Device Encryption

CS@CU

Device Encryption

- How are devices encrypted?
- Why are these methods strong?
- How do these methods fail?

Threat Models

- Who are your (plausible) enemies?
- What are their (plausible) powers?
- What do they want?

Attack Surfaces

- What are the places an attacker can reach?
- How weak are those places?
- Note: both answers depend on your threat model

The Door to the Computer Science Building

How would you break in? What are the defenses?

A View from the Inside

An Inner Door

Other Defenses

Keys

- If the attacker has the key and access to the file, they can decrypt it
- Assumption: law enforcement (or Customs) will always have access to the file; the “defense” here must be legal
- **The question: can the key be protected?**
  - Related: where do the keys come from? Where, if anywhere, are they stored?

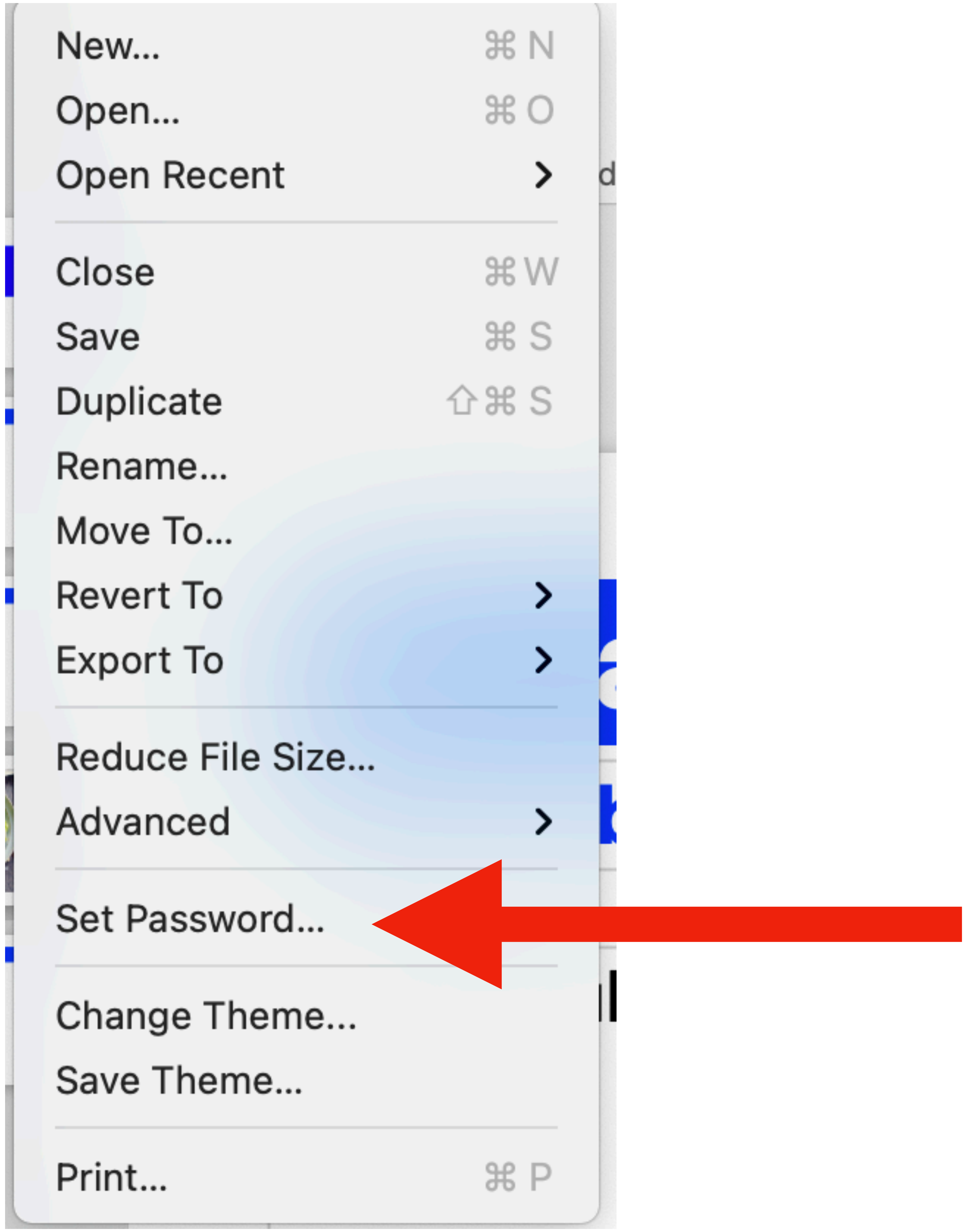
CS@CU

Encryption Types

- Manual encryption
- Disk encryption
- File system encryption
- Data class encryption

Manual Encryption

- Specify a key when saving or opening a file
- Could be inconvenient—must do it every time
- Every file utility must support encryption and decryption
- Attack surface: the entire OS, if running
- There are human and technical objections



What’s a File System?

- A disk is like a giant book of *sectors* (rather than pages)
- A *file system* is a way of organizing the sectors, to let you find the information you want—something like a table of contents, but instead of chapters, sections, subsections, etc., we have folders (AKA *directories*)
  - The sectors in a file need not be in order—the file system says where they all are
- We also have a supply of unused sectors called the *free list*
  - When a file is deleted, its table-of-contents entry is cleared, but the the sectors typically are not erased
  - Forensic analysis tools can recover the contents of these sectors!

Disk Encryption

- Encrypt every sector, whether in use or part of the free list
- Minimal attack surface if the key isn't supplied
  - Use indirection: encrypt the disk with a random key; encrypt that key with a password
  - To permanently destroy the whole disk, just destroy that encrypted key—that's fast!
- But—protection is all-or-nothing; the same key protects the entire disk
- To boot the system, the key must (somehow!) be supplied
  - (Could a key have been left lying around after the last boot?)

CS@CU

File System Encryption

- Each file is encrypted separately
- You can have different keys for different files or directories full of files
- These keys can be supplied at different times, or as needed
- Good granularity of protection—but there can be a much larger attack surface

iOS Security

- The CPU contains a *secure enclave*—a way to run selected pieces of code completely isolated from the rest of the system
  - Has separate memory not accessible from the rest of the system
  - In theory, minimal attack surface for the code in the secure enclave
- Specific iOS functions are executed in the enclave
  - Examples: data and keychain encryption, PIN entry, biometric authentication
- Contains a random *device UID key* generated at first power-on or at hard reset

Data Protection Classes

- iOS permits different classes of data to be encrypted differently
  - In other words, encrypted with different keys
- But: it's up to application programmers to select the proper storage class for different types of data
- Attack surface: whatever part of the OS is able to run

Data Protection Classes on iOS

No protection: Encrypted with device UID key only

After First Unlock: Protected at boot time; key available after first PIN-based unlock

Protected Unless Open: When a file is closed, its key is “evicted”; data cannot be decrypted after that

Complete Protection: Key evicted when the device is locked

How It's Done

- The device UID key is used to create the No Protection key, via a hash
- The other keys are created by hashing together the UID key and the user-supplied PIN
- A *tag* for each class is also hashed in

Why It's Secure

Powered off: Disk is encrypted; UID key is in the secure enclave only

If the UID key is changed, the disk is effectively erased

Powered Up: The UID key is available—can produce key to decrypt many phone functions

Large attack surface—but what of the PIN?

After First Unlock: PIN has been entered; secure enclave can calculate other protection class keys

Protected Unless Open; Complete Protection: The secure enclave decides when to restore the key to iOS

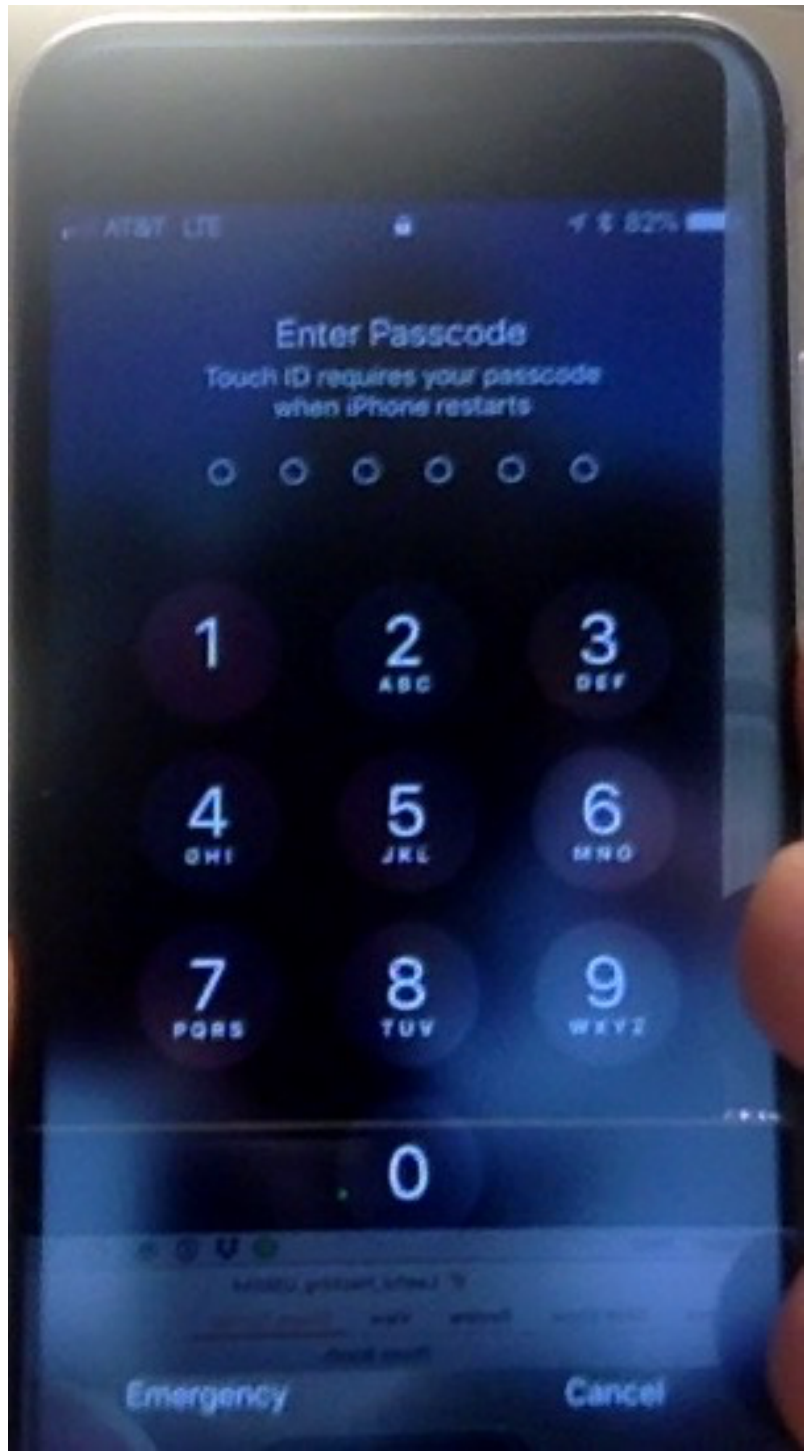
Generating PIN-Protected Keys

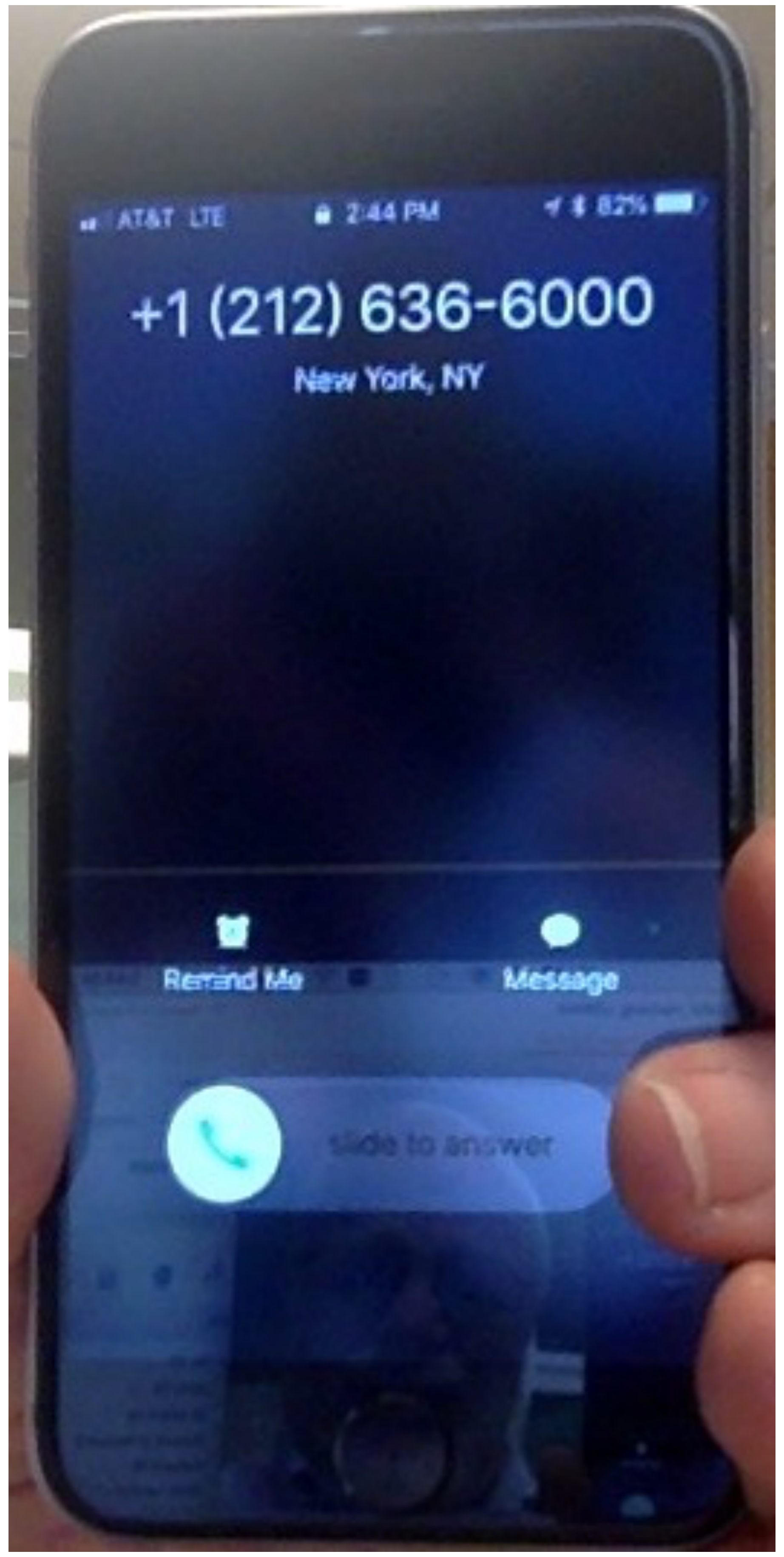
- Can only be done by the secure enclave—nothing else has access to the UID key
- The key generation process is inherently slow: 80 ms per guess
  - There are 1,000,000 possible six-digit PINs—will take 80,000 seconds to try all possible PINs
  - If lower-case letters and digits are used: 5.5 years
- The secure enclave, by default, wipes the phone after 10 failed guesses

CS@CU

iPhone Attack Surface

It can answer calls before a PIN is entered…





Attacking iOS Security: What Works

- iOS has a large attack surface while running, both before and especially after first unlock
- Many companies have found security flaws in iOS that allow them in
    - They market forensic software to law enforcement
- Recall: after first unlock, most keys are present in RAM, and the secure enclave knows the the rest…

CS@CU

Before First Unlock?

- In theory, there are no attacks possible before first unlock—the PIN and the device UID key are necessary, only the secure enclave can access the device UID, and it limits guesses
- Are there security holes in the secure enclave API? Unclear…
    - Holes to allow grabbing the UID?
    - Holes to reset the guess counter or up the limit?

What No Longer Works

- Insert a boobytrapped USB device
  - iPhones won’t talk to newly inserted USB devices more than a short while after the phone is locked
- Replace the firmware with an image that reveals the device UID
  - Installing new firmware requires knowledge of the PIN—Apple locked themselves out of the ability to create nasty images

Android Phones

- There's no one Android
  - Google supplies a base system, but each vendor customizes it
  - Apple controls the hardware and the software for iOS; Google does not
  - Some vendors with high-end hardware implement encryption; low-end platforms do not
  - Some Android phones have a weaker analog to Apple's Secure Enclave

Android Encryption

- Two types, full disk encryption and file encryption
- Full disk: must supply PIN at boot time for the phone to do anything
- File encryption: two types
  - Device-encrypted: always available after boot; similar to Apple's scheme. Used for system applications
  - Credential-encrypted: similar to Apple's AFU encryption
    - Keys are never evicted from memory after boot; no equivalent to iOS Complete Protection

CS@CU

Going Around Encryption

- “You don’t go through strong security, you go around it”
- Smart guesses at the PIN
- Faking the biometrics
- The cloud

Guessing PINs

- Your fingers leave oily marks on the screen when you touch it
  - Is there a pattern around the PIN digits?
  - Even better with Android's pattern unlock: look for smears
- Thermal sensing immediately after unlock
- Ubiquitous surveillance cameras
  - Given mask-wearing, face-unlock doesn't work that well anymore; people have to use their PIN…

Faking—or Forcing—the Biometrics

- Creation of fake fingerprints has been demonstrated several times
  - Some sensors have been buggy and accept almost anything…
  - Better sensor designs incorporate *liveness detectors*, but sometimes those can be fooled
- Apple's facial recognition appears to be quite strong—it looks for facial features, open eyes, a face that's close enough, etc.
- Phones typically disable biometric authentication periodically, or after a few failed tries

Cloud Backups

- Many phones and apps back up their data to some form of cloud storage
- Apple explicitly makes some of this available via iCloud
- App stores know what you've installed
- Most email services use IMAP, where the primary copy of all messages is on the server
- Google collects *lots* of data about people, from Android phones and from Google apps

Does the party that seized the device have the ability to compel cooperation from cloud services?

Laptops

- All modern laptop operating systems support full disk encryption
  - MacOS FileVault; Windows BitLocker
- What about backup keys (AKA recovery keys)?
  - Apple: print out a recovery key and/or enable unlock via AppleID password
  - Microsoft: print out a recovery key; cloud key backups are also used
- Is the owner’s backup disk encrypted?

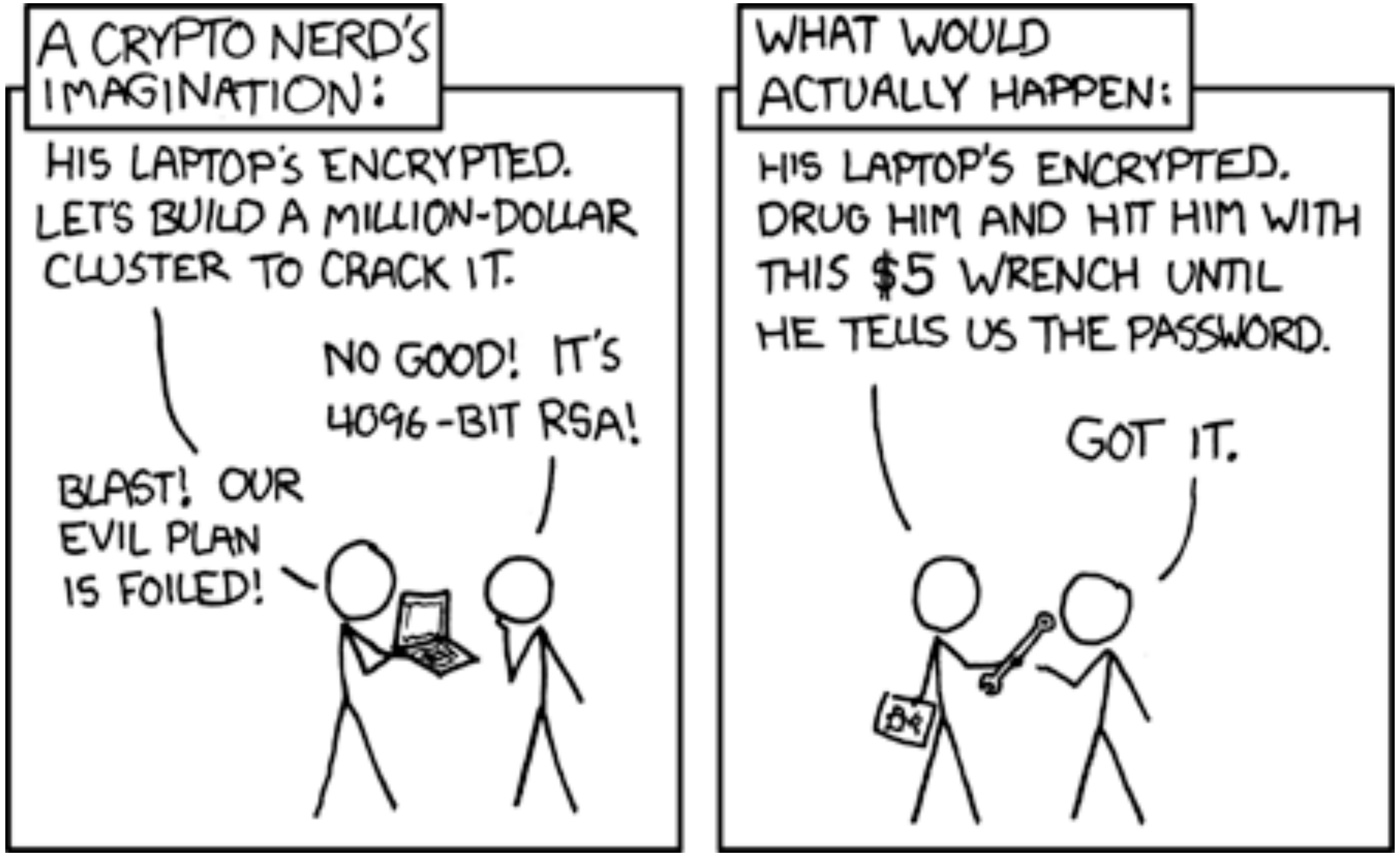
Device Encryption

- Most devices are pretty secure if they’re powered off when seized
- Much of the data may be available via cloud services
- There are often ways around the encryption

Outside the Usual Threat Model

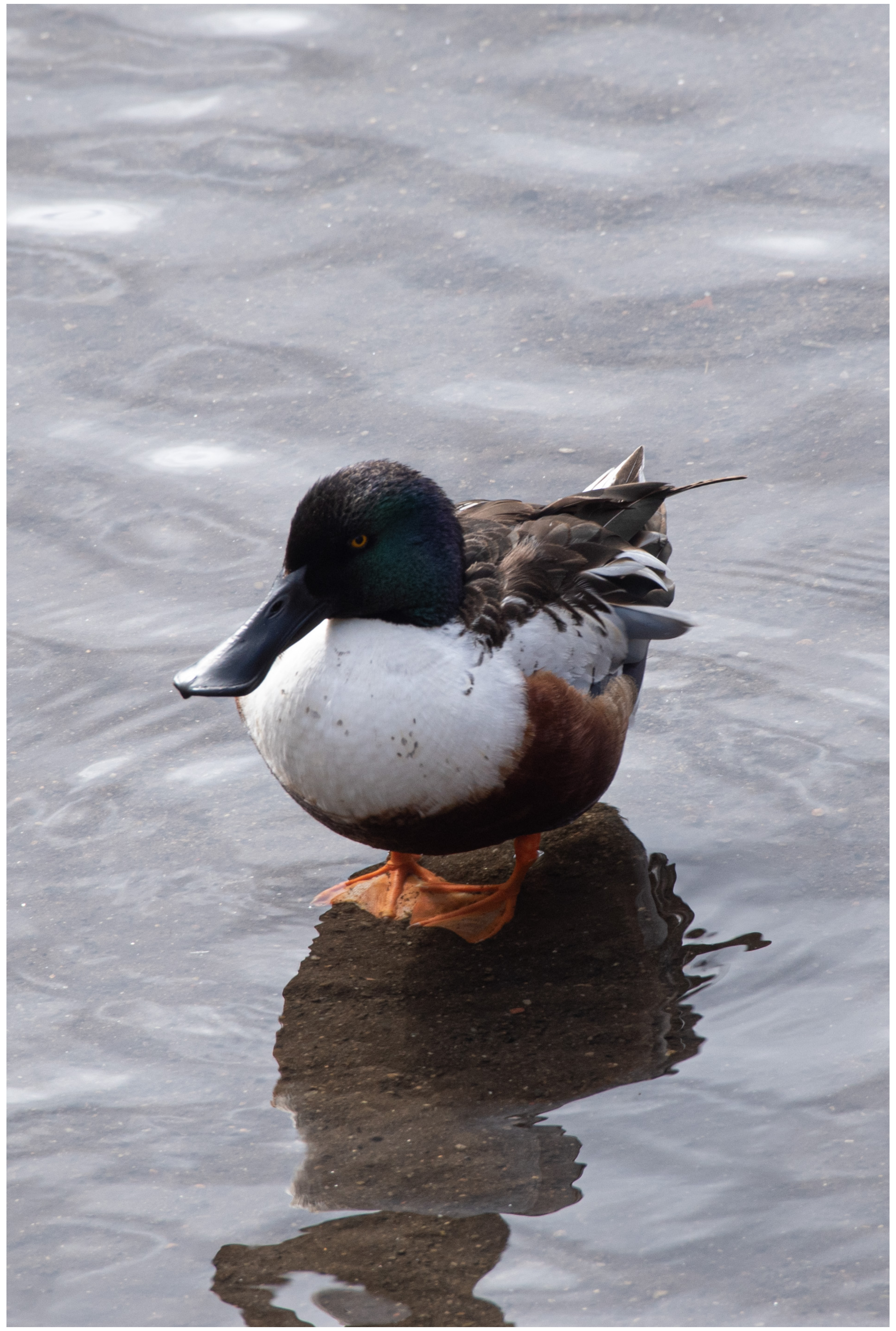
- “Decap” the chip and read out the device UID directly
  - This will let you speed up the 80 ms per guess rate
- Unsolder the SSD, back it up, and try guesses—when you hit the limit, restore the SSD from the backup

A Threat Model in Some Places…



https://xkcd.com/538/

Bird of the Day



Northern shoveler, Central Park, March 6, 2021

CS@CU