iOS ENCRYPTION
Lecture 19b

COMPSCI 702
Security for Smart-Devices

Nalin Asanka Gamagedara Arachchilage

Slides from Muhammad **Rizwan** Asghar

April 27, 2021

DATA PROTECTION

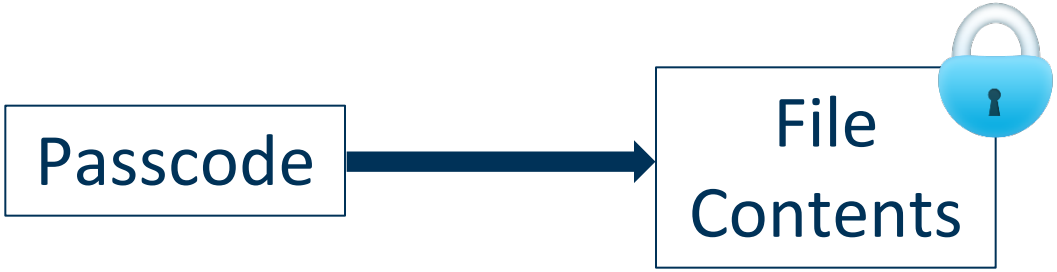
Do we need to protect data stored on the device?

WHY DATA PROTECTION

- An increased risk of sensitive data compromise through
  - A lost device
  - A stolen device
- Data protection features safeguard sensitive data
- How can we protect data on the device?

ATTEMPT 1: PROTECTION USING PASSCODE

- Encrypt each file using a passcode

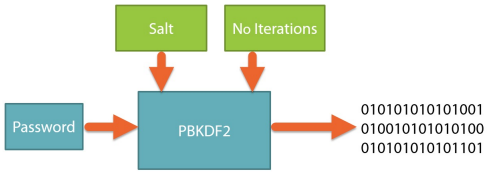


- Main issues
  - Brute-forcing
  - Changing a passcode will require re-encryption of file content using a new passcode

HARDENING BRUTE-FORCING

- Can we achieve better security using passcodes?
- Yes, we can stretch passcodes using KDF (Key Derivation Function)
  - Derive a secret key from the passcode

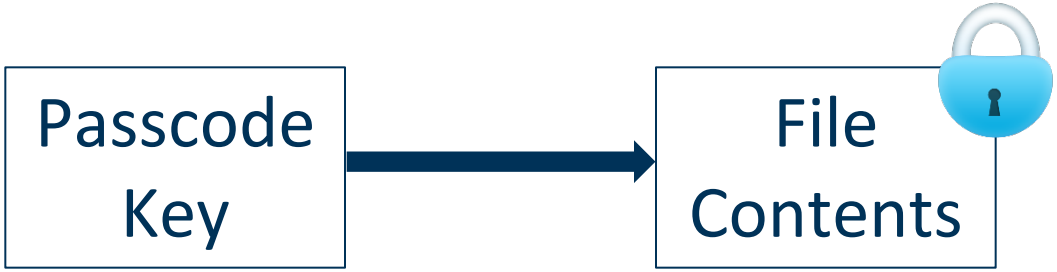
PBKDF2



- A Password-Based Key Derivation Function
- *DK* = PBKDF2(*password, salt, count, klen*)
  - *password* is the master password from which a derived key is generated
  - *salt* is a random sequence of bits
  - *count* represents number of iterations
  - *klen* is the desired length of the derived key
  - *DK* denotes the desired key
- It is known as a key stretching function
- The added computational work makes password cracking much more difficult
- Using PBKDF2, we can generate a **passcode key** from the passcode

ATTEMPT 2: PROTECTION USING PASSCODE KEY

- Encrypt each file using the passcode key



- Main issues
  - Brute-forcing is still possible
  - Changing the passcode will require re-encryption of file contents using the new passcode key

ATTEMPT 3: LAYERED PROTECTION

- Encrypt each file using a file key
- Encrypt each file key using the passcode key

- Changing a passcode will just require re-encryption of the file key using the new passcode key
- Now, the issue is the passcode guessing attack!

PASSCODE GUESSING (iPHONE 4)

- 9.18 guesses per second

Passcode Length	Complexity	Worst Case Time
4	Numeric	18 minutes
4	Alphanumeric	51 hours
6	Alphanumeric	8 years
8	Alphanumeric	13 thousand years
8	Alphanumeric, Complex	2 million years

Source:
"iOS Hacker's Handbook"

- Alphanumeric: 10 digits and both upper and lower-case letters
- Alphanumeric, complex: adds 35 symbols

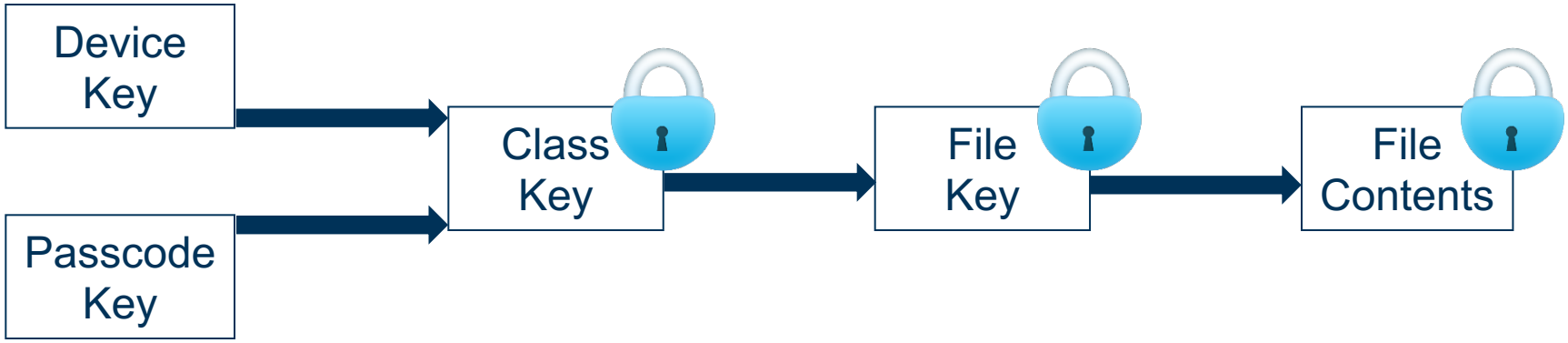
UID AND GID

- UID key is unique to the device
  - An AES 256-bit hardware key
  - Inside the cryptographic accelerator
  - Cannot access it through software
  - This encrypts a static string to produce the **device key**
  - Neither recorded by Apple nor by any of its suppliers

- There is also a GID key unique to the device model
  - Used to decrypt iOS firmware images

iOS FILE DATA PROTECTION

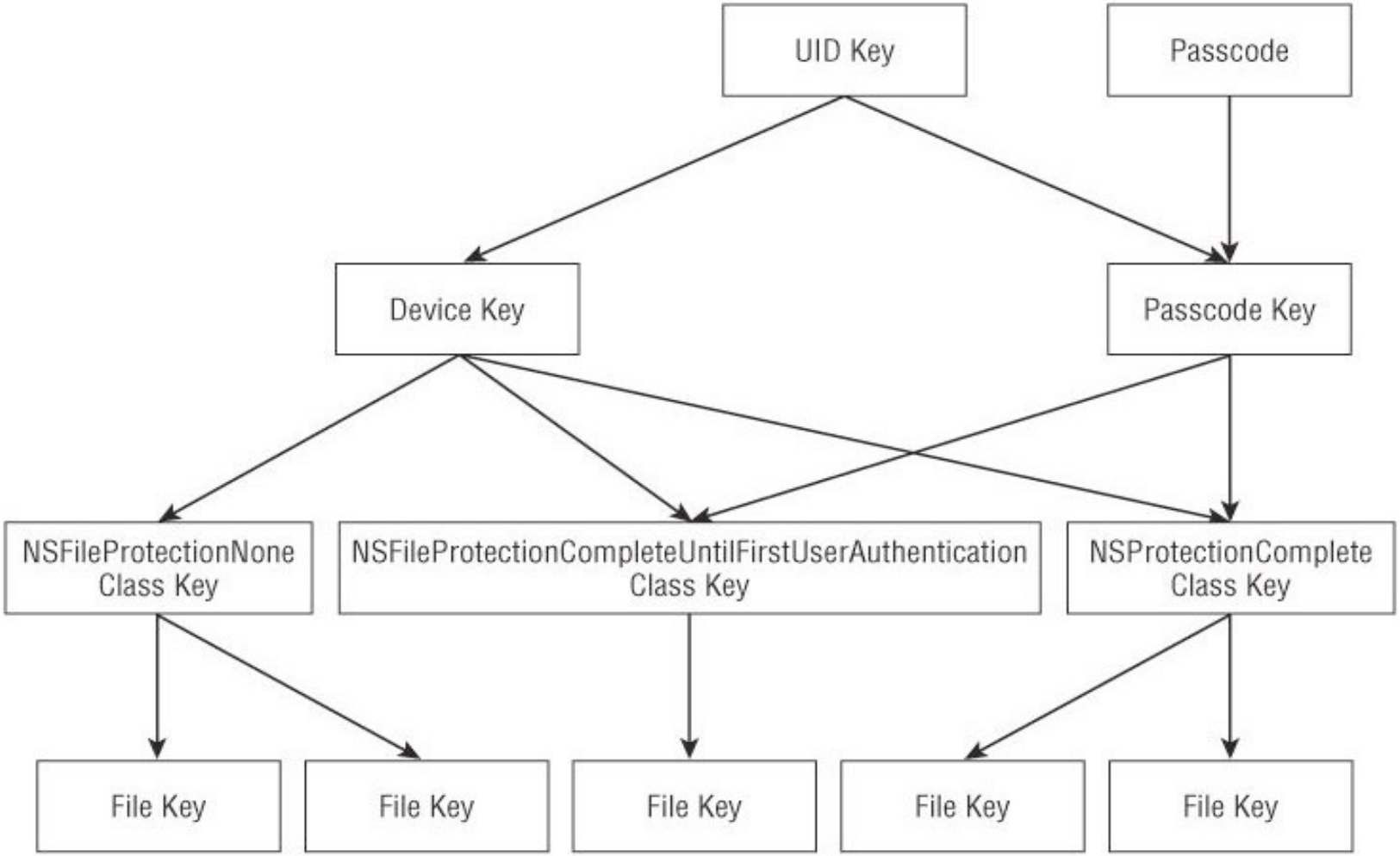
- A file key is encrypted using a class key
- Different class keys represent circumstances under which files should be accessed
- Device key and passcode key are used to protect class keys



FILE PROTECTION CLASSES

Protection Class	Description
NSFileProtection **Complete**	The file can be accessed only when the device is unlocked – the class key is **removed 10s after** a device gets locked either automatically or manually (useful for, e.g., **mobile banking**)
NSFileProtection **Complete Unless Open**	The file can be opened when the device is unlocked – once opened, it could be accessed afterwards, even if the device gets locked (useful for, e.g., **email** while the device is locked)
NSFileProtection **Complete Until First User Authentication**	The file is protected unless the device is booted and the user enters the passcode for the first time (useful for, e.g., **SIM contacts**)
NSFileProtection **None**	The file is not protected and it can be accessed at any time (useful for things like background **wallpaper**)

THE KEY HIERARCHY



Source: "iOS Hacker's Handbook"

TO BE CONTINUED

- See the next lecture

Questions?

Thanks for your attention!