ELCOMSOFT
PROACTIVE SOFTWARE

OVERCOMING iOS DATA PROTECTION TO RE-ENABLE iPHONE FORENSICS

ANDREY BELENKO
CHIEF SECURITY RESEARCHER
ELCOMSOFT CO. LTD

CONTENTS

SUMMARY

Data protection is a feature available for iOS 4 devices with hardware encryption: iPhone 4, iPhone 3GS, iPod touch (3rd generation or later), and all iPad models. This whitepaper outlines the internal workings of this feature, the implications of the feature for smartphone forensics procedures, and ways to overcome the limitations imposed by this feature.

iPHONE FORENSICS BEFORE iOS 4

Early models of Apple mobile devices – original iPhone and iPhone 3G, as well as first two generations of iPod touch – had no hardware encryption module. Data was stored on user partition unencrypted and could be read by “jailbreaking” the device or by booting custom firmware.

Next iteration of Apple mobile devices – iPhone 3Gs, 3rd generation of iPod touch and original iPad – featured hardware encryption. Those devices were originally shipped with iPhoneOS 3.x, which made limited use of available hardware encryption.

All data on user partition was encrypted with so-called whitening key, and the encryption and decryption was completely transparent for the applications. Main goal of this encryption scheme was to provide the ability to do quick device wipe. With encryption in place it only required to erase whitening key to render data inaccessible. Without such encryption a full disk erase would be needed (e.g. fill whole disk with zeroes) and that could take a lot of time and battery power.

Encryption present in iPhoneOS 3.x had no effect on the ability to obtain decrypted data from the device: decryption was completely transparent for the applications and the device would decrypt any data an application might try to read. With introduction of iOS 4 that was not longer the case.

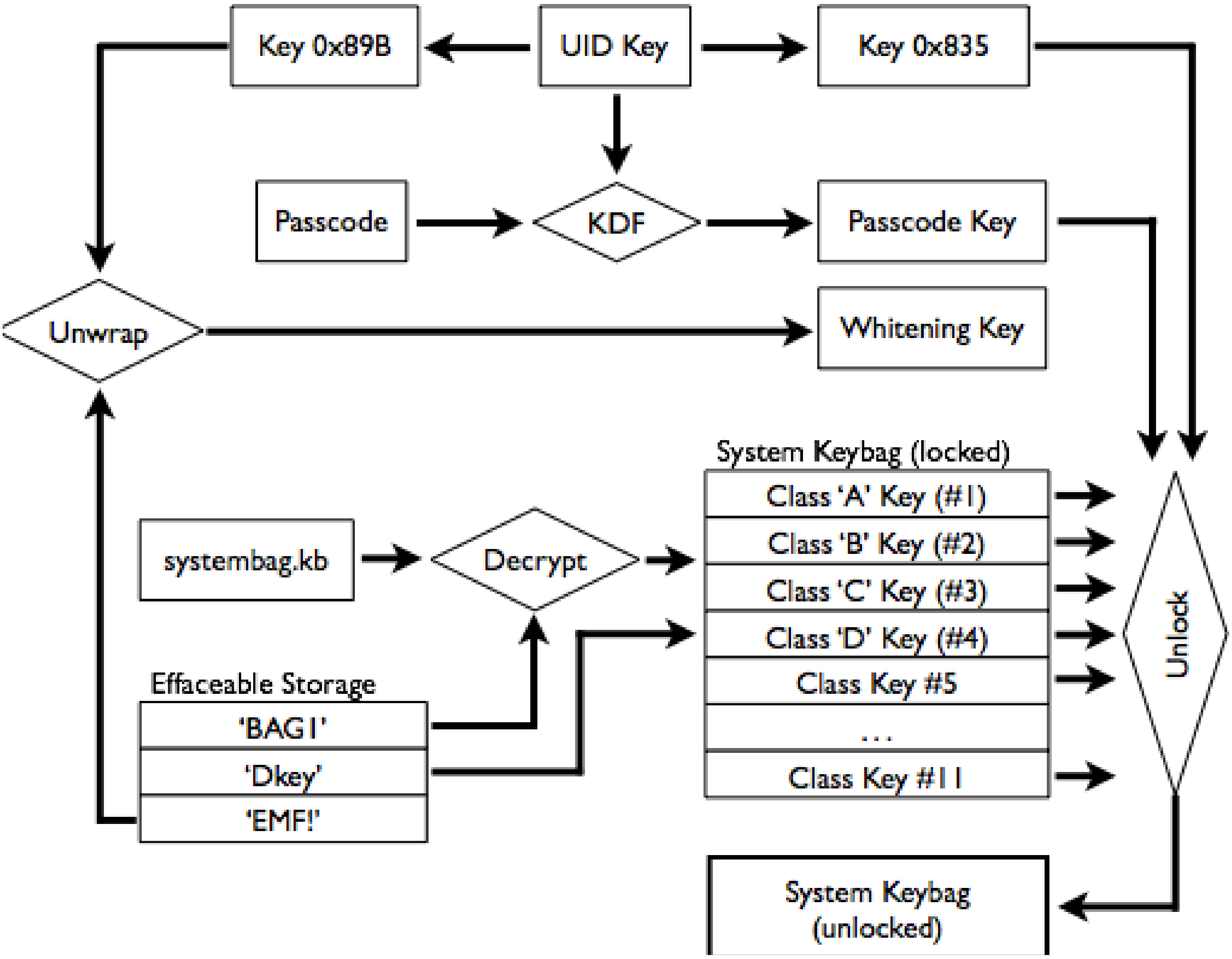
iOS 4 FORENSICS CHALLENGES

iOS 4 further improved device encryption by providing means to encrypt individual files in addition to whitening encryption which was already in place since iPhoneOS 3.x. This means the "default" whitening key (random one) is used to encrypt all data written to user partition unless some other key is specified. iOS 4 utilizes this new feature by encrypting every file it creates with random key. This random key is then put on the lowest level of key hierarchy and can be later obtained for file I/O using one of the root keys from the device keybag (more on this below).

If one tries to obtain disk image using conventional tools and methods, the result will not be satisfactory: only parts of the disk encrypted to "default" whitening key will be decrypted correctly; everything else will remain encrypted. This means that filesystem metadata will be decrypted, but actual file contents will remain encrypted. And since iOS 4 encrypts virtually all files with random keys, the typical resulting image is of very limited value for mobile forensics investigators.

iOS 4 DATA PROTECTION 101

iOS Data Protection feature is designed to protect data “at rest” and to make offline attacks (i.e. when the iOS device is not physically available) difficult. It tries to achieve those design goals by tying encryption keys used for actual data encryption to UID encryption key that is embedded in hardware and is unique to each iOS device. UID key cannot currently be extracted from the device. This tying is not direct but achieved through multi-level key hierarchy. The process is depicted on the following diagram.



Unlocked system keybag and whitening key are the ultimate secrets required to decrypt each and every file on the iOS device as well as every record in the keychain.
System Keybag

Keybag is a collection of protection class keys. Each key is utilized to protect a certain type of files or data on the device. All protection class keys are tied to hardware UID key (via key 0x835 which is computed directly from the UID), and some of them are additionally tied to user-specified passcode. Knowing all 11 protection class keys enables the decryption of all data on the device; knowing only subset of the keys gives the ability to decrypt data belonging to protection classes for which keys are known.

For example, one might not be able to obtain all protection class keys without knowing the passcode, and in this case only files and keychain records not requiring a device to be unlocked will be decrypted.

List of protection classes is shown in the table (protection classes 2, 3, 5 are not used in iOS 4):

ID	Name	Description
1	NSProtectionComplete	File is available only when device is unlocked
4	NSProtectionNone	File is available even when device is locked
6	kSecAttrAccessibleWhenUnlocked	Keychain item is available only when device is unlocked
7	kSecAttrAccessibleAfterFirstUnlock	Keychain item is available only after device has been unlocked
8	kSecAttrAccessibleAlways	Keychain item is available even when device is locked
9	kSecAttrAccessibleWhenUnlockedThisDeviceOnly	Same as kSecAttrAccessibleWhenUnlocked and keychain item is not included in backup
10	kSecAttrAccessibleAfterFirstUnlockThisDeviceOnly	Same as kSecAttrAccessibleAfterFirstUnlock and keychain item is not included in backup
11	kSecAttrAccessibleAlwaysThisDeviceOnly	Same as kSecAttrAccessibleAlways and keychain item is not included in backup

PASSCODE RECOVERY

As shown in the diagram above, key derivation function (KDF) to transform user-supplied passcode into passcode key depends on hardware UID key. Because of this fact passcode bruteforce attempts can be made only on the device itself.

It is possible to bruteforce passcode without triggering the “Wipe after 10 incorrect passcode attempts” protection.

ESCROW KEYBAG

Escrow keybag is, according to Apple, a feature to improve usability. It allows iTunes to unlock the device (when syncing or creating backup, for example) without asking user to enter the passcode. Escrow keybag is stored on the computer and is created when the device is connected to the iTunes for the first time.

Escrow keybag contains the very same protection class keys as the system keybag and is protected by 32-byte random “passcode” which is stored on the iOS device. Because it stores same protection class keys as original system keybag, escrow keybag allows one to bypass the passcode protection and decrypt all files and keychain records even if passcode is not known for the device.

REFERENCES

iOS 4: Understanding data protection
by Apple
http://support.apple.com/kb/HT4175