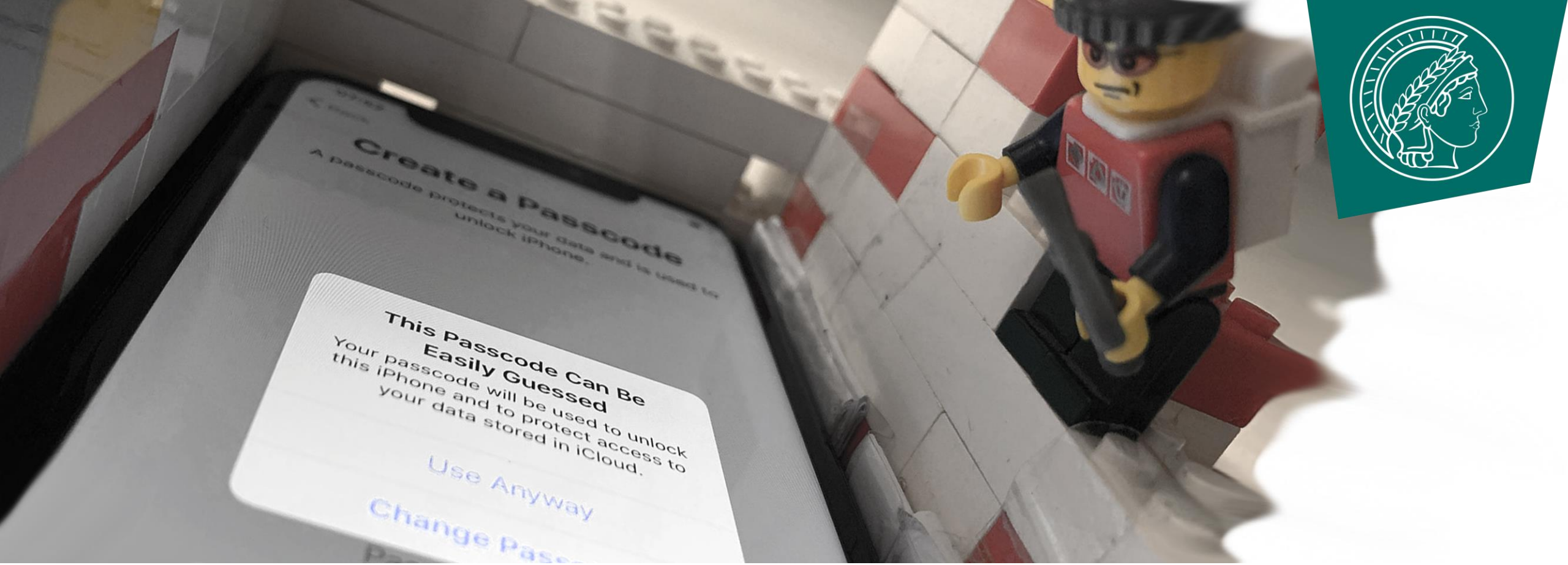


Max Planck Institute | Security and Privacy

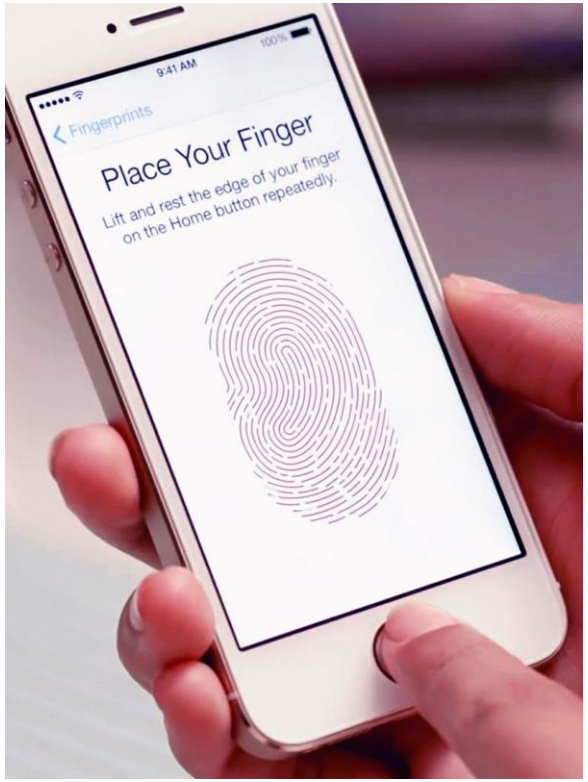
Extracting iOS' Passcode Blacklist

Maximilian Golla

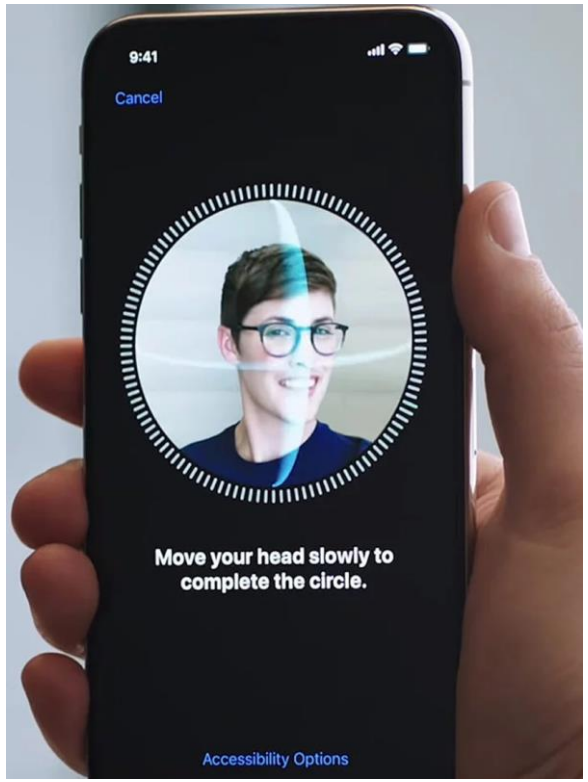
March 2020 | Bochum, Germany

Biometric-Based Reauthentication

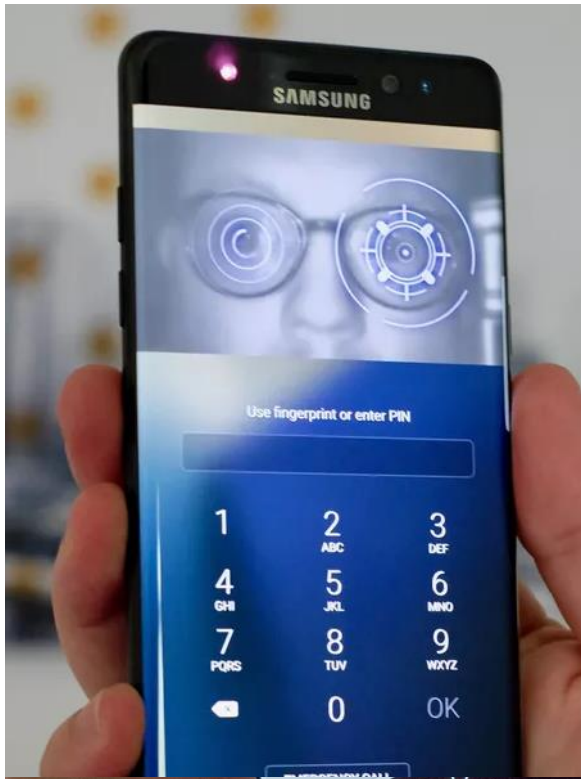
"Face unlock feels almost like not having any lock screen security."



Fingerprint



Face

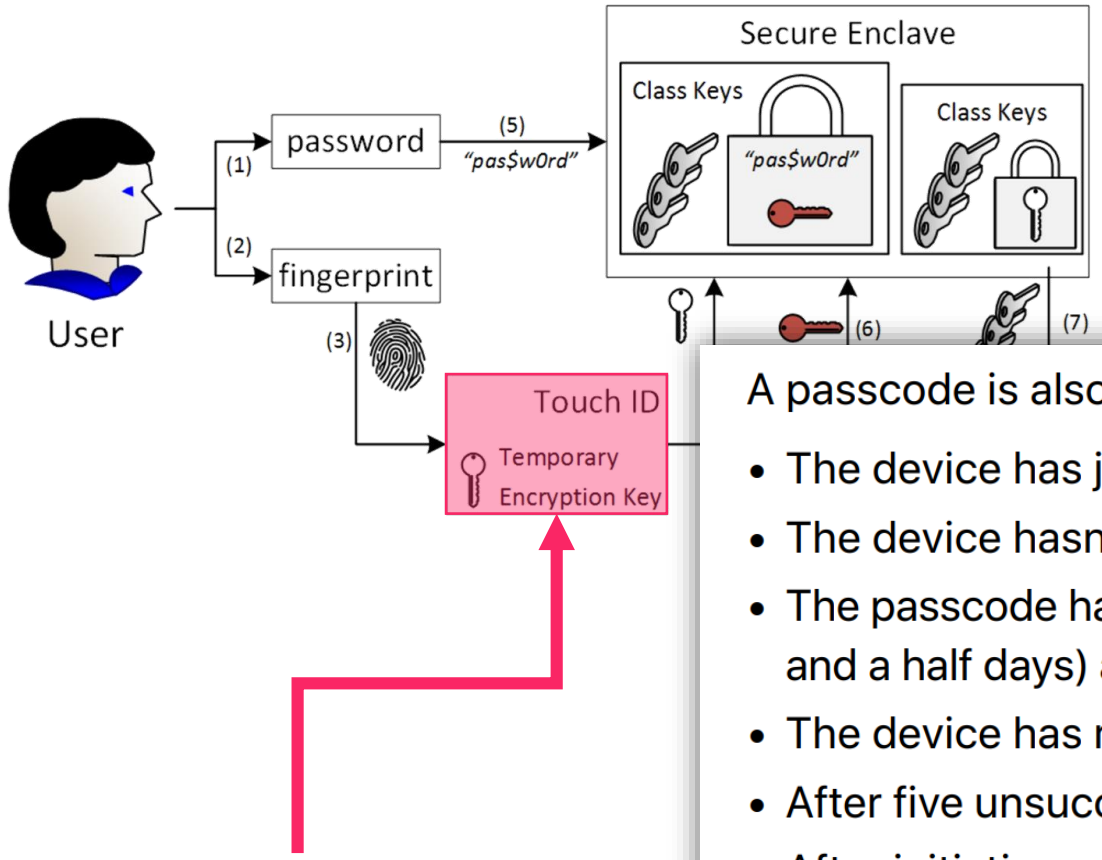


Iris

"Intelligent Scan"

[Fingerprint] apple.com; [Face] apple.com; [Iris] Jason Cipriani/CNET

Biometrics are a Convenience Feature!



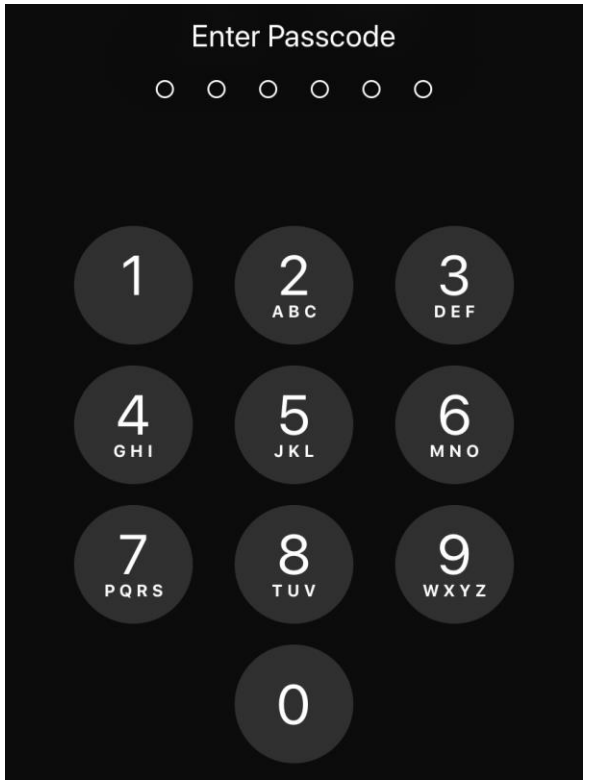
A passcode is also required if your device is in the following states:

- The device has just been turned on or restarted.
- The device hasn't been unlocked for more than 48 hours.
- The passcode hasn't been used to unlock the device in the last 156 hours (six and a half days) and a biometric hasn't unlocked the device in the last 4 hours.
- The device has received a remote lock command.
- After five unsuccessful biometric match attempts.
- After initiating power off/Emergency SOS.

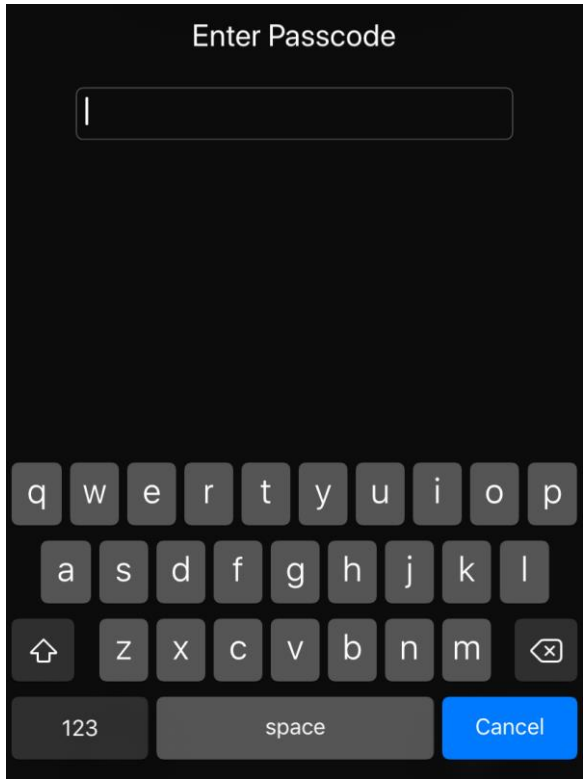
TEK is flushed periodically and on device reboot.

[Ref. 1] Cherapau et al.: **On the Impact of Touch ID on iPhone Passcodes**. (SOUPS '15)
[Ref. 2] **Apple Platform Security** – Fall 2019 – PDF Version Page 10

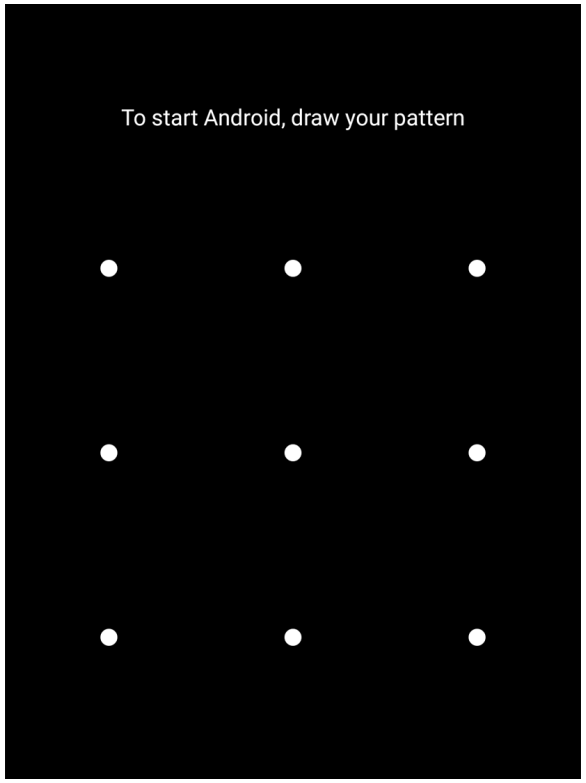
Knowledge-Based Authentication



4/6-digit PINs



Passwords



Pattern

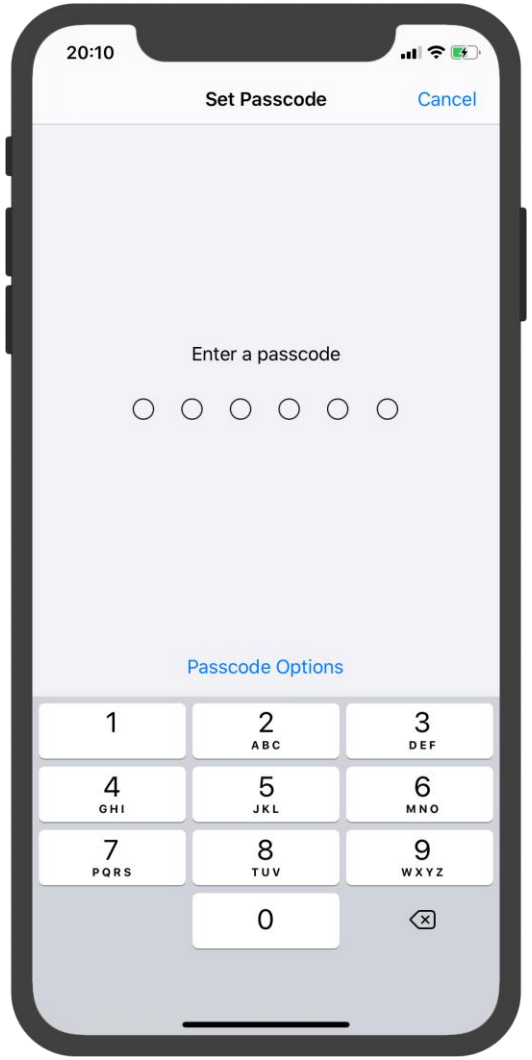
iOS Passcode

Knowledge-based auth. scheme

- **6-Digit Numeric Code (default)**
- 4-Digit Numeric Code
- Custom Numeric Code
- Custom Alphanumeric Code

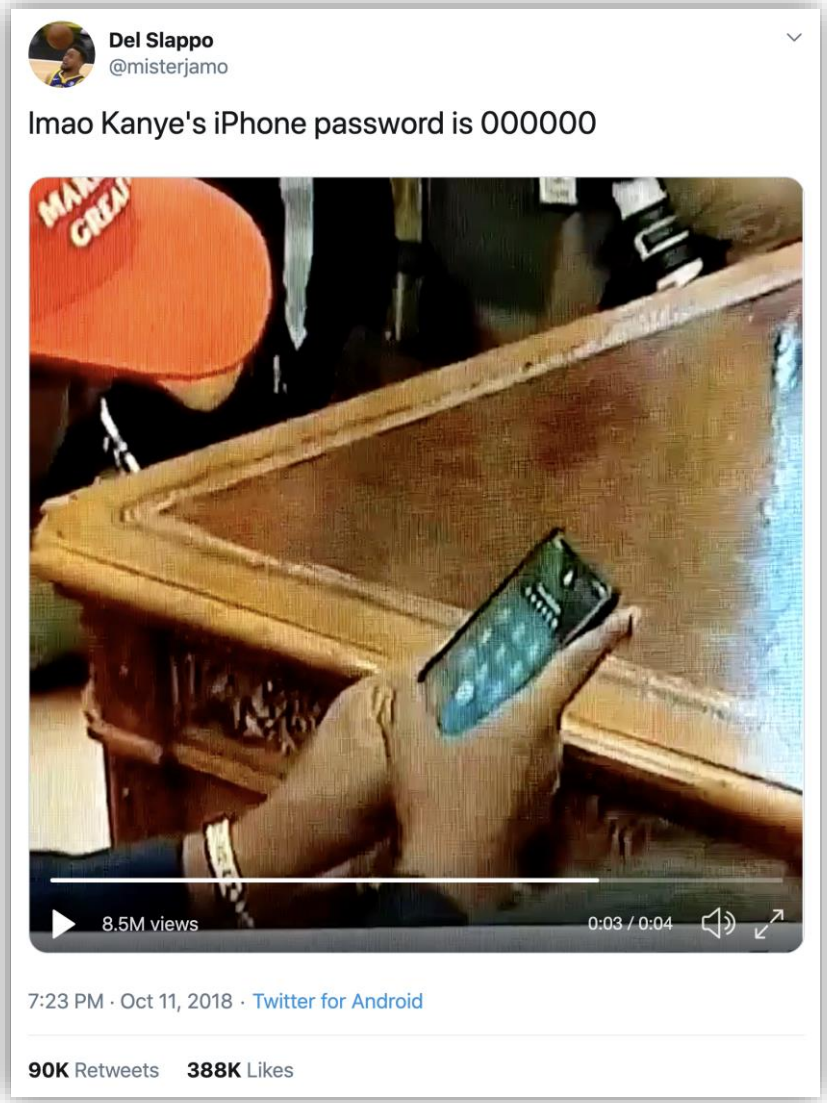
Strict rate-limiting (10 guesses)

Enrollment



Authentication

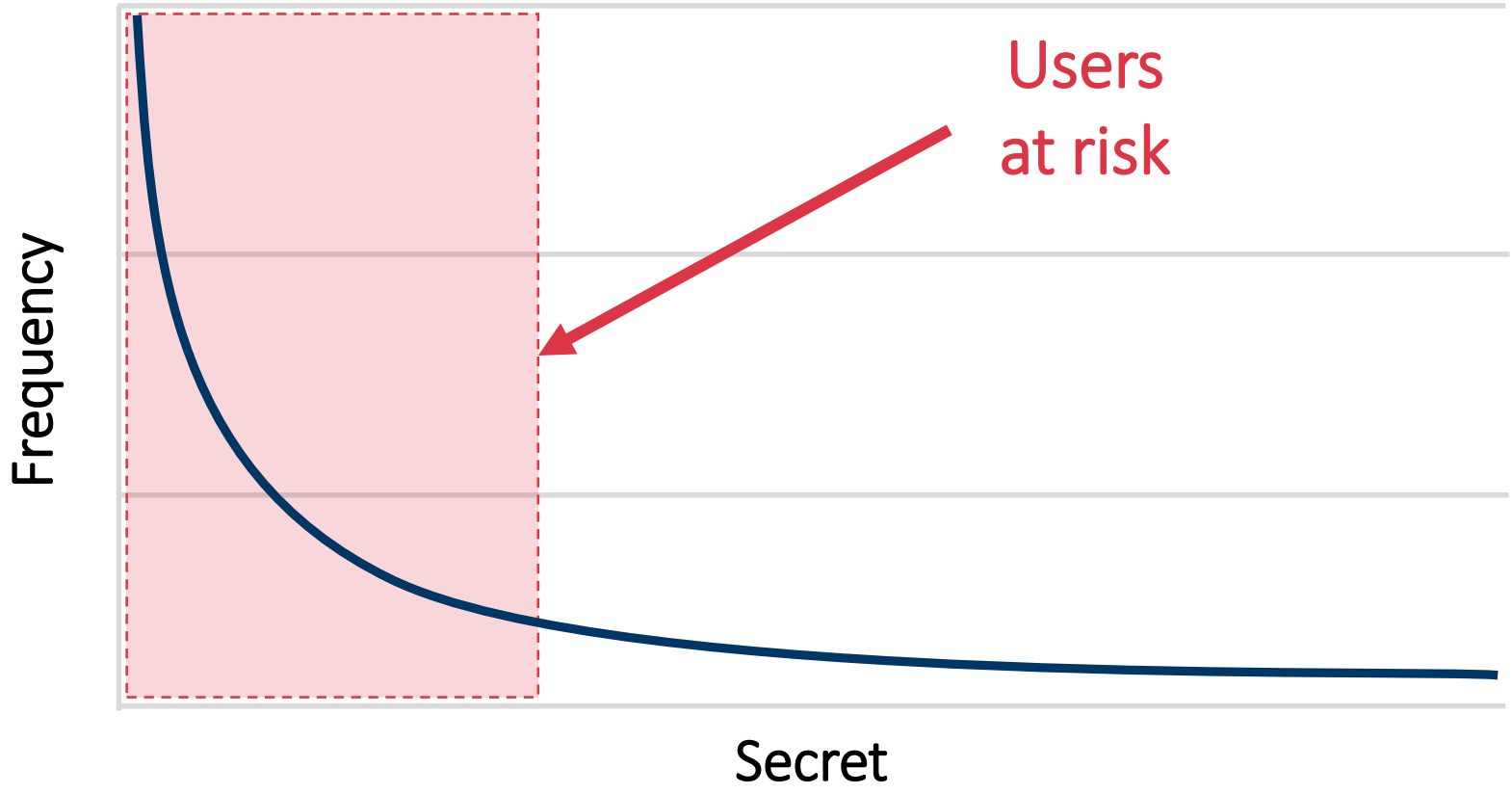
User-Chosen PINs



 [Tweet] https://twitter.com/misterjamo/status/1050436638520823809

Selection Bias

User-choice heavily biased.

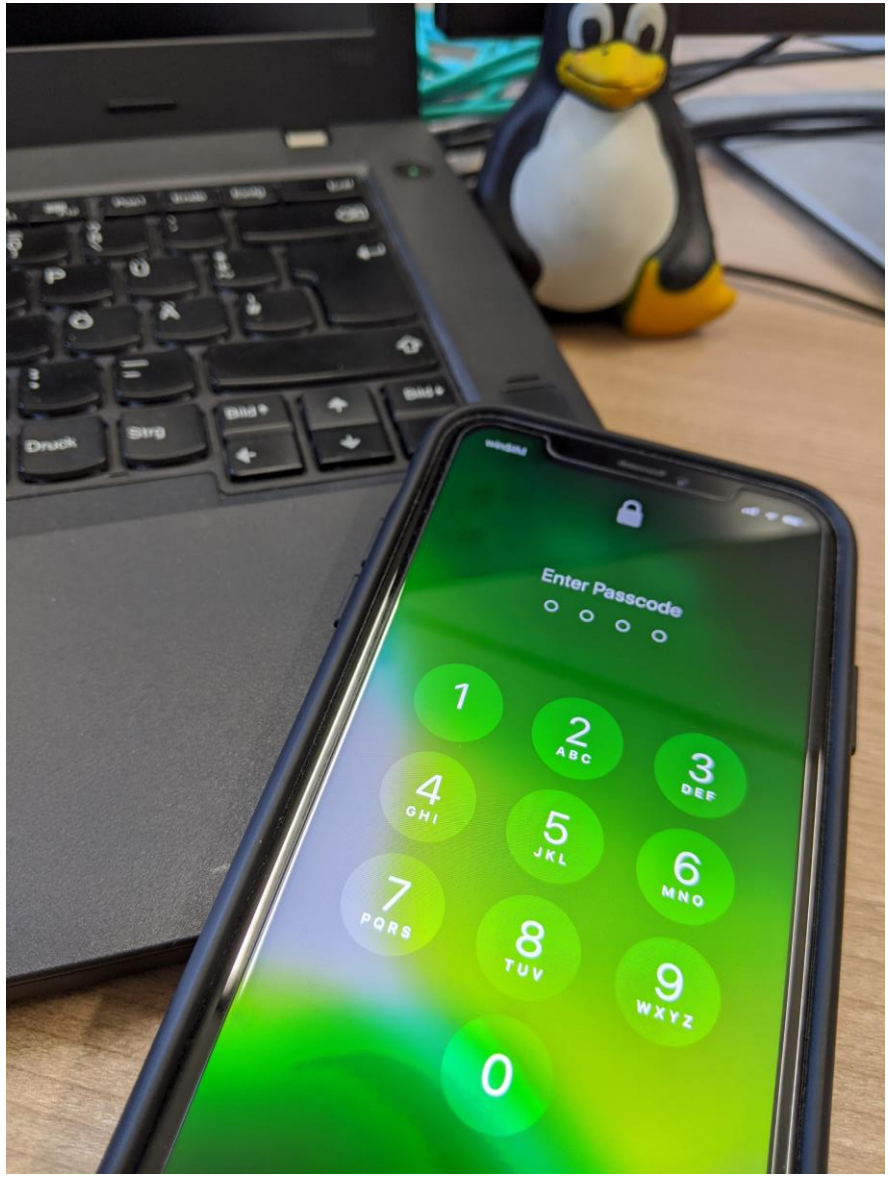


Threat Model

Attacker guesses the ***n*** most common secrets in decreasing order of success.

Throttled Guessing Attack:

	iOS 9-13	Android 7-10
3 Guesses	00s	00s
10 Guesses	1h 36m 00s	30s
30 Guesses	Disabled	10m 30s
100 Guesses	Disabled	10h 45m 30s



What This Talk is Not About!

~~iPhone unlocking as used by law enforcement.~~

→ We only extract the iOS Passcode blacklist!

 [Photo] GrayKey by MalwareBytes (Thomas Reed) - blog.malwarebytes.com

iOS Passcode Blacklist

- “Blacklist” consisting of weak PINs
- Not documented
- Allows users to click-through (“Use Anyway”)

Examples:

“000000”

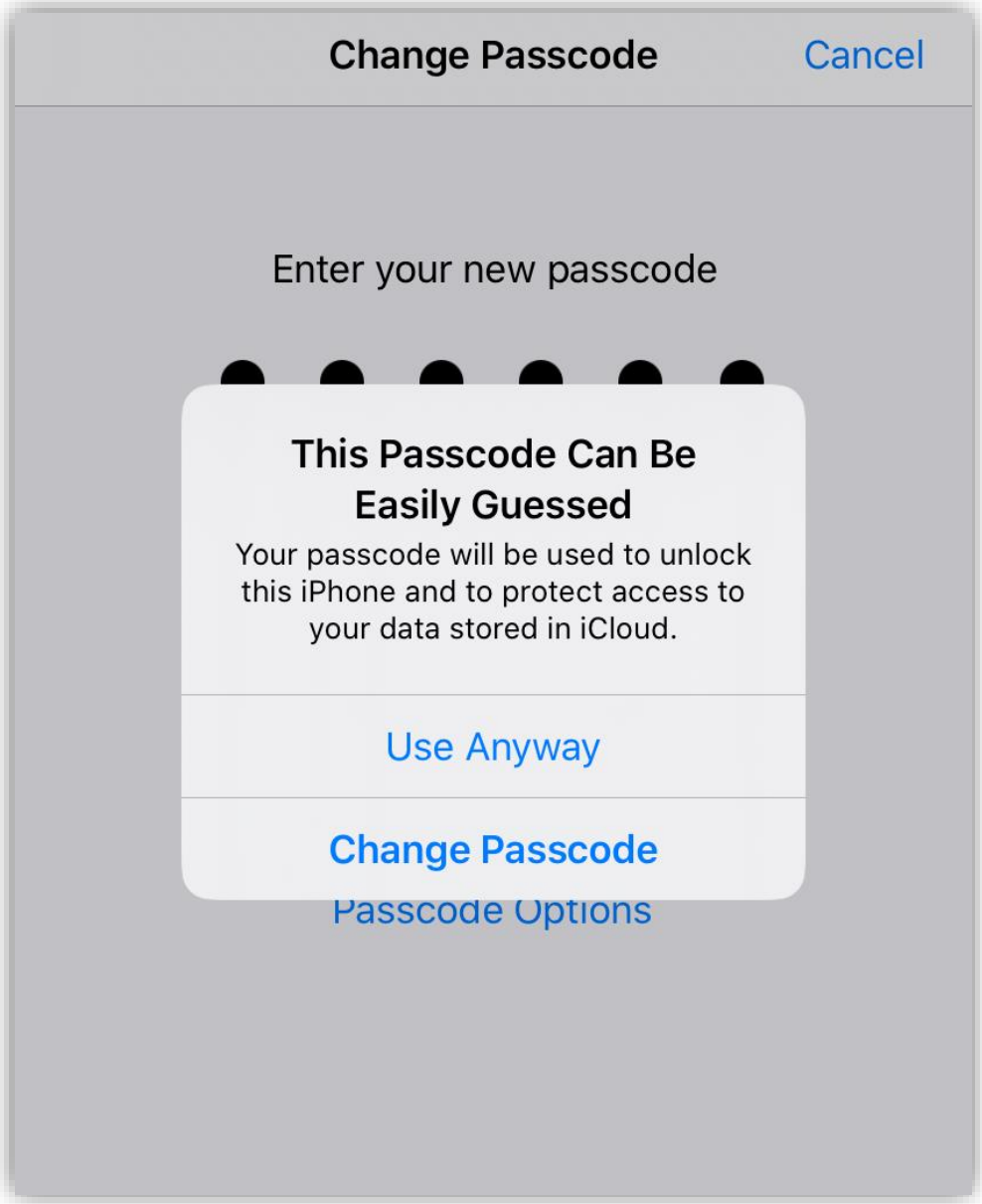
“123456”

or

“2580”

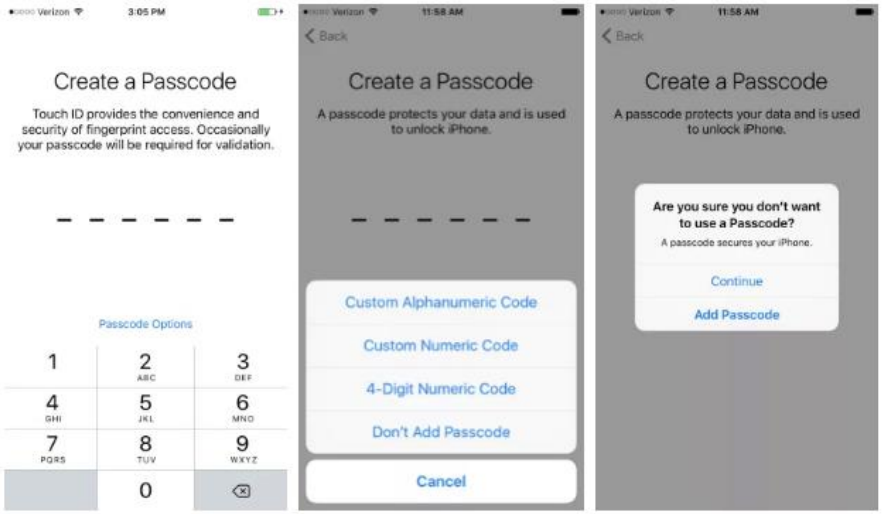
“1956”

How to obtain the only “available” PIN blacklist on the market?

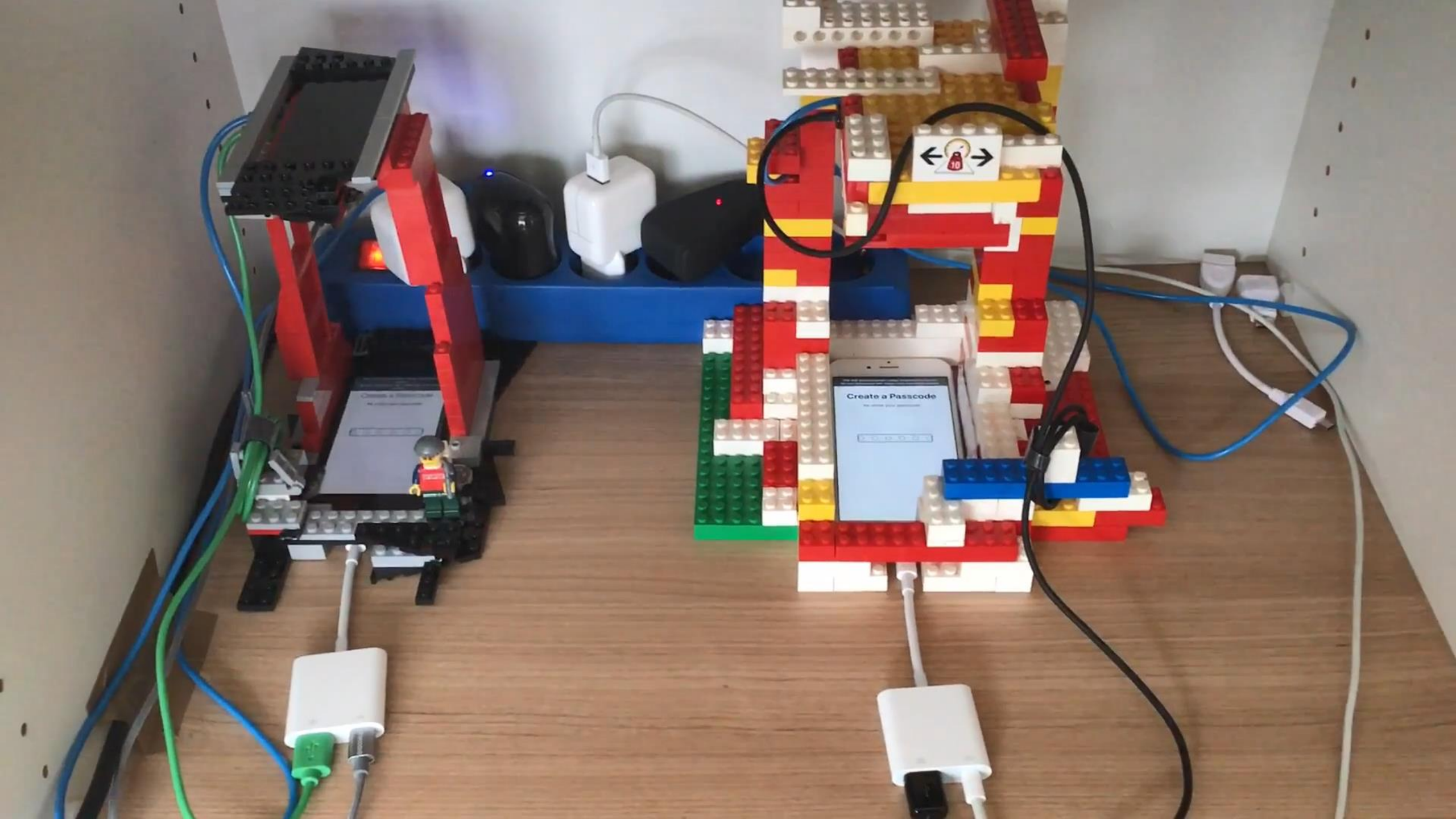


Documentation?

When creating a Passcode, the system is smart and does detect if you are wanting to use a simple, easy to guess, commonly used passcode and it has you confirm whether or not you want to do it. Examples of easy to guess, commonly used passcodes are 000000, 111111, 222222, 123456, etc. Before iOS 7, if you use a simple, easy to guess, commonly used passcode it did allow you to proceed and create the passcode without questioning you. Having a passcode is not required, and you can choose to not have a passcode by tapping on the Passcode Option. However, if you want to use iCloud Keychain and Touch ID, it is required to have a passcode.

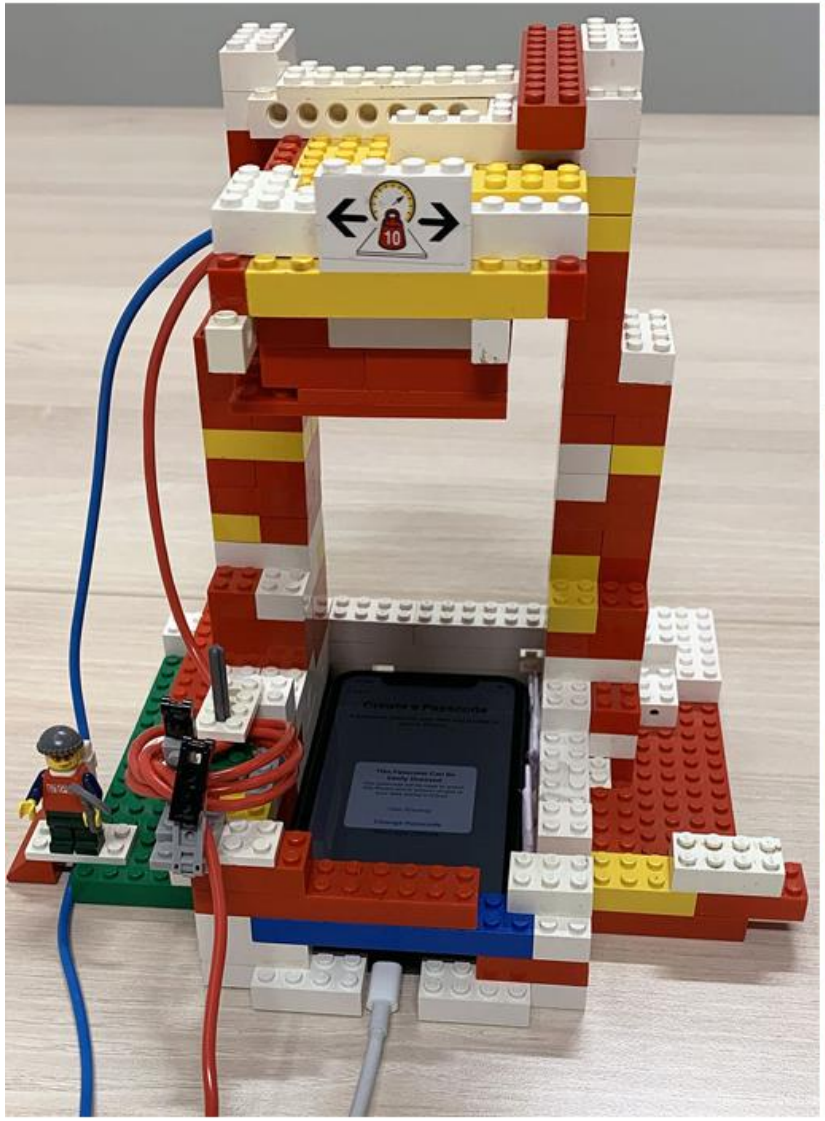


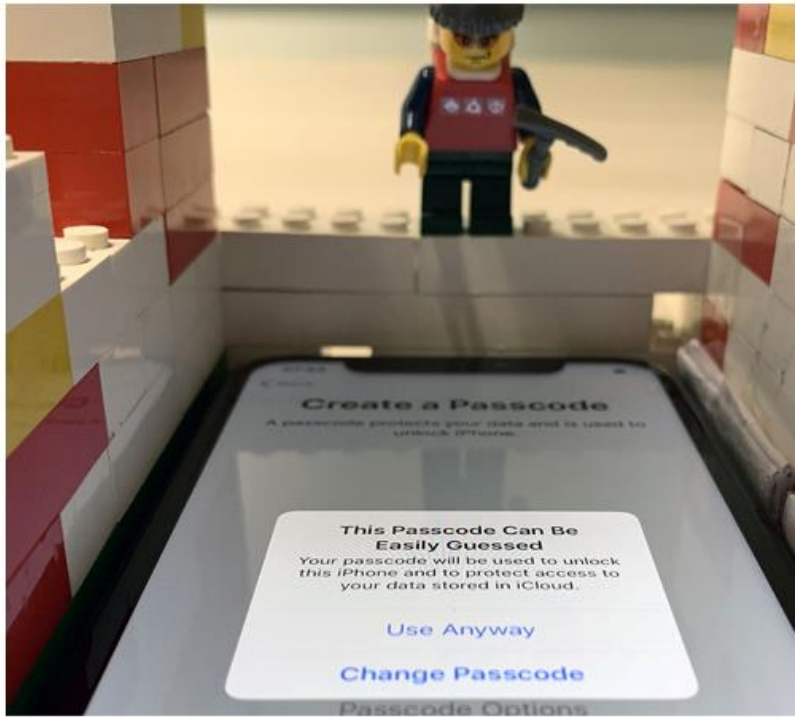
 [Screenshot] "Getting Ready for iOS 9" by Sarah Guarino – 9to5mac.com

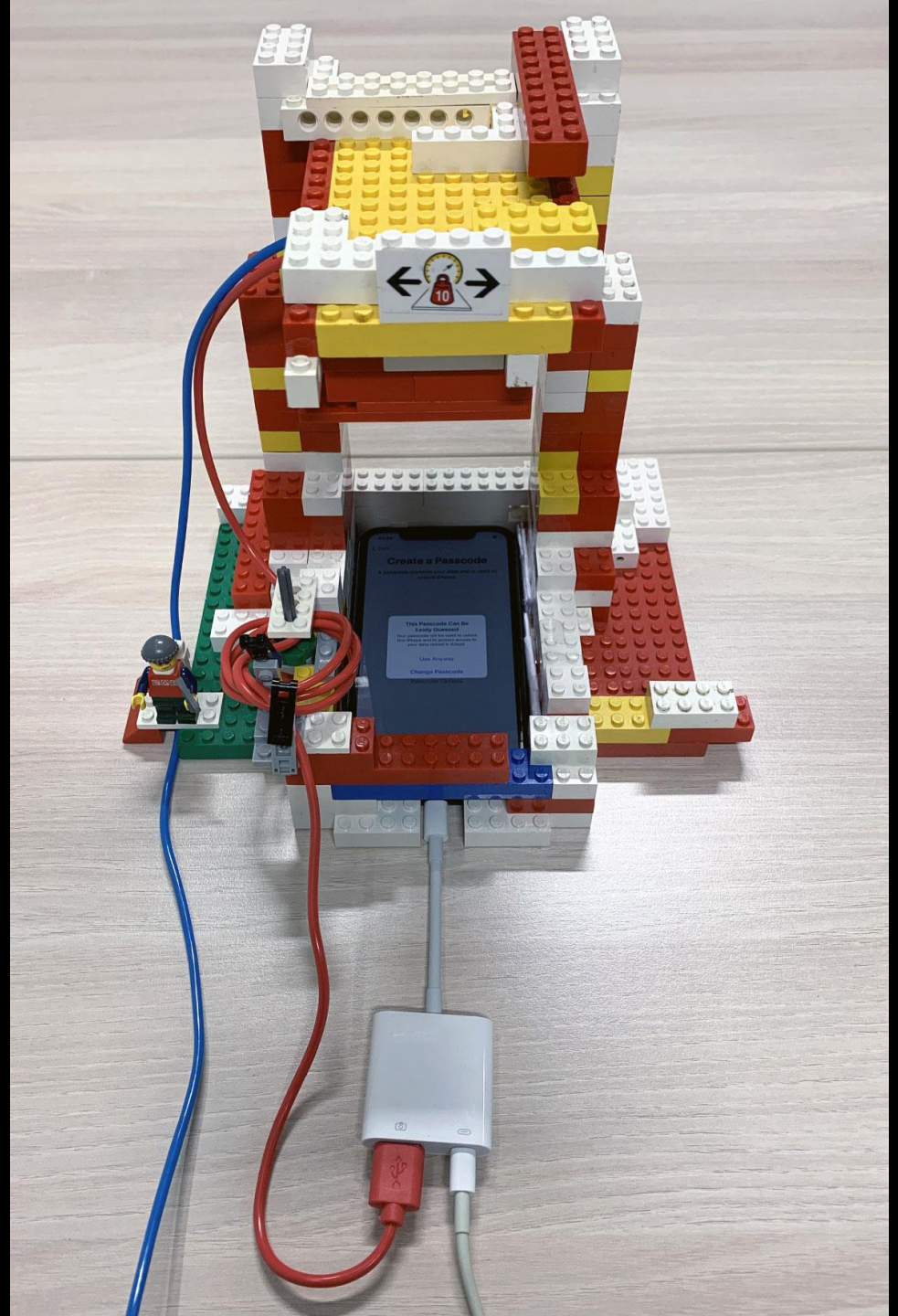

Create a Passcode

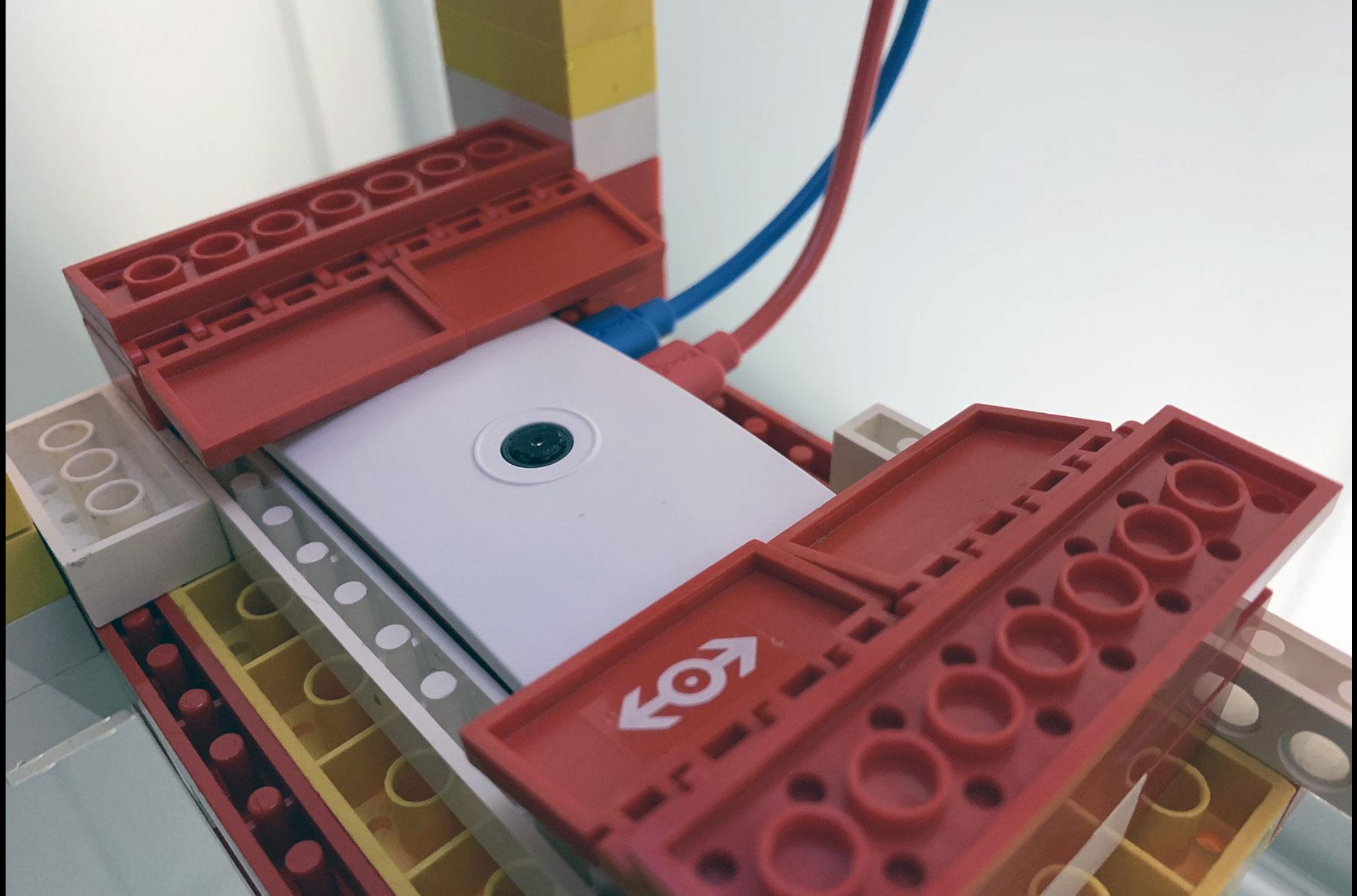
Create a Passcode
The iOS screen reader called VoiceOver is active.
To turn VoiceOver off, triple-click the Home button

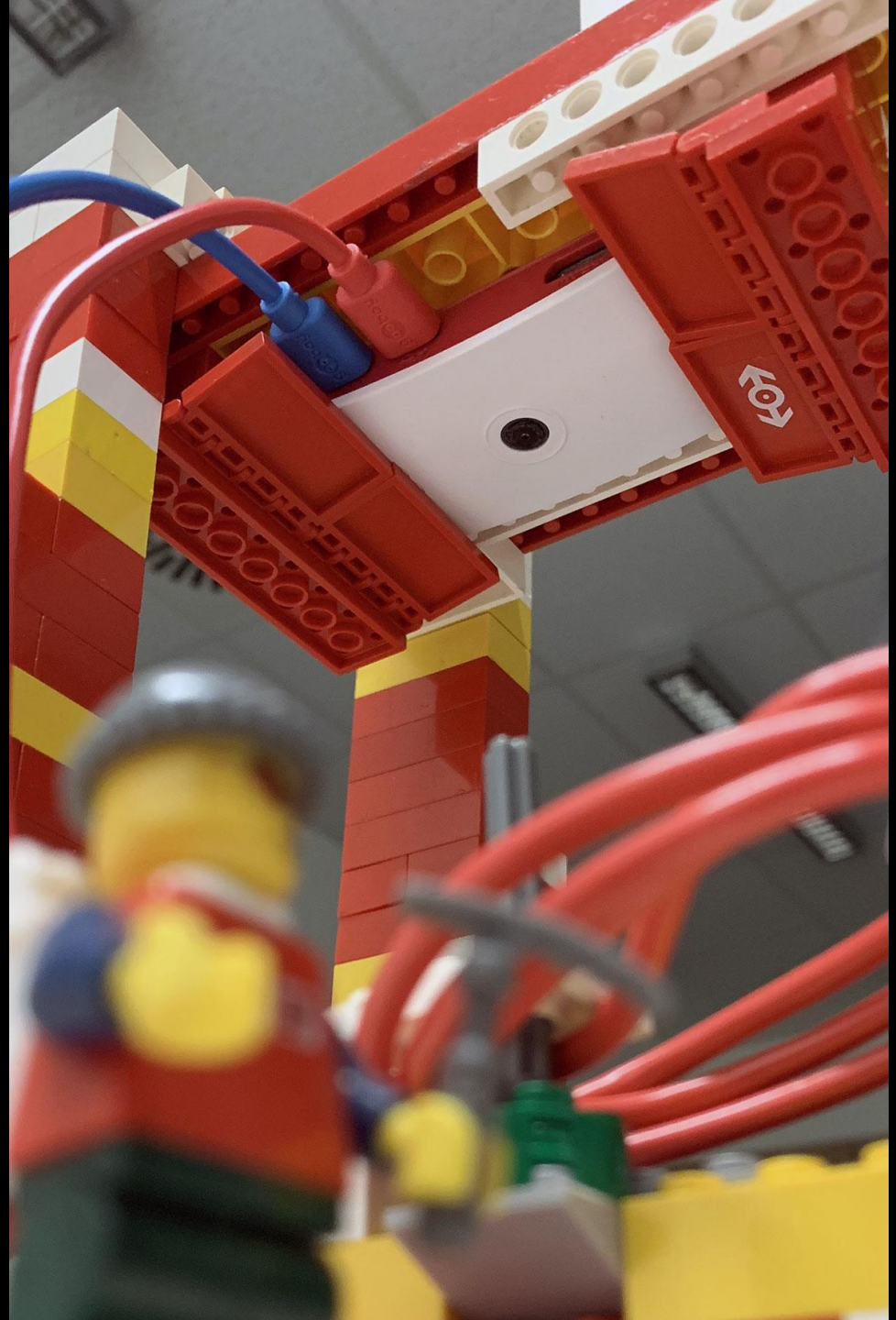
Interlocking Plastic Bricks Robot

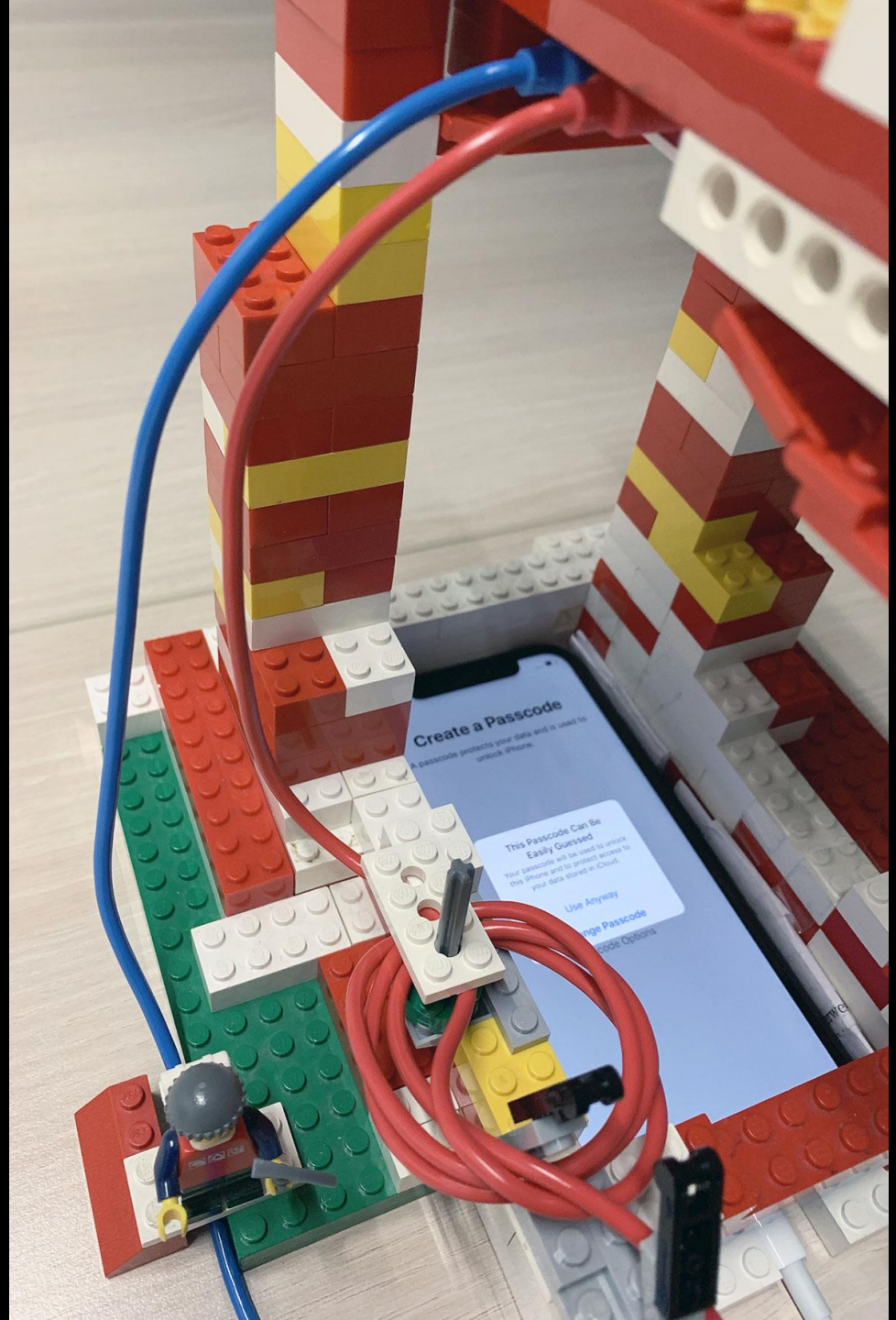


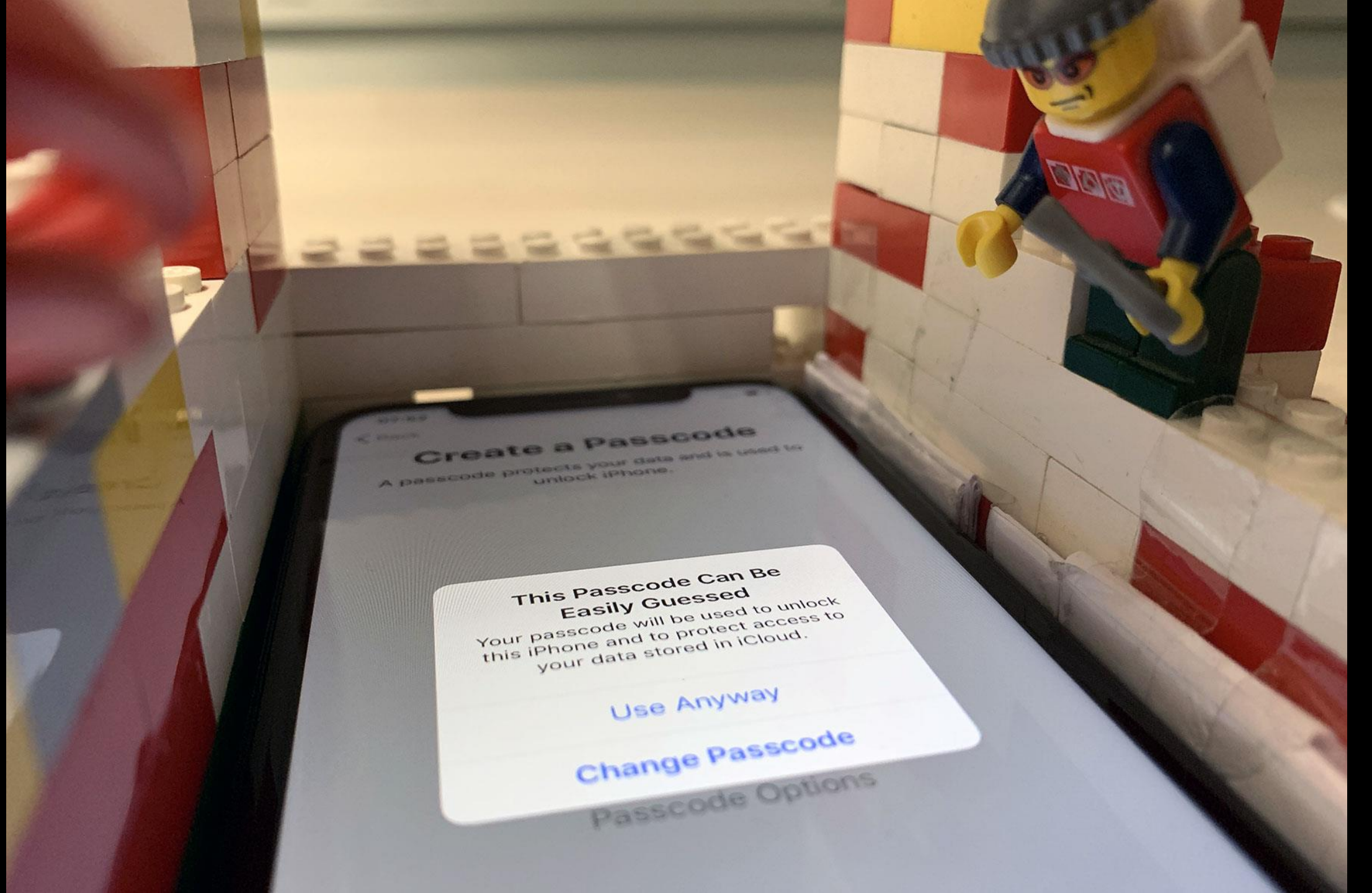


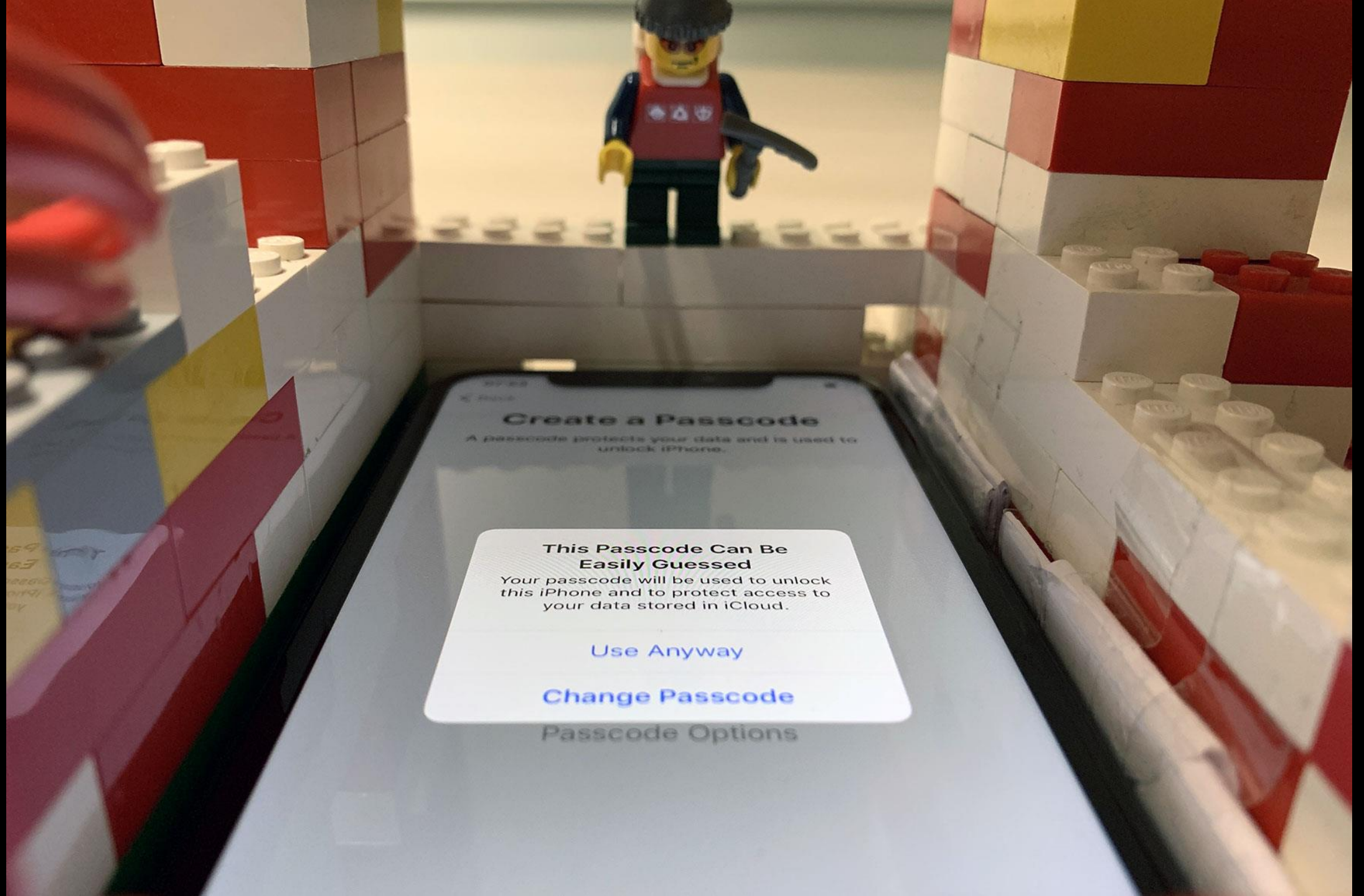

Create a Passcode
This Passcode Can Be Easily Guessed
Change Passcode



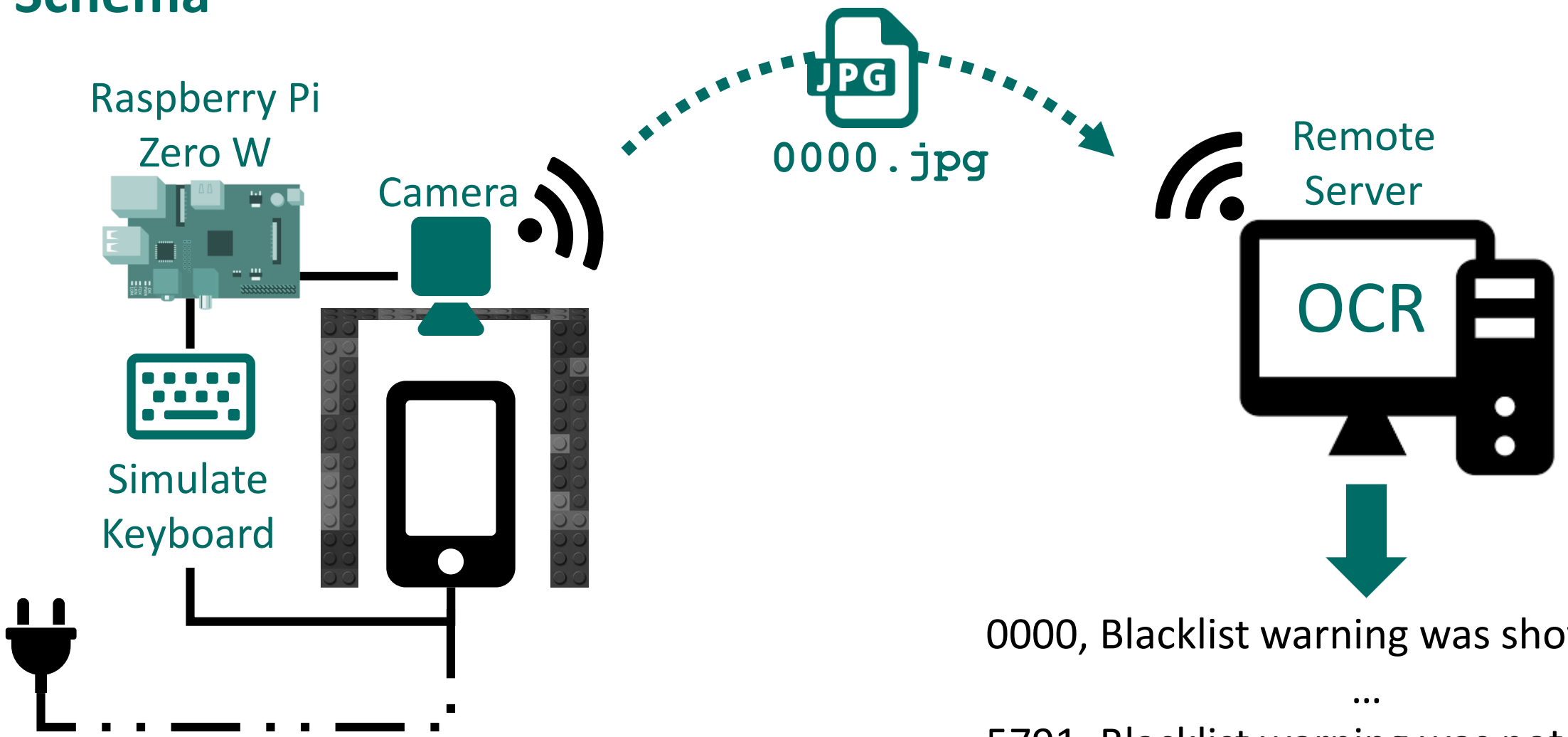



Create a Passcode
This Passcode Can Be Easily Guessed
Use Anyway


Create a Passcode
A passcode protects your data and is used to unlock iPhone.
This Passcode Can Be Easily Guessed
Your passcode will be used to unlock this iPhone and to protect access to your data stored in iCloud.
Use Anyway
Change Passcode
Passcode Options


Create a Passcode
A passcode protects your data and is used to unlock iPhone.
This Passcode Can Be Easily Guessed
Your passcode will be used to unlock this iPhone and to protect access to your data stored in iCloud.
Use Anyway
Change Passcode
Passcode Options

Schema



0000, Blacklist warning was shown

...

5791, Blacklist warning was not shown

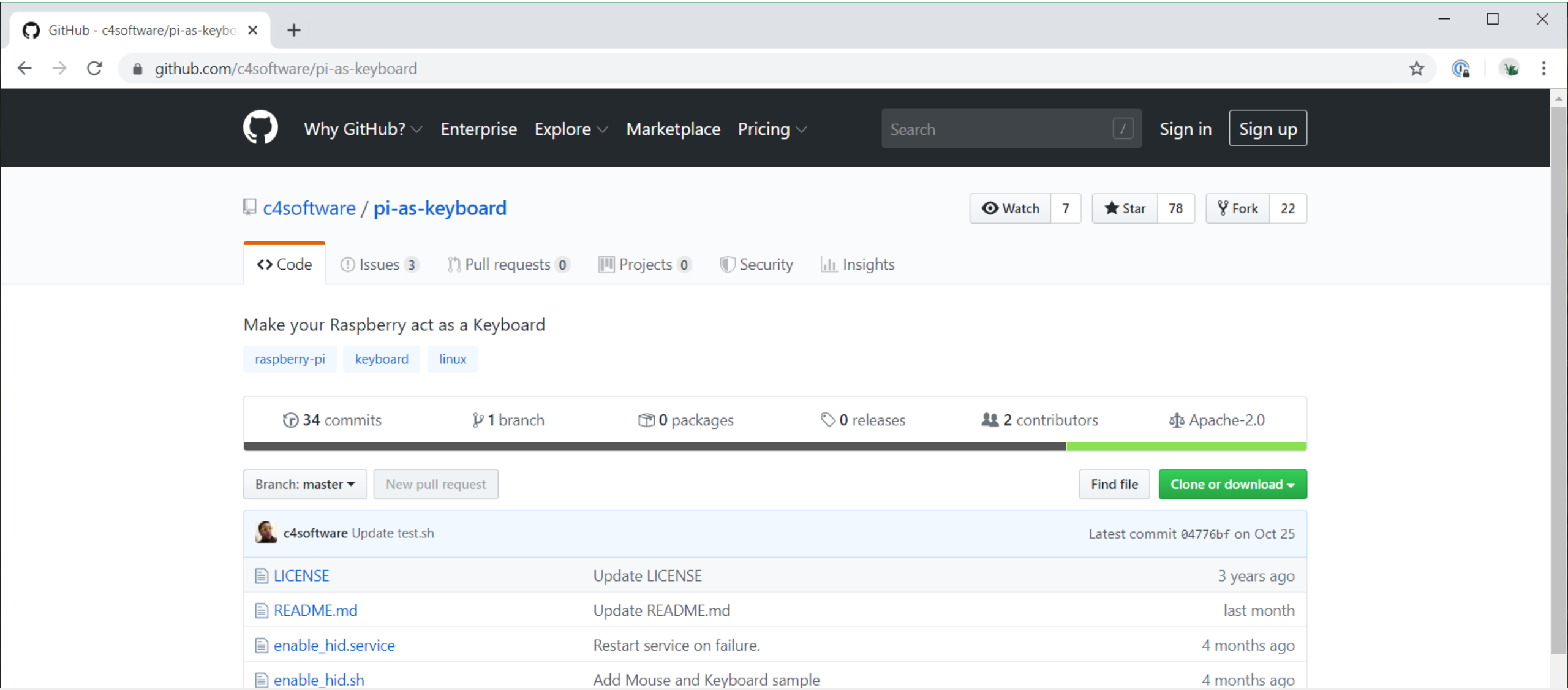
 [Icon] Raspberry Pi by OpenClipart

Parts List (~$100 + Phone)

- 1x Raspberry Pi Zero W – $30
  - ✓ Case
  - ✓ Power supply
  - ✓ Micro SD card
- 1x Raspberry Pi Camera Module v2 – $25
- 1x Micro USB cable – $5
- 1x Lightning to USB 3 Camera Adapter – $39
- 1x Apple iPhone 6s (or newer) – ~$150
- Some interlocking plastic bricks

 [iPhone] apple.com; [USB-Adapter] apple.com; [Plastic Bricks] “Block party” by The Lego Group

Turn a Raspberry Pi Zero W Into a Keyboard



 [Screenshot] https://github.com/c4software/pi-as-keyboard by Brosseau Valentin

Turn a Raspberry Pi Zero W Into a Keyboard

```
50 # OS descriptors
51 echo 1          > os_desc/use
52 echo 0xcd       > os_desc/b_vendor
53 echo MSFT100 > os_desc/qw_sign
54 ln -s configs/c.1 os_desc
55
56 mkdir -p configs/c.1/strings/0x
57 echo "Config 1: Keyboard" > con
58 echo 100 > configs/c.1/MaxPower
59 ls /sys/class/udc > UDC
```

~~250~~ -> 100 (**milliwatts**)

[Photo] “How to connect a USB Sound Interface that draws too much power to an iPad” by Gary Judge – head4space.com

Idea

Test all possible PINs, e.g., via a keyboard

Problems:

- **Keyboard navigation**
- Rate-Limiting
- Battery

How to navigate in iOS using a keyboard?

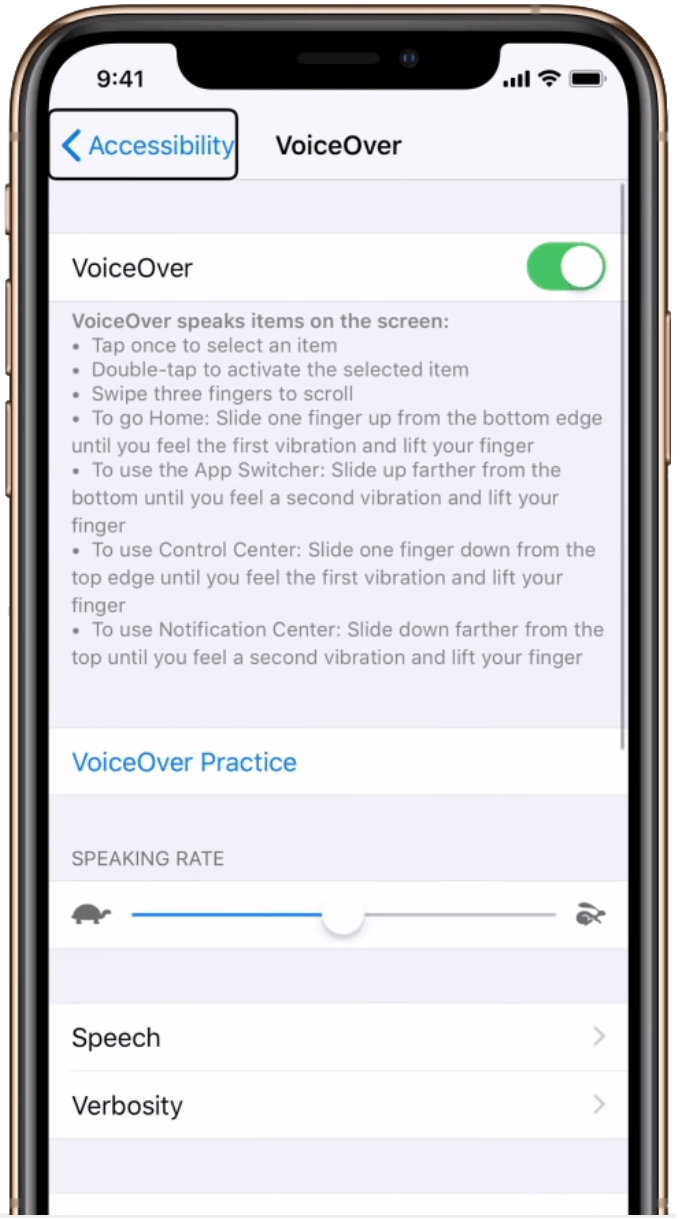
 [Photo] “How to connect the Magic Keyboard to your iPhone” by Luke Filipowicz – imore.com

Idea

Test all possible PINs, e.g., via a keyboard

Problems:

- **Keyboard navigation** -> Activate “VoiceOver”
- Rate-Limiting
- Battery



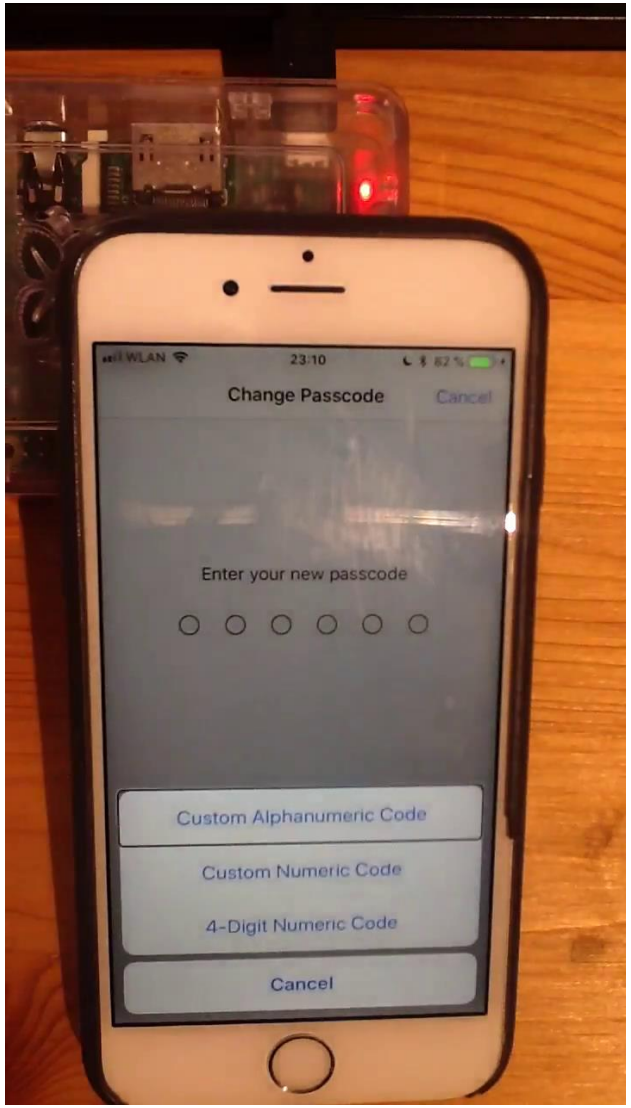
 [Photo] VoiceOver - apple.com

Idea

Test all possible PINs, e.g., via a keyboard

Problems:

- Keyboard navigation
- **Rate-Limiting**
- Battery



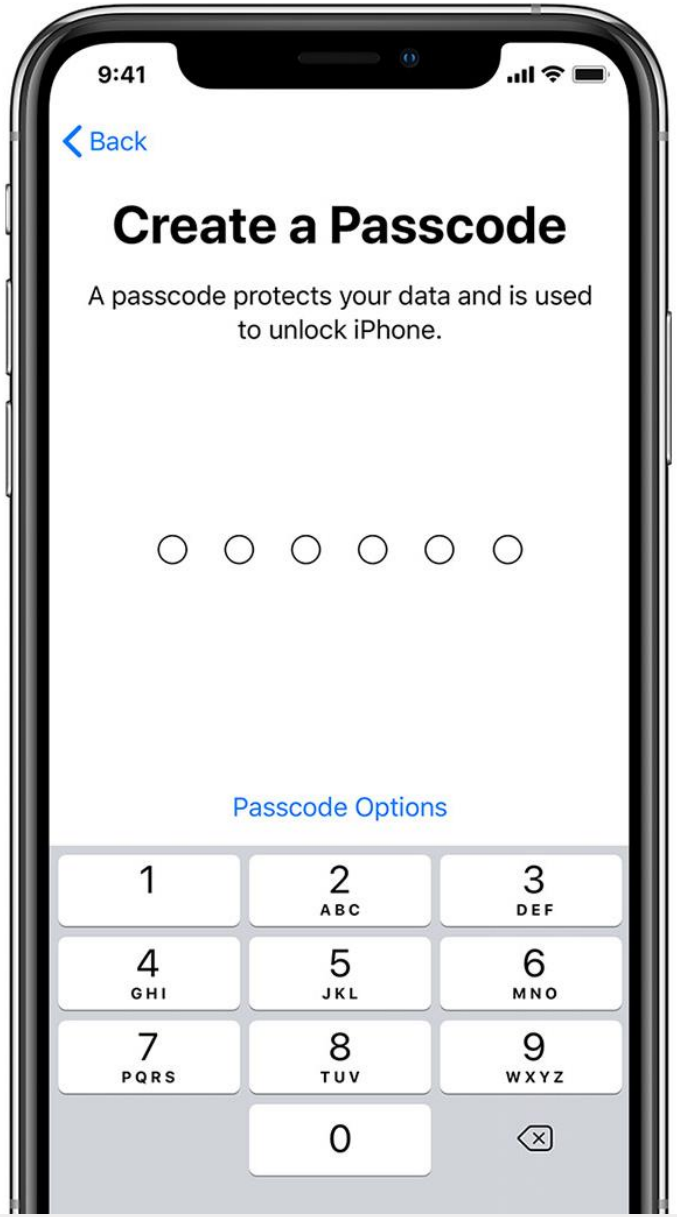
Bluetooth Prototype

Idea

Test all possible PINs, e.g., via a keyboard

Problems:

- Keyboard navigation
- **Rate-Limiting** -> Exploit initial setup
- Battery



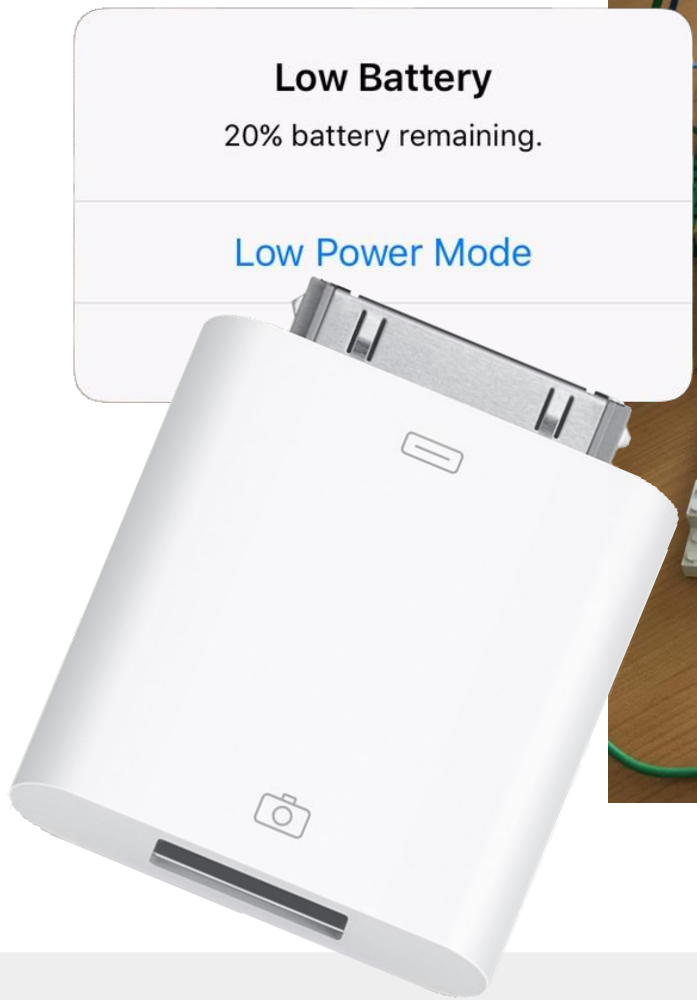
 [Photo] Create a Passcode - apple.com

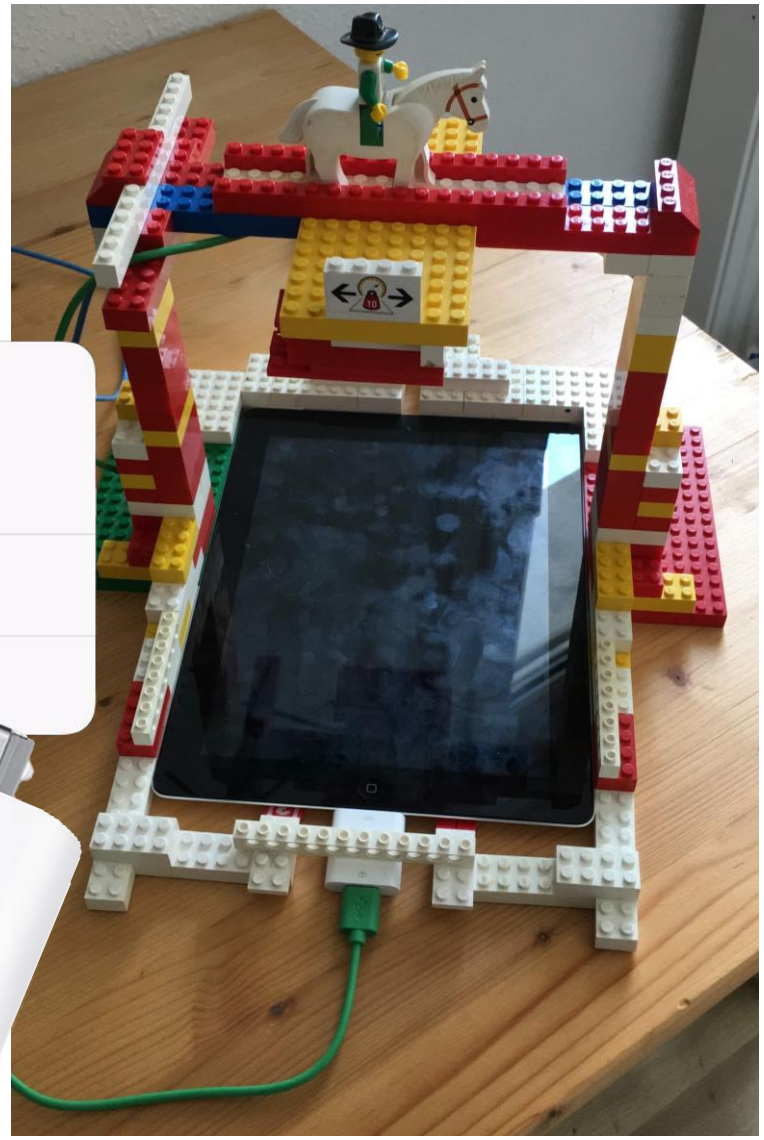
Idea

Test all possible PINs, e.g., via a keyboard

Problems:

- Keyboard navigation
- Rate-Limiting
- **Battery**





iPad 2 Prototype

 [USB-Adapter] apple.com

 [Tweet] https://twitter.com/dbreunig/status/792034409788518401

Idea

Test all possible PINs, e.g., via a keyboard

Problems:

- Keyboard navigation
- Rate-Limiting
- **Battery** -> Use “Lightning to USB 3 Camera Adapter”

[USB-Adapter] apple.com

Enter PIN + Take a Photo (Raspi Part)

Bash

```
for pin in pins:
    for digit in pin:                    # Enter a PIN
        send_key(digit)
    time.sleep()

     take_picture(camera_pid, pin) # Tell Pi to take a photo

     discard_warning()                   # Navigate down
     time.sleep()

     enter_stop_pin()                    # Intentionally fail to
     time.sleep()                        # re-enter the PIN
```

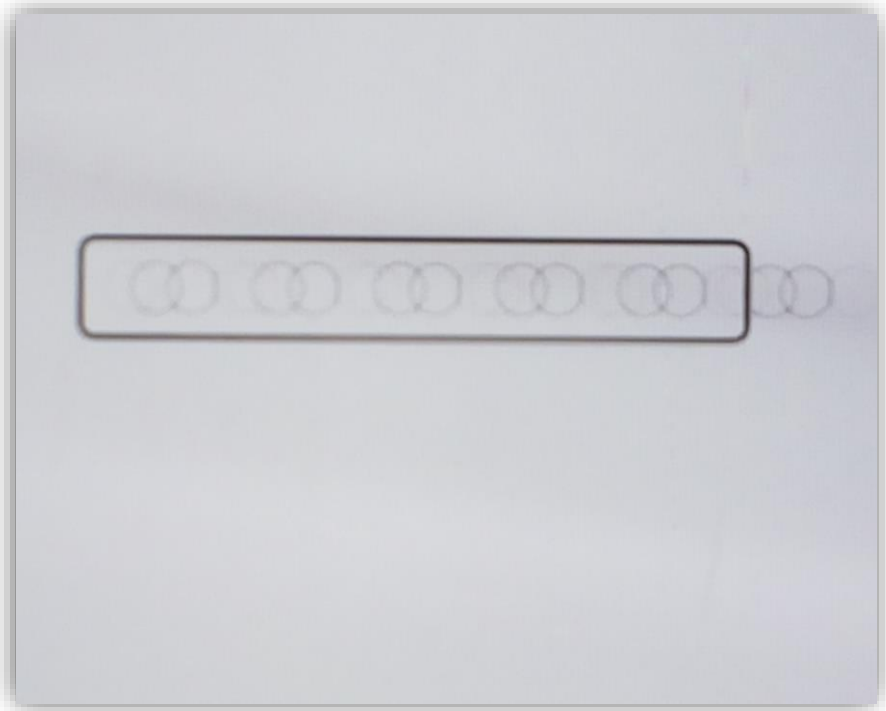

Optical Character Recognition (Server Part)

Bash

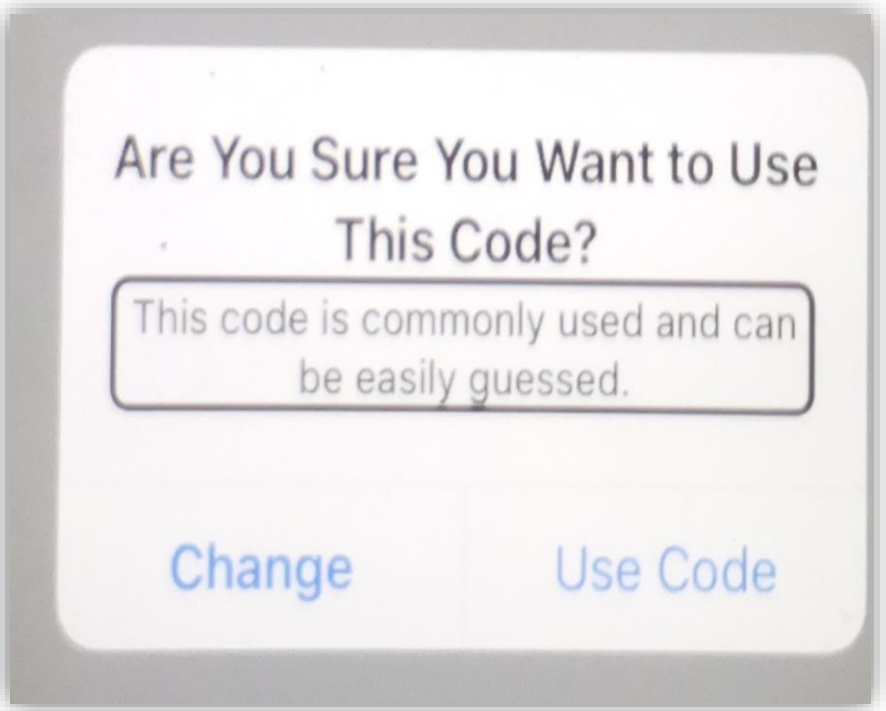
```
import pytesseract                          # Tesseract OCR Engine
import cv2                                  # OpenCV (Computer Vision)

for file in files:
    cv2.imread()                            # Read img. from disk
    cv2.threshold()                         # Convert img. to grayscale
    pytesseract.image_to_string()           # Convert img. to text

    if len(text) > 14:
        output = "PIN is blacklisted"
    else:
        output = "No warning was shown"
```

Raspberry Pi Camera Module v2 (8MP)

```
raspistill -hf -vf -roi 0.4,0.37,0.24,0.24 --width 1280 --height 1024
--nopreview --quality 25 --timeout 0 --signal -o /dev/shm/%06d.jpg
```



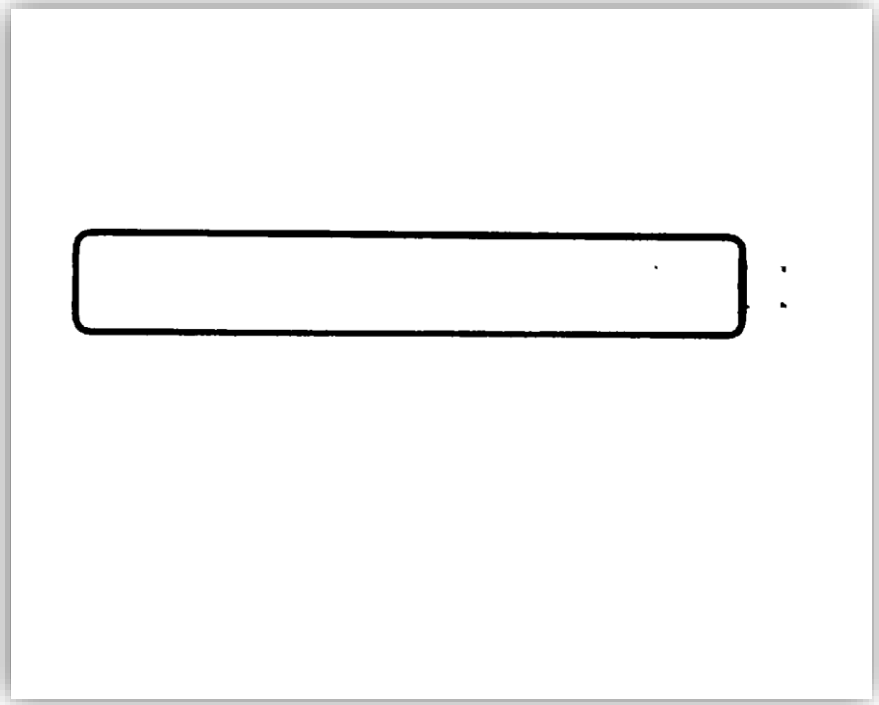
999000.jpg



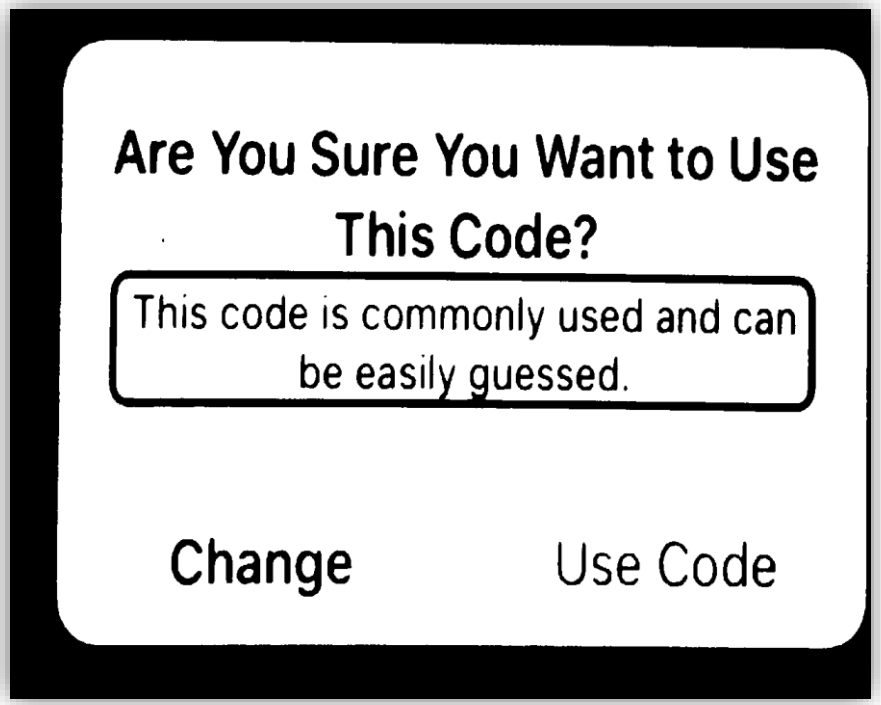
159753.jpg

OpenCV – RGB ↔ GRAY and Thresholding

```
cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY | cv2.THRESH_OTSU)[1]
```



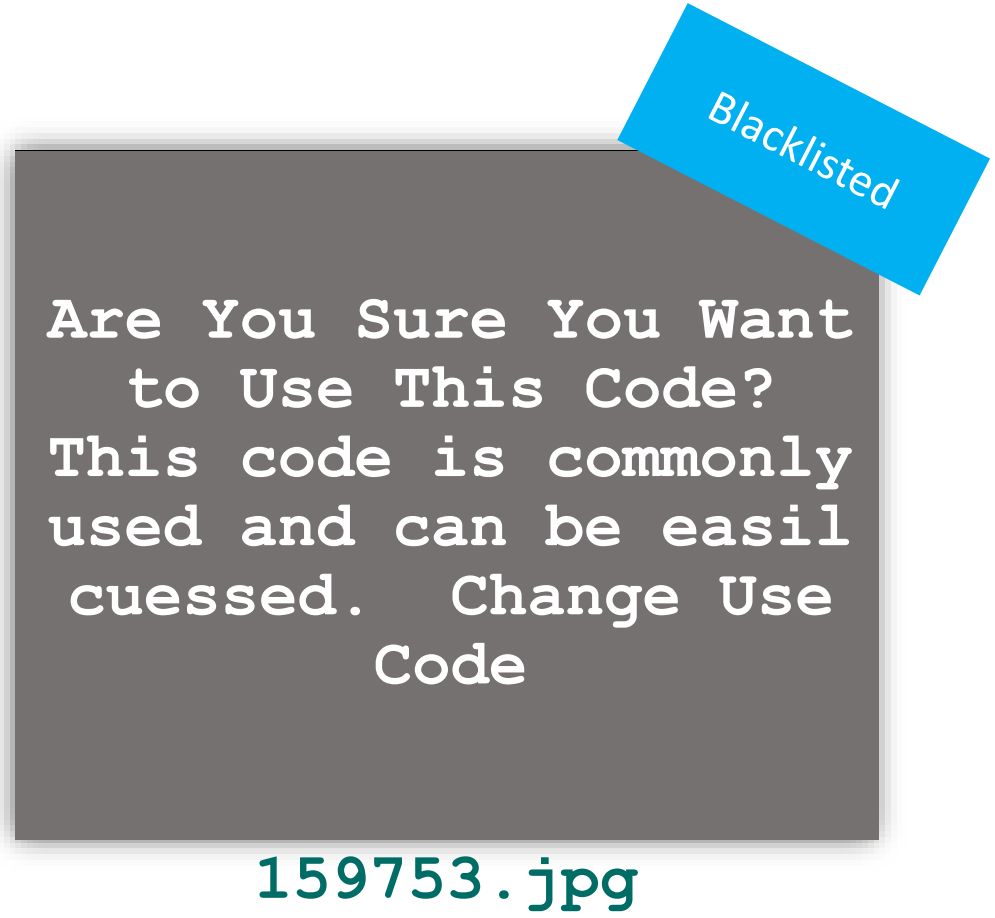
999000.jpg



159753.jpg

Tesseract Open Source OCR Engine

```
pytesseract.image_to_string(Image.open(filename))
```



The iOS Passcode Blacklist

4-digit Passcodes:

Search: ~9h (1x Setup)
Key space: 10,000 PINs

Blacklisted: 274 PINs (2.74%)

- Common PINs: Bonneau et al. [1]
- Years: 1956-2015
- Patterns:
  - ✓ aaaa
  - ✓ abab
  - ✓ aabb

6-digit Passcodes:

Search: ~30 days (2x Setups)
Key space: 1,000,000 PINs

Blacklisted: 2910 PINs (0.29%)

- Common PINs: Wang et al. [2]
- Ascending/Descending: “543210”
- Patterns:
  - ✓ aaaaaa
  - ✓ abcabc
  - ✓ abccba

[Ref. 1] Bonneau et al.: **A Birthday Present Every Eleven Wallets?** The Security of Customer-Chosen Banking PINs. (FC ‘12)
[Ref. 2] Wang et al.: **Understanding Human-Chosen PINs:** Characteristics, Distribution and Security. (Asia CCS ‘17)

Lessons Learned

- Be prepared to reset the devices
- iOS 9.3.5 blacklist != iOS 10.3.3 blacklist
- Mute the phone, because it is called "VoiceOver"!
- We reduced the brightness to a minimum, still, 2x iPhones were harmed in the process 😭

 [Photo] https://twitter.com/hackerfantastic/status/1022142922052067331

Responsible Disclosure

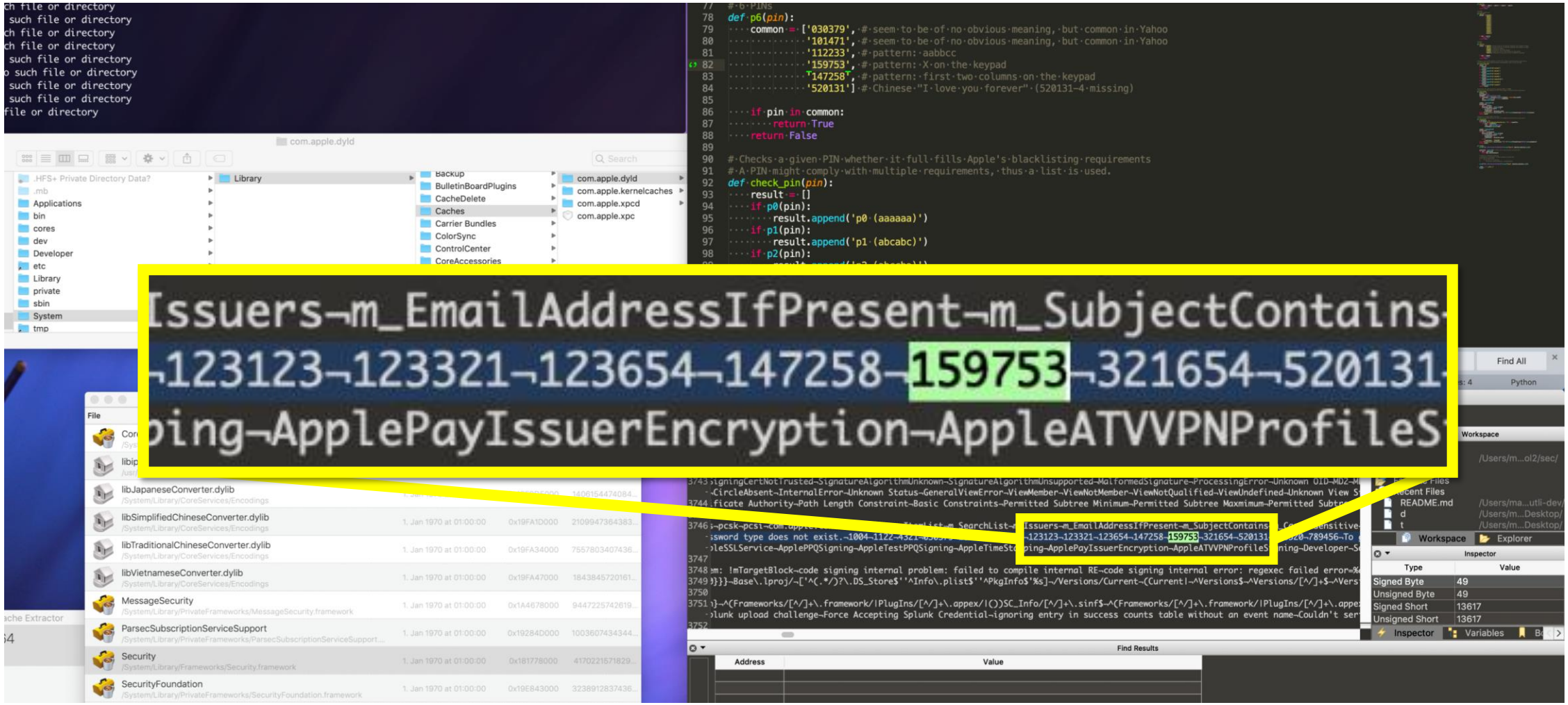
Subject: …
From: Maximilian Golla <maximilian.golla@rub.de>
Date: 04.06.19, 15:13
To: product-security@apple.com

Hello Apple Product Security,

We would like to responsibly disclose some vulnerabilities we discovered while conducting research into iOS numeric 4- and 6-digit Passcodes.

First, during the initial setup of an iOS device where a user can create a Passcode, we found that there is **no rate-limiting in place**. While the initial setup phase has no data worth protecting, this flaw allows a user to enumerate all Passcodes that are deemed not strong enough, or "blacklisted" by iOS 9-12, and we did so for 4- and 6-digit numeric Passcodes for the purposes of academic research. We enumerated the blacklist by detecting when iOS prompts the user with the warning message:

Great Robot, But Isn't There an Easier Way?



Extracting the “Common” Passcodes Directly From the IPSW

1. Download and install Malus-Security/**iExtractor** by Răzvan Deaconescu
   (A tool for macOS to automate the extraction of data from iOS firmware files.)
2. Download and decrypt the latest iOS version using iExtractor
3. Use the following code to extract the PINs directly from the dyld_shared_cache:

Bash

```
VERSION="13.3.1_17D50"
CODENAME="YukonD17D50.D22D221OS"
FILE="dyld_shared_cache_arm64" # or "dyld_shared_cache_armv7s" for iOS 7 to 10.3

hdiutil attach decrypted.dmg

strings /Volumes/$CODENAME/System/Library/Caches/$FILE | \
grep "\bSecPasswordSeparator\b" -A 120 > blacklist_iOS_$VERSION.txt

hdiutil unmount $CODENAME
```