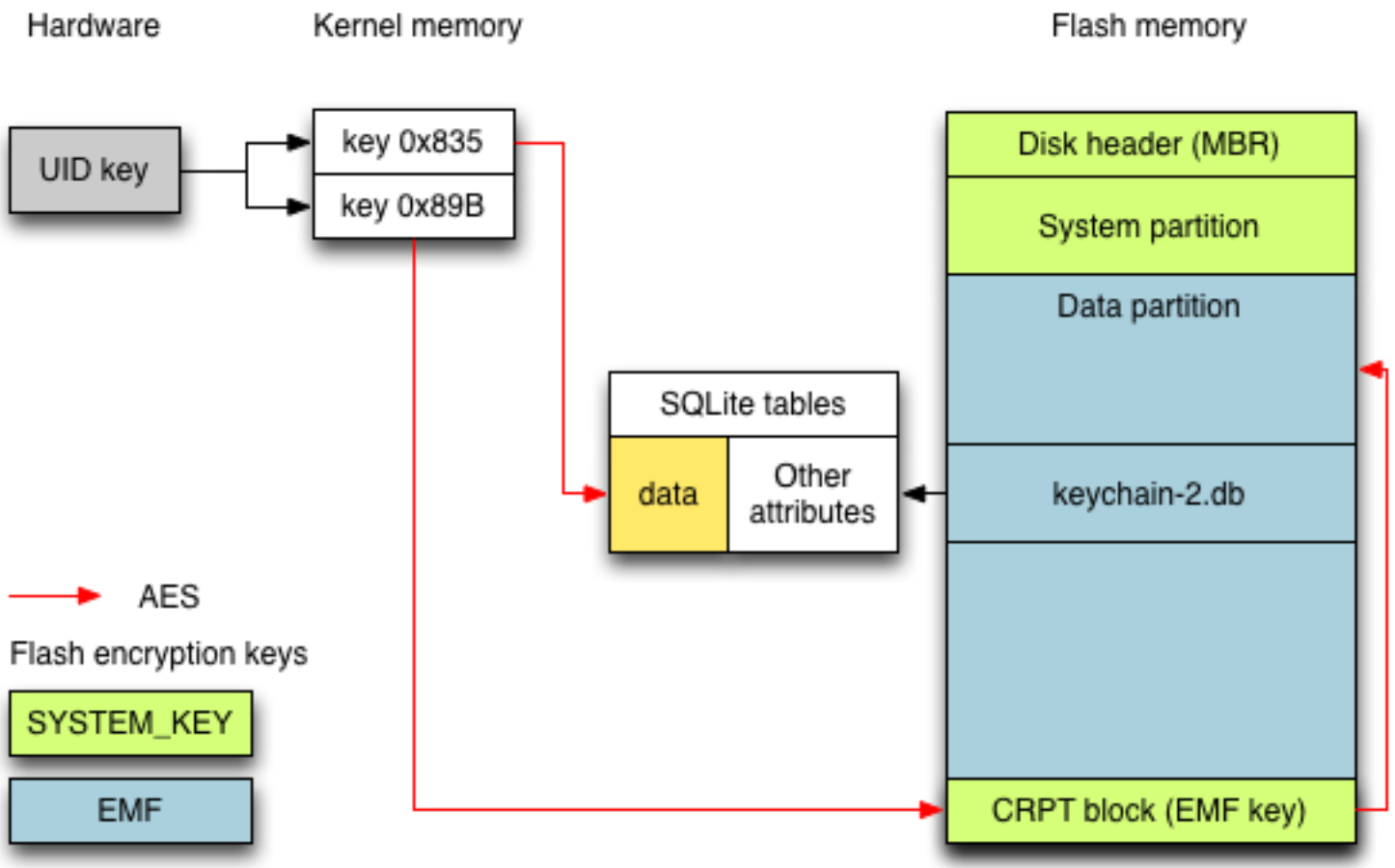


Chiffrement iOS 3 - Keychain

- Base de données SQLite
  - Mots de passe, certificats, clés privées
- Secrets chiffrés avec la clé 0x835 (AES128-CBC)
  - Mécanisme présent depuis iOS 1.x

```
$ sqlite3 /var/Keychains/keychain-2.db
sqlite> select acct, svce, hex(data), agrp from genp;
    linksys            |AirPort    |F6A82EDFC424050B.....|apple
    BackupPassword     |BackupAgent|F6B9813BDD97BB86.....|apple
    DeviceLockPassword|SpringBoard|F37A7F64BF2B91C0.....|apple
sqlite> select acct,srvr,port,hex(data),agrp from inet;
    test@gmail.com|imap.gmail.com|143|D1369EFF14830.....|apple
```

Chiffrement iOS 3 - Keychain



iOS 4

- iPhone 4, juin 2010
  - Fonctionnalités « data protection »
- Classes de protection
  - Accessibilité des données selon l'état de verrouillage du terminal
  - Applicable aux fichiers et secrets du Keychain
    - APIs disponibles pour les développeurs
  - 1 classe = 1 clé maitre (AES256)
  - Clés maîtres stockées dans un fichier spécial : keybag système
- Utilisation du code PIN téléphone pour protéger certaines clés maîtres
- Zone effaçable

Classes de protection

- Fichiers
  - NSProtectionNone
    - Classe par défaut, fichiers toujours accessibles
  - NSProtectionComplete
    - Fichiers accessibles uniquement lorsque le terminal est déverrouillé
- Keychain
  - kSecAttrAccessibleAlways(thisDeviceOnly)
  - kSecAttrAccessibleAfterFirstUnlock(thisDeviceOnly)
  - kSecAttrAccessibleWhenUnlocked(thisDeviceOnly)

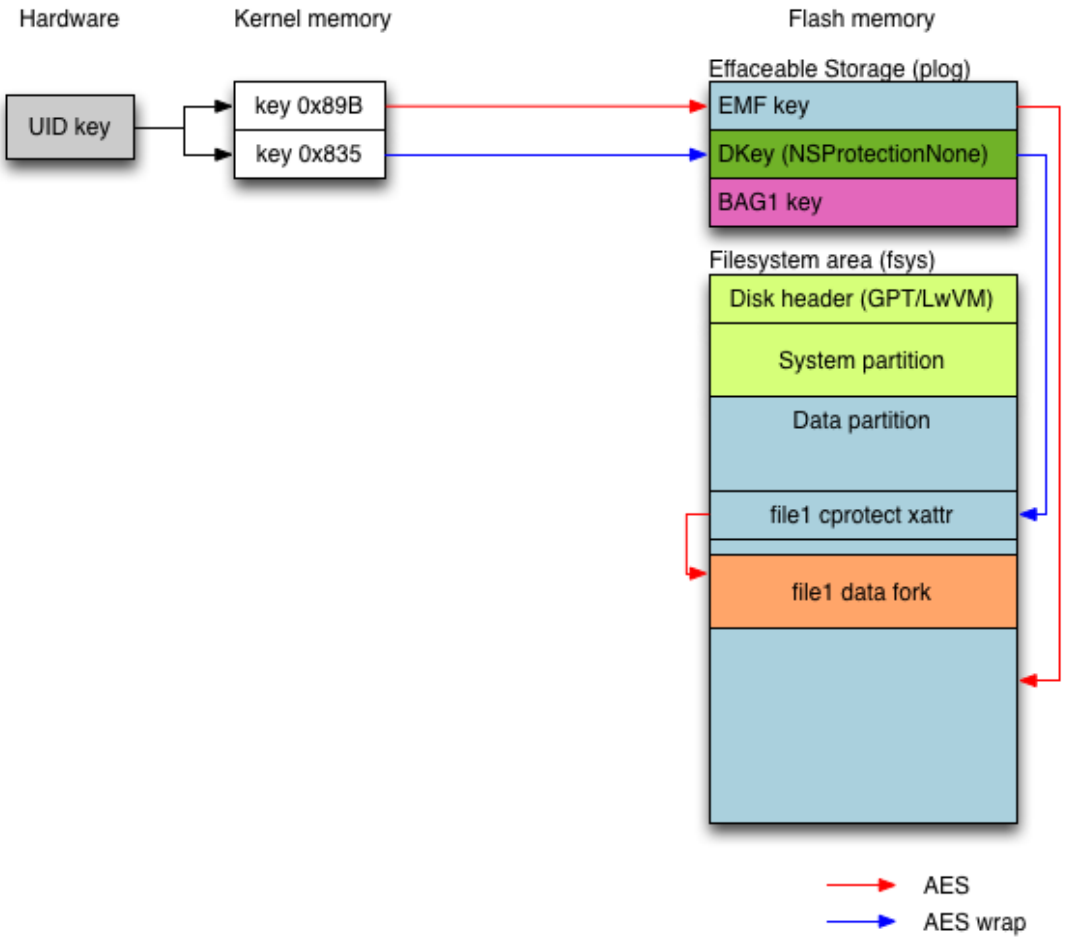
Classes de protection - fichiers

- La clé de volume (EMF) chiffre la partition de données
  - Sauf le contenu des fichiers
- Chaque fichier possède une clé unique (file key)
  - Chiffre le « data fork » (contenu du fichier)
  - 1 seule couche de chiffrement
- File key protégée par la clé maitre de la classe choisie
  - AES-wrap (RFC 3394)
  - Stockée dans un attribut étendu du fichier
    - com.apple.system.cprotect

Zone effaçable

- Bloc de mémoire Flash réservé
  - Non soumis au FTL
- Contient les 3 clés permettant le montage de la partition de données
  - EMF : clé du volume
  - DKey : clé maitre NSProtectionNone
  - BAG1 : clé du keybag système
- Bloc effacé lors du wipe

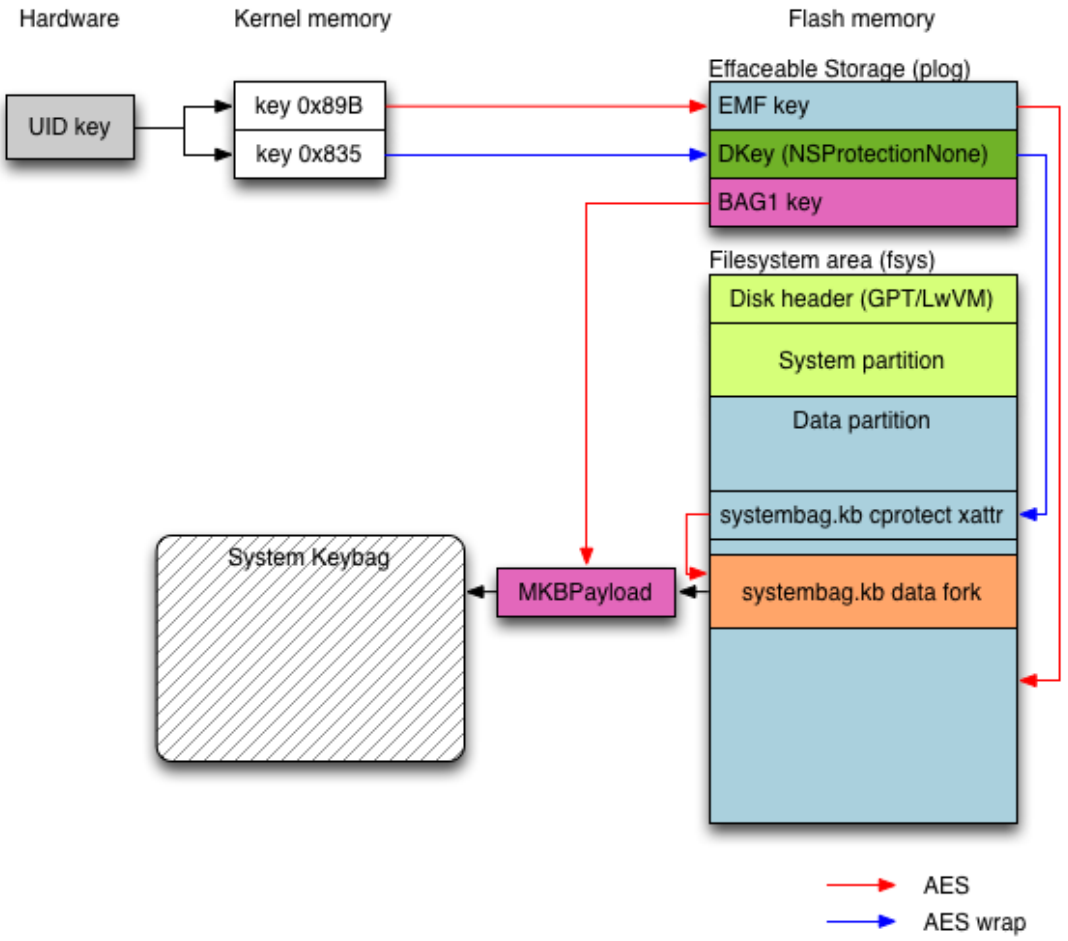
iOS 4 - NSProtectionNone



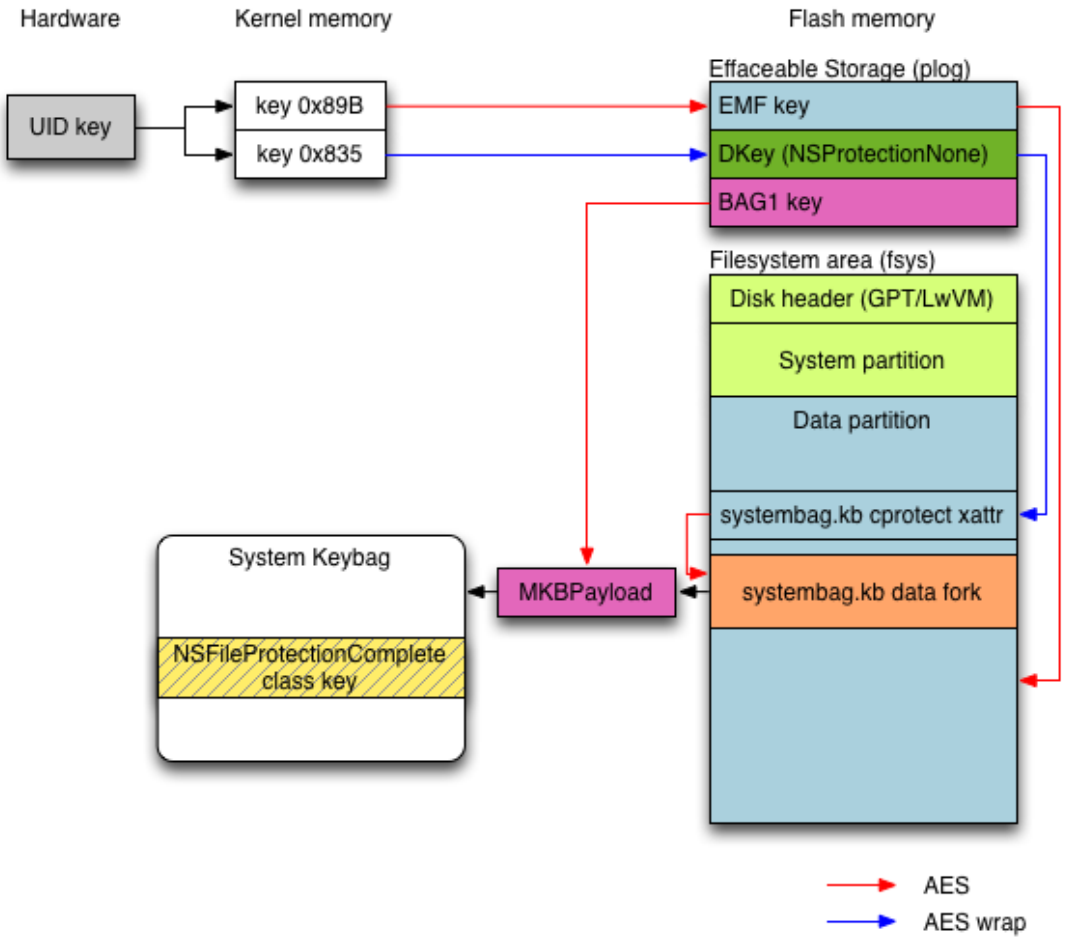
Keybag système

- Contient les clés maîtres des autres classes de protection
  - Fichier /var/keybags/systembag.kb
  - NSProtectionNone + surchiffrement
- Clés maîtres chiffrées
  - Par la clé 0x835 (AES) pour toutes les classes
  - Par la passcode key (AES-wrap) pour les classes liées au code PIN téléphone
- Autres informations
  - Sel et nombre d'itérations pour la dérivation du code PIN en passcode key

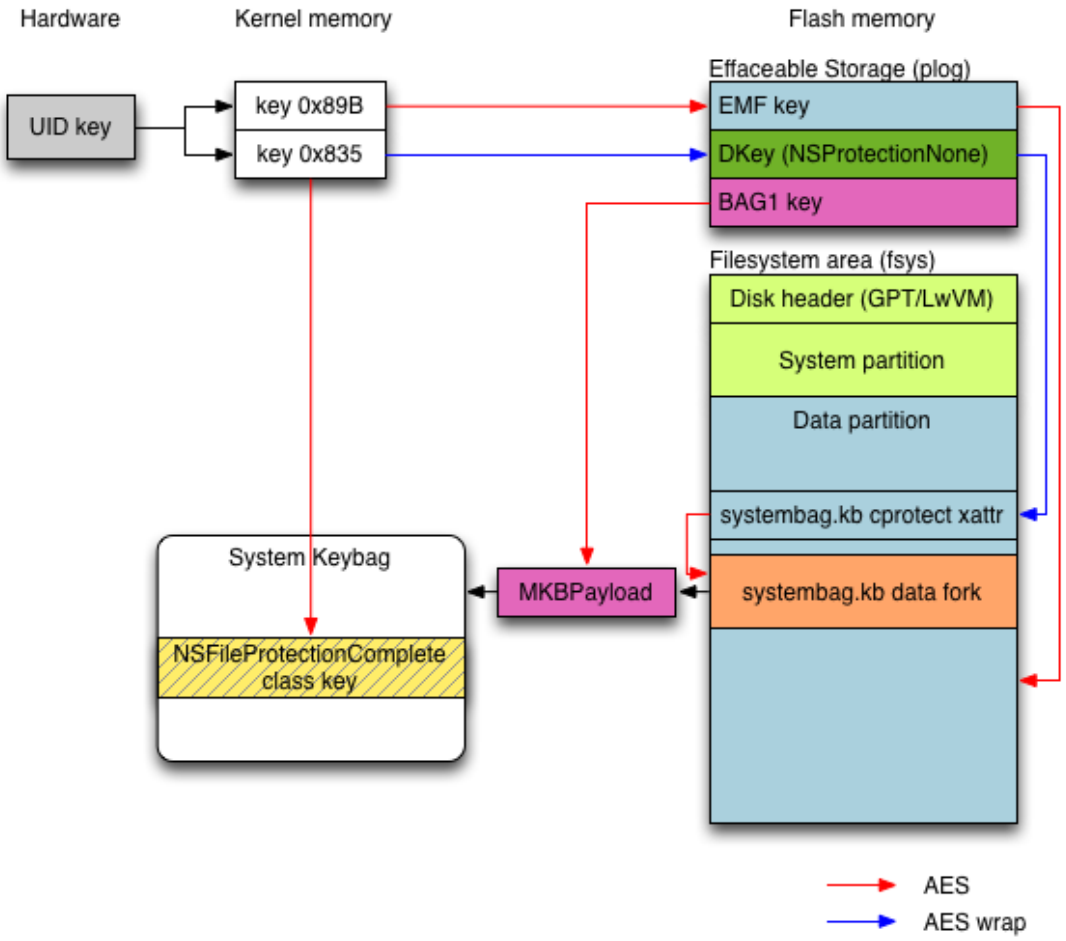
Chargement du keybag système



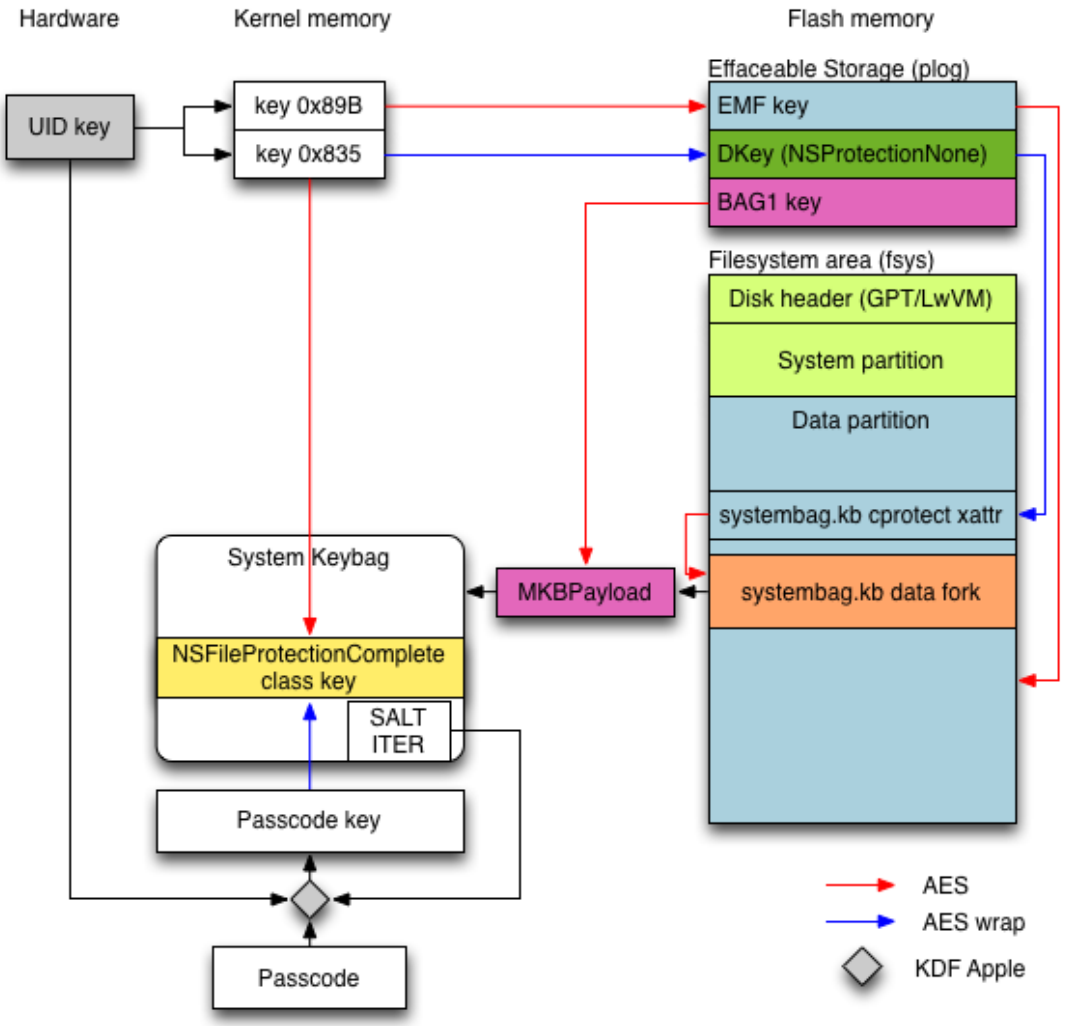
Chargement du keybag système



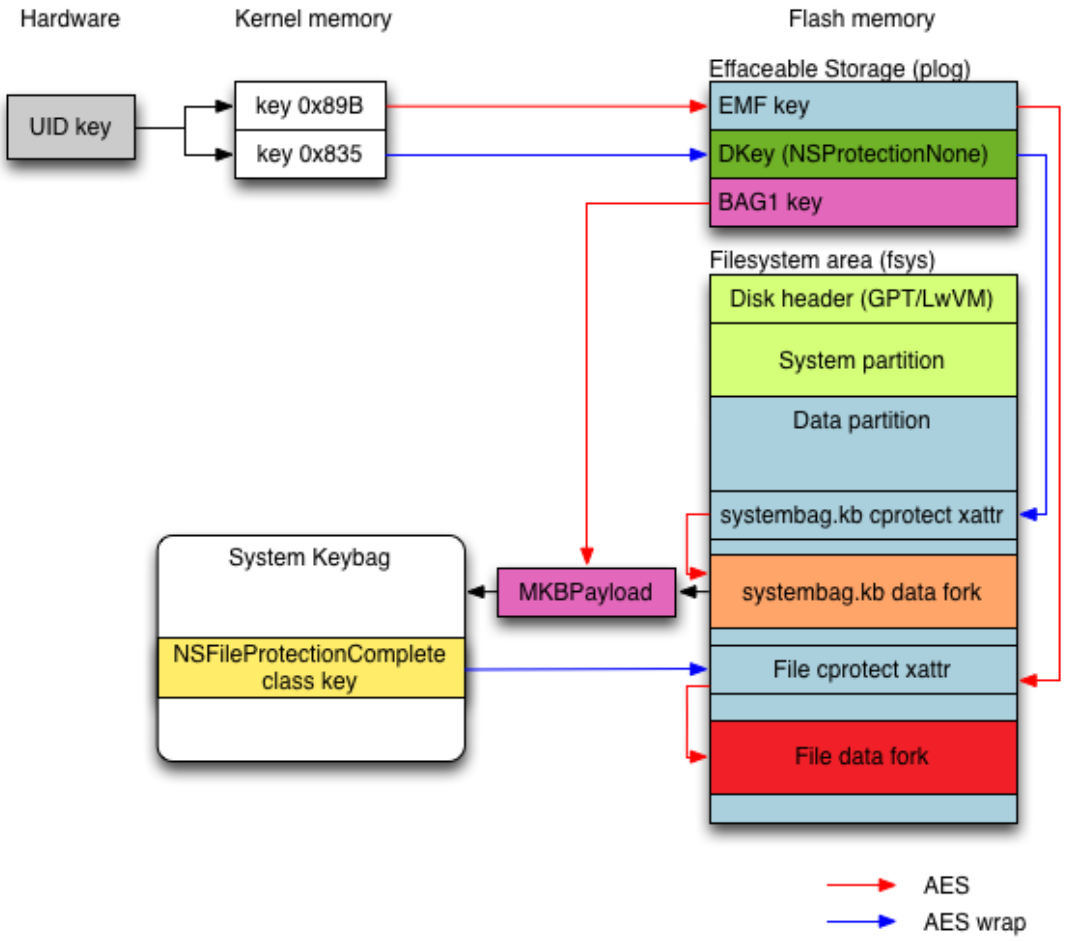
Déverrouillage du keybag système



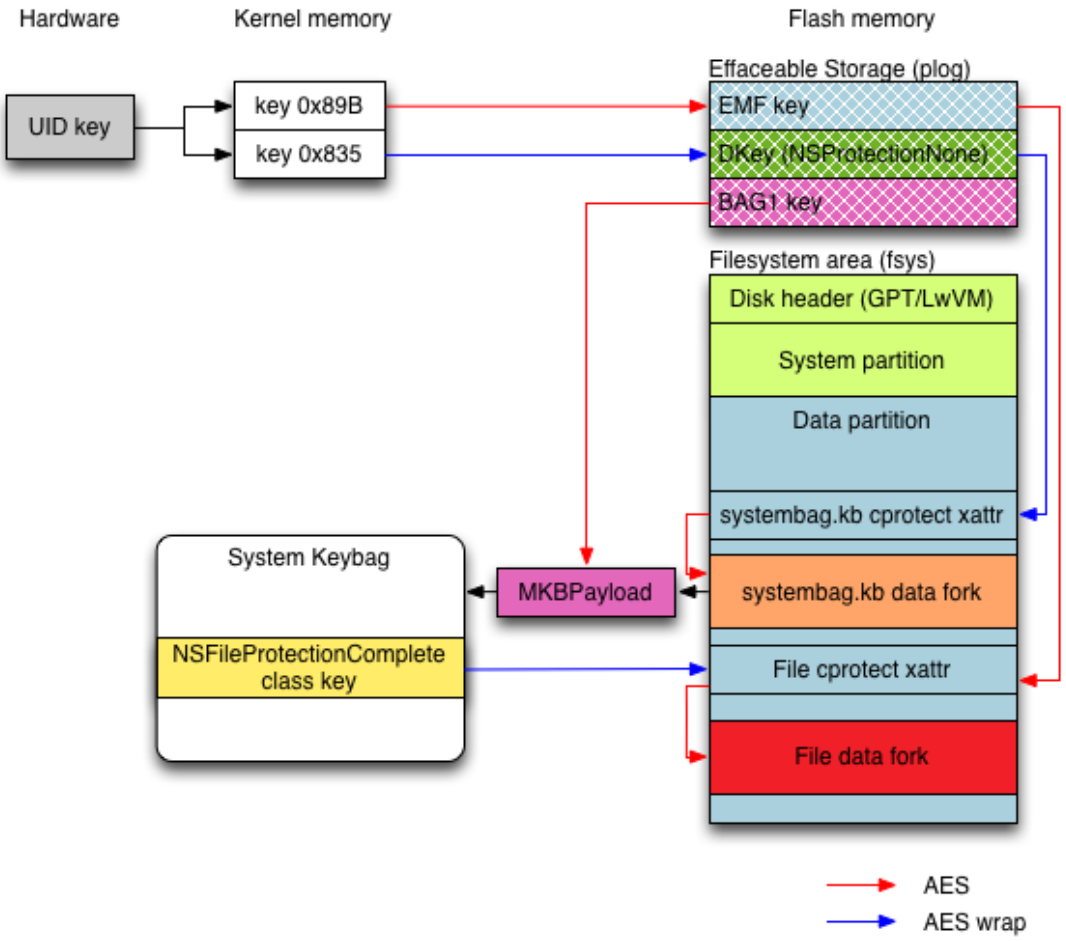
Déverrouillage du keybag système



Accès à un fichier protégé



Wipe


Hardware
Kernel memory
Flash memory
UID key
key 0x89B
key 0x835
Effaceable Storage (plog)
EMF key
DKey (NSProtectionNone)
BAG1 key
Filesystem area (fsys)
Disk header (GPT/LwVM)
System partition
Data partition
systembag.kb cprotect xattr
systembag.kb data fork
File cprotect xattr
File data fork
System Keybag
NSFileProtectionComplete class key
MKBPayload
AES
AES wrap

Fonction de dérivation

- Transforme le code PIN téléphone en passcode key
- Algorithme propriétaire Apple (« tangling »)
  - 1 itération de PBKDF2
  - 390 itérations d'AES avec la clé UID
    - La clé dérivée est liée au terminal d'origine
- Limite la vitesse d'une attaque par bruteforce
  - Environ 10 mdp/s

Complexité du code PIN

- 4 digits

Complexité du code PIN

- 4 digits
- N digits

Complexité du code PIN

- 4 digits
- N digits
- N caractères

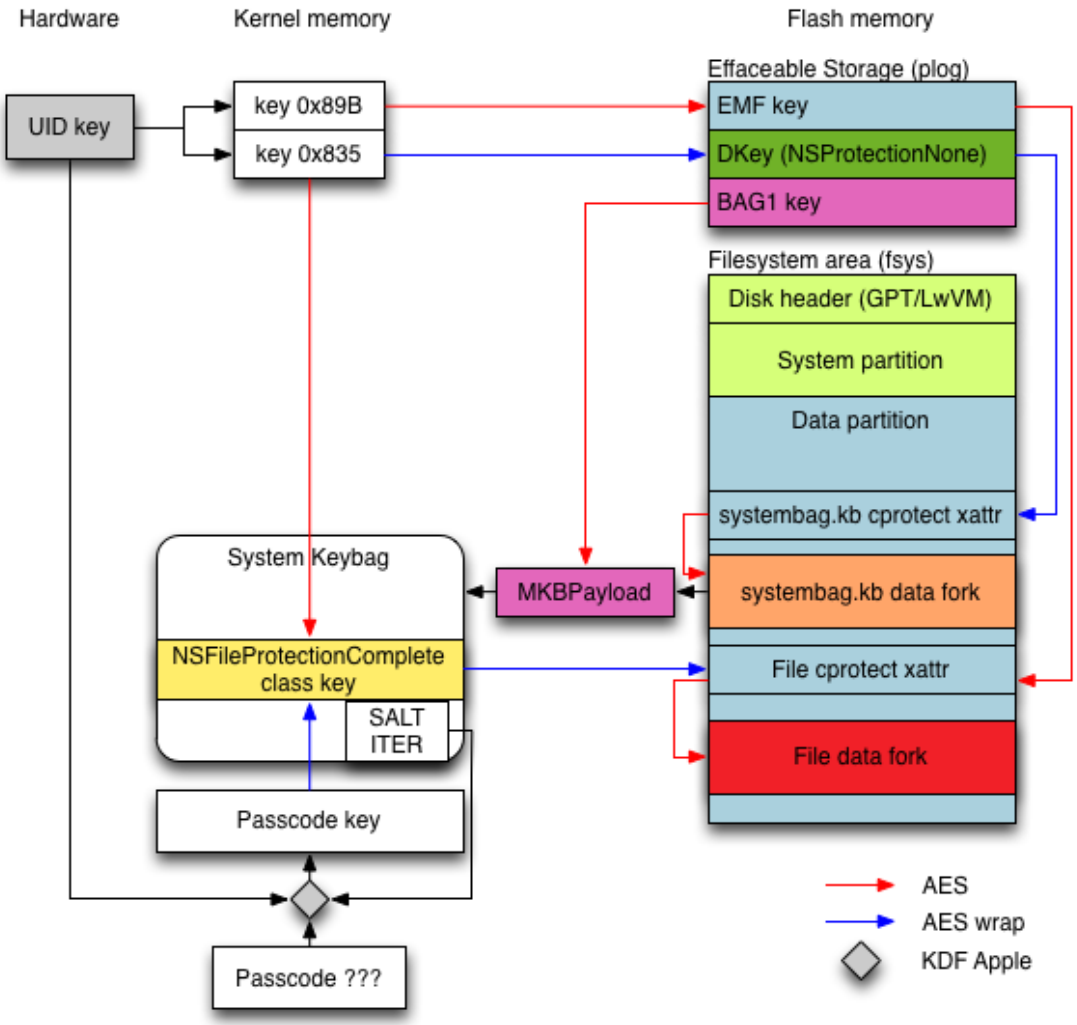
Bruteforce du code PIN

- Code PIN requis pour les classes de protection :
  - NSProtectionComplete
  - kSecAttrAccessibleAfterFirstUnlock(thisDeviceOnly)
  - kSecAttrAccessibleWhenUnlocked(thisDeviceOnly)
- Bruteforce possible via un accès root
  - Dérivation avec la clé UID
  - AES unwrap des clés et test d'intégrité
  - Le service noyau AppleKeyStore peut être utilisé directement

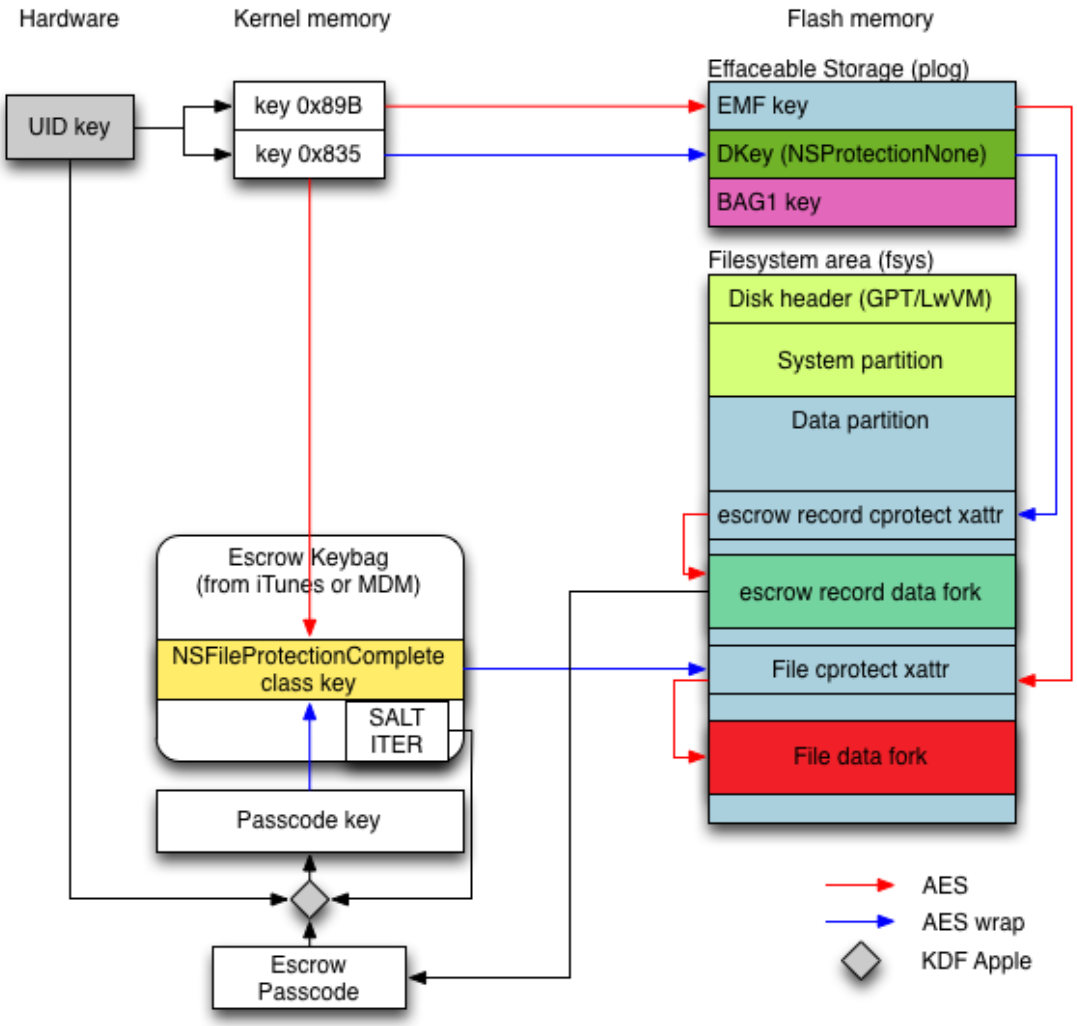
Escrow keybag

- Export du keybag système
  - iTunes : synchronisation d'un terminal verrouillé (mais appairé)
  - Serveur MDM : remise à zero du code PIN oublié
- Protégé par une passphrase aléatoire
  - Stockée sur le terminal
    - /var/root/Library/Lockdown/escrow_records/
  - NSProtectionNone (iOS 4)
- Si un attaquant a accès au terminal et à un escrow keybag alors il n'a pas besoin de bruteforcer le code PIN téléphone
  - Compromis sécurité/fonctionnalités

Escrow keybag



Escrow keybag



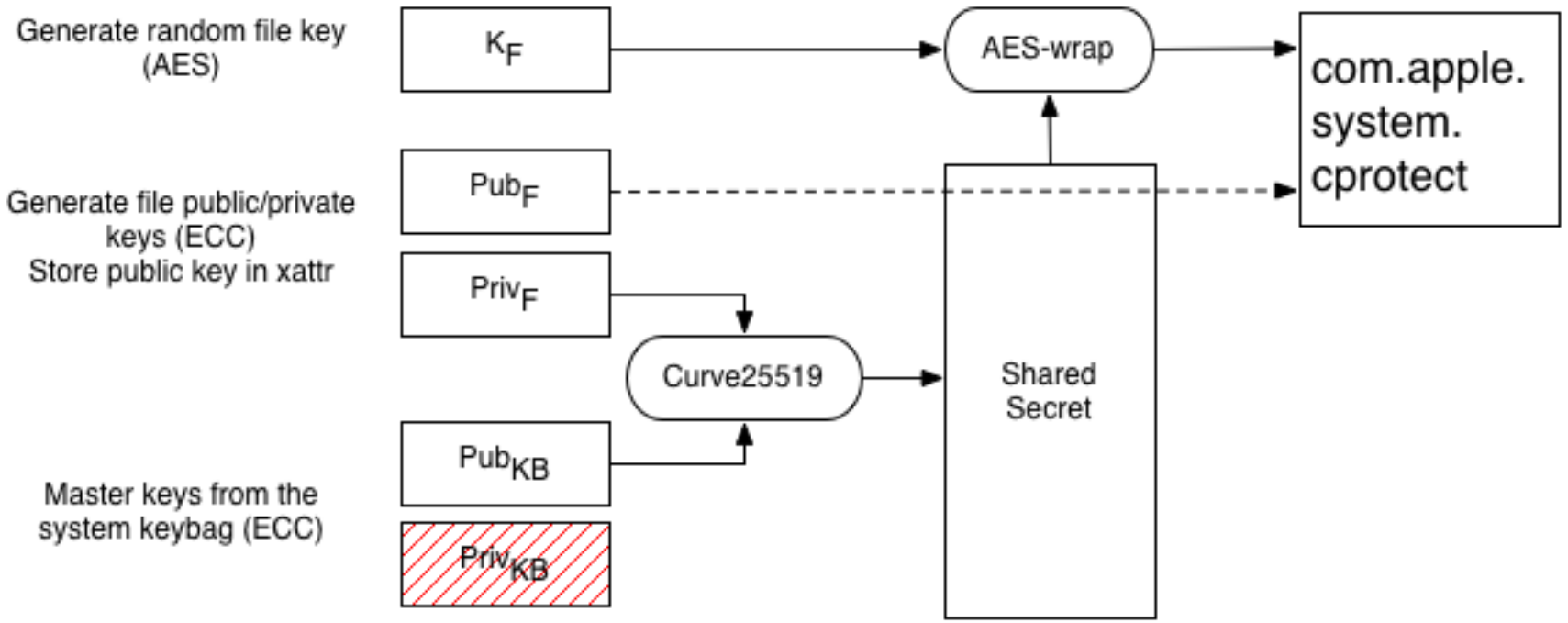
iOS 5

- iPhone 4S, octobre 2011
- Modification du chiffrement du Keychain
  - Tous les attributs sont chiffrés (login, @email, etc.)
  - AES256 en mode GCM
- Nouvelles classes de protection pour les fichiers
  - NSFileProtectionCompleteUntilFirstUserAuthentication
    - Liée au code PIN
    - Utilisée pour protéger les escrow records
      - Corrige la vulnérabilité des escrow keybags
  - NSFileProtectionCompleteUnlessOpen

ProtectionCompleteUnlessOpen

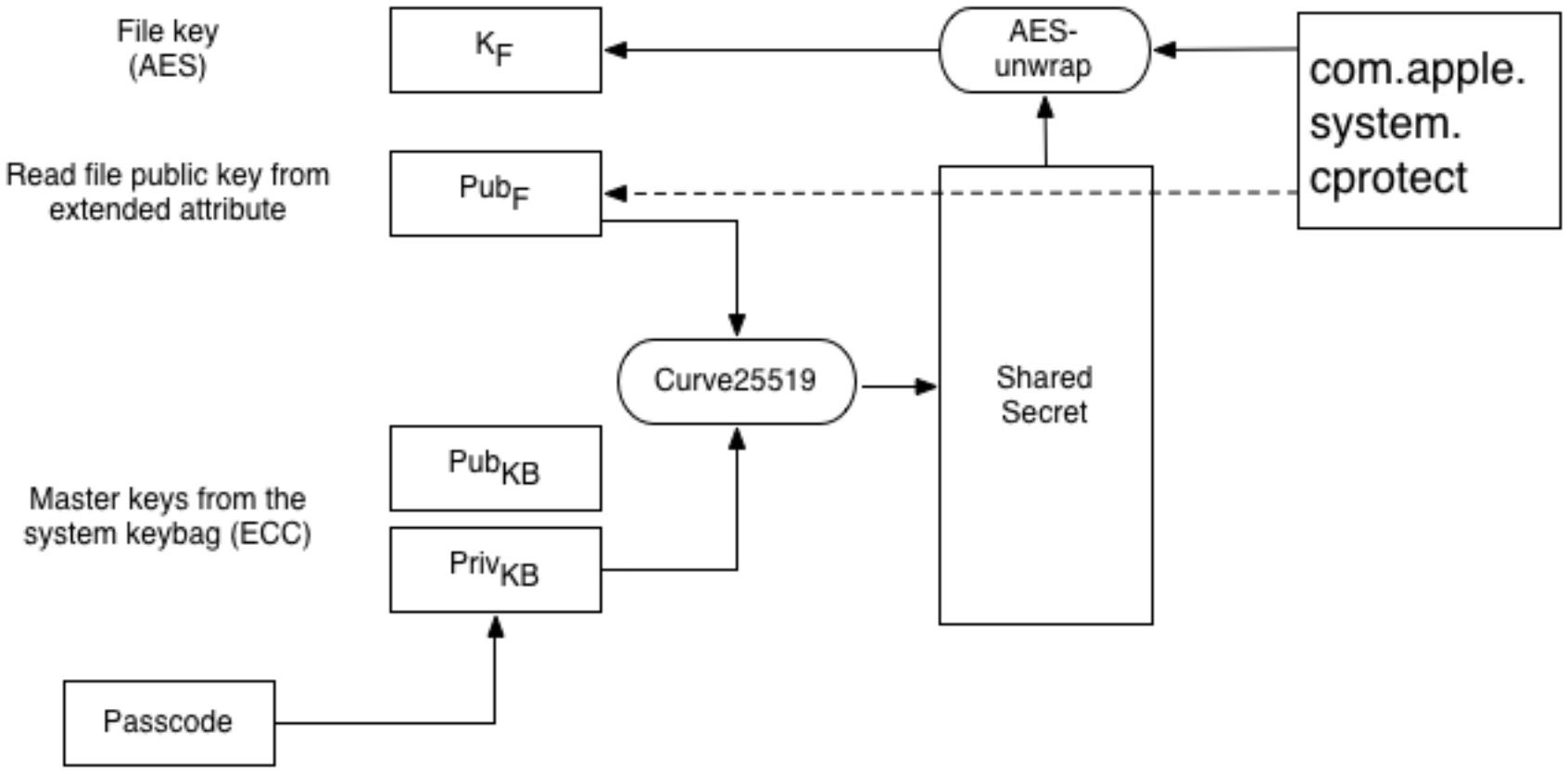
- Permet de créer des fichiers protégés par le code PIN même si le terminal est verrouillé
  - Exemple : e-mails et pièces jointes
- Cryptographie asymétrique
  - Diffie-Hellman sur courbes elliptiques
    - Curve25519 (D. J. Bernstein)
  - Secret partagé
    - Bi-clef du keybag système
    - Bi-clef éphémère associée au fichier a protéger
  - Le secret partagé chiffre la clé du fichier

ProtectionCompleteUnlessOpen - Création d'un fichier



Source : Evolution of iOS Data Protection and iPhone Forensics (Elcomsoft)
https://media.blackhat.com/bh-ad-11/Belenko/bh-ad-11-Belenko-iOS_Data_Protection.pdf

ProtectionCompleteUnlessOpen – Ouverture d'un fichier



Source : Evolution of iOS Data Protection and iPhone Forensics (Elcomsoft)
https://media.blackhat.com/bh-ad-11/Belenko/bh-ad-11-Belenko-iOS_Data_Protection.pdf

Backup iTunes iOS 4/5

- Les données sont chiffrées coté iOS
- Le terminal génère un Backup Keybag
  - Clés maîtres différentes de celles du keybag système
- Dérivation du mot de passe : PBKDF2
  - 10000 itérations
- Protection du Keychain
  - Pas de mot de passe : clés maîtres protégées par la clé 0x835
  - Avec mot de passe
    - Toutes les clés sont protégées par la passcode key
    - Seules les clés thisDeviceOnly sont en plus protégées par la clé 0x835

Démo

- iPhone 4, iOS 5.1.1
- Démarrage d'un ramdisk avec redsn0w
  - Vulnérabilité limera1n
- Calcul des clés dérivées (0x835 et 0x89B)
- Accès bas niveau à la mémoire Flash
  - Implémentation Python des couches FTL et HFS+

Conclusion

- Les mécanismes cryptographiques d'iOS sont robustes
  - Le code PIN doit être bruteforcé sur le terminal cible
- Mais peuvent être contournés ou attaqués dès que l'accès root est possible
  - Seule l'application Mail utilise Data Protection
- Vulnérabilités BootROM
  - Permettent l'accès aux données et l'attaque du code PIN
  - iPhone 4 et modèles précédents vulnérables
  - iPad 2/iPad 3/iPhone 4S non vulnérables pour l'instant
- Lecture de la Flash NAND et récupération de fichiers effacés
  - http://esec-lab.sogeti.com

Références

- Apple's WWDC 2010 Session 209 – Securing Application Data
- Apple's WWDC 2011 Session 208 – Securing iOS Applications
- http://images.apple.com/iphone/business/docs/iOS_Security_May12.pdf

- http://esec-lab.sogeti.com/dotclear/public/publications/11-hitbamsterdam-iphonedataprotection.pdf
- http://esec-lab.sogeti.com/post/iOS-5-data-protection-updates
- http://cr.yp.to/ecdh.html
- https://media.blackhat.com/bh-ad-11/Belenko/bh-ad-11-Belenko-iOS_Data_Protection.pdf
- http://msftguy.blogspot.com/2012/01/automatic-ssh-ramdisk-creation-and.html
- http://dubeiko.com/development/FileSystems/HFSPLUS/tn1150.html
- http://www.dfrws.org/2008/proceedings/p76-burghardt.pdf
- http://www.theiphonewiki.com
- https://github.com/iDroid-Project/openiBoot
- http://www.freemyipod.org/wiki/Nano2G_FTL
- http://openembed.googlecode.com/hg/som2416/wince5/SMDK2416_WinCE5.0_PM_MLC_NANDSolution_PortingGuide.pdf
- http://storageconference.org/2010/Papers/MSST/Hu.pdf
- http://idke.ruc.edu.cn/people/dazhou/Papers/AsurveyFlash-JSA.pdf
- http://csl.cse.psu.edu/publications/dftl-asplos09.pdf
- http://hal.univ-brest.fr/docs/00/60/73/39/PDF/Papier.pdf
- http://code.google.com/p/iphone-dataprotection/w/list