Comp 590-184: Hardware Security and Side-Channels

Lecture 14: Hardware Security Modules continued

March 5, 2026
Andrew Kwong

THE UNIVERSITY *of* NORTH CAROLINA *at* CHAPEL HILL

Slides adapted from UCSD CSE 127

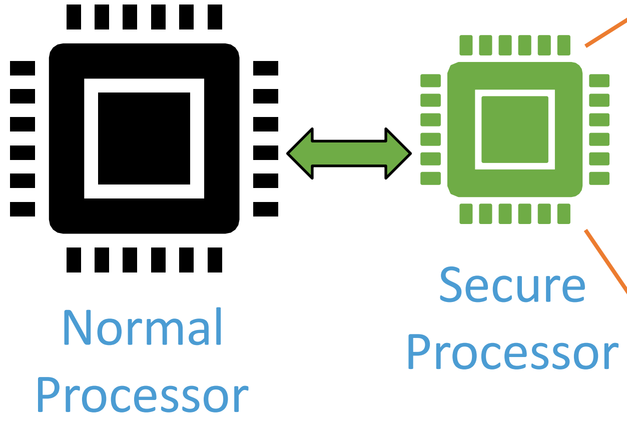
Hardware Security Modules

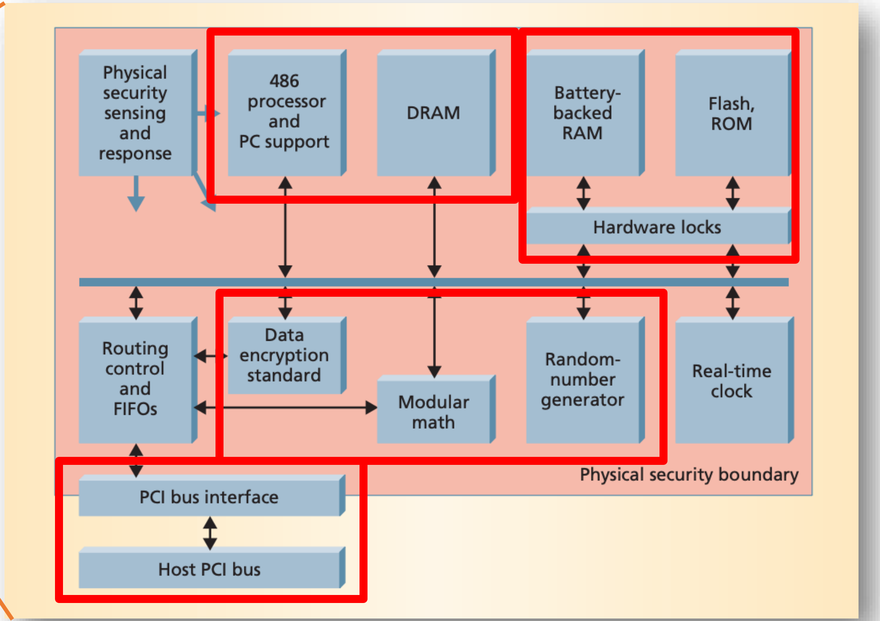
- Hardware can offer security primitives we cannot achieve with only software

Secure Co-Processors

General-purpose processor, rather than ASIC, with isolated DRAM.

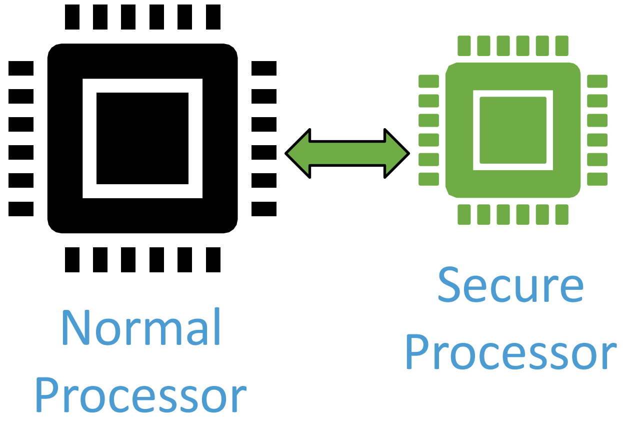
Hardware lock, resilient against physical attacks to modify firmware





Narrow interface, only interact with external worlds via APIs
(keys do not leave the co-processor)

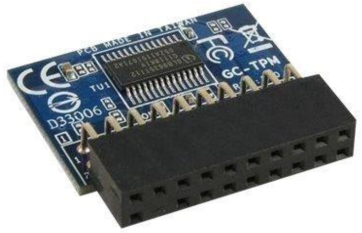
Secure Co-Processors

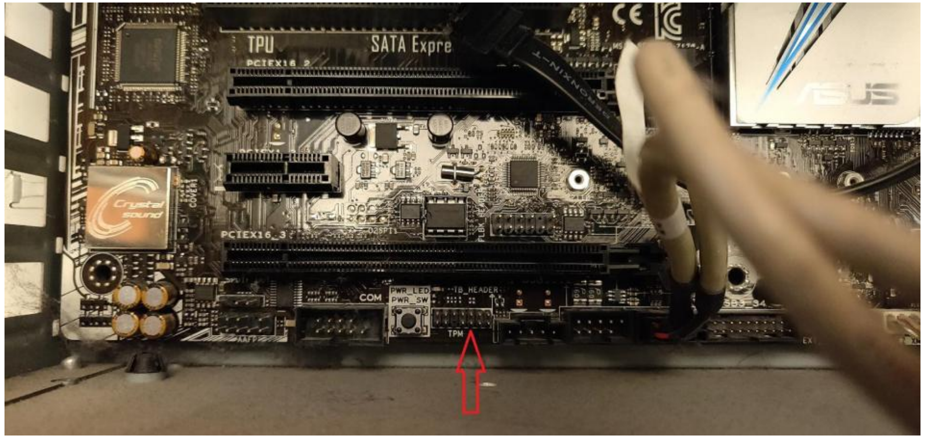


- Before IBM 4758 (1999):
  - Crypto accelerators (AES, RSA, etc.)
  - Store crypto keys inside the accelerator
  - Want to run more applications on the co-processor
- IBM 4758 (1999) -- 4765 (2012)
  - **Programmable** secure co-processor
  - Idea: create a virtual locker room
  - Problem?
  - The SOFTWARE! Bad programmability.
  - Need to find a middle ground: run selected applications that offer strong security functionality

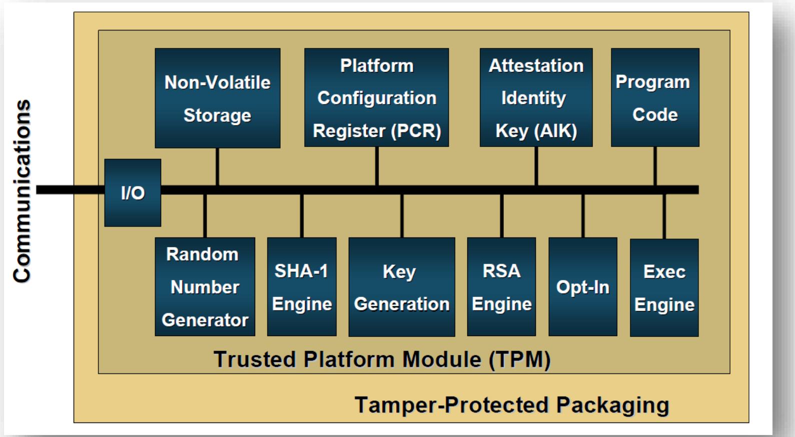
Trusted Platform Module (TPM)

- “*Commoditized* IBM 4758”: Standard LPC interface attaches to commodity motherboards



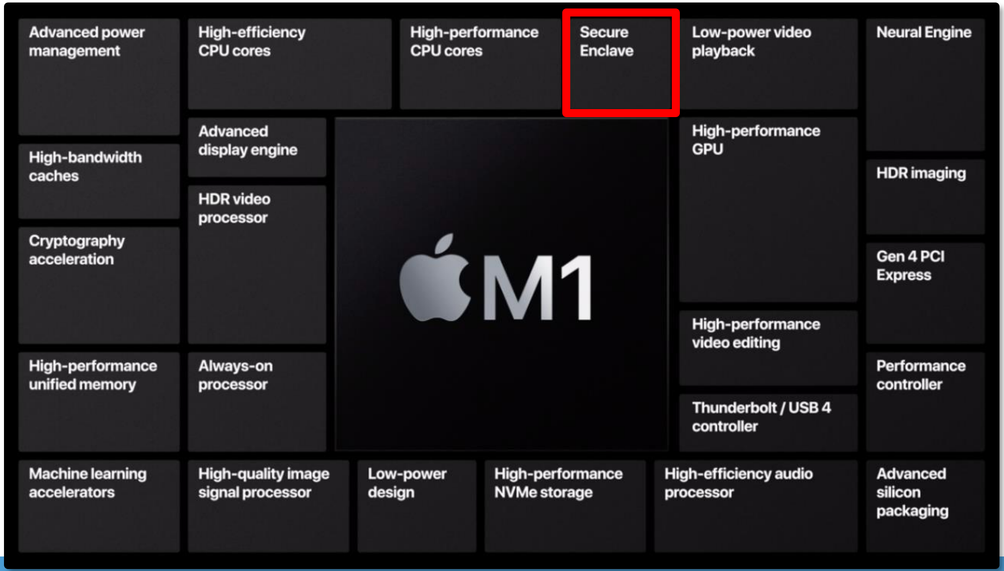


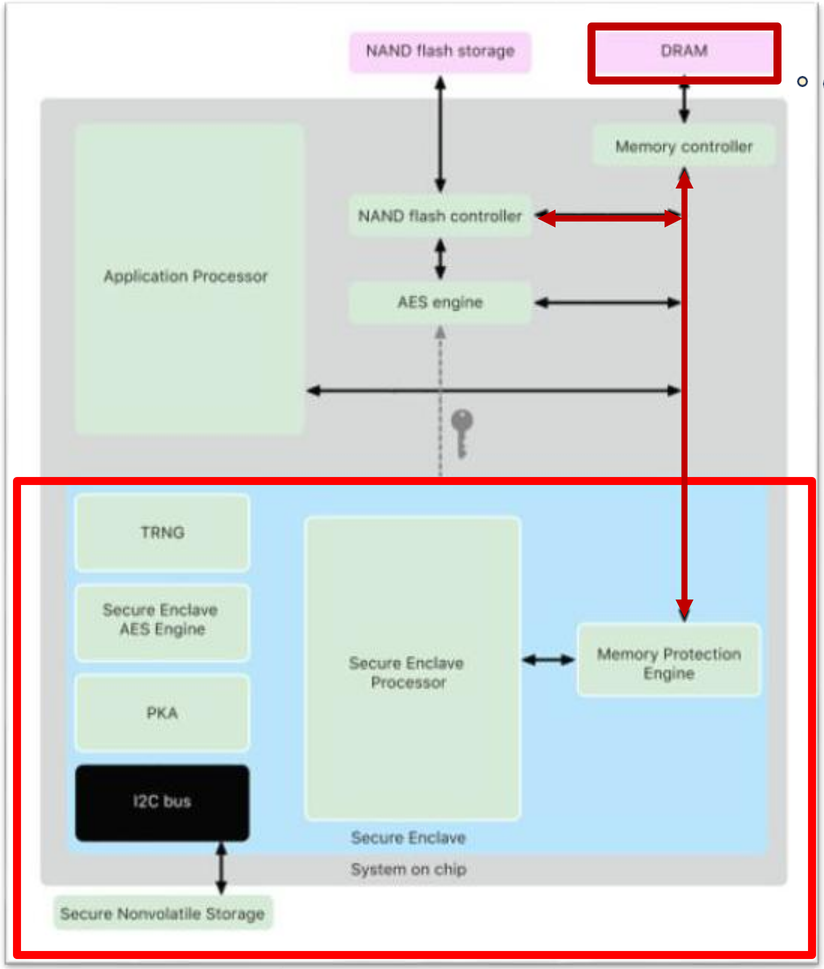
https://scotthelme.co.uk/upgrading-my-pc-with-a-tpm/



Apple Secure Enclave

- Advantage: one company controls both the hardware and the software
- Apple secure enclave runs a customized formally verified micro-kernel OS


Advanced power management
High-efficiency CPU cores
High-performance CPU cores
Secure Enclave
Low-power video playback
Neural Engine
High-bandwidth caches
Advanced display engine
High-performance GPU
HDR imaging
HDR video processor
Cryptography acceleration
M1
Gen 4 PCI Express
High-performance video editing
High-performance unified memory
Always-on processor
Performance controller
Thunderbolt / USB 4 controller
Machine learning accelerators
High-quality image signal processor
Low-power design
High-performance NVMe storage
High-efficiency audio processor
Advanced silicon packaging

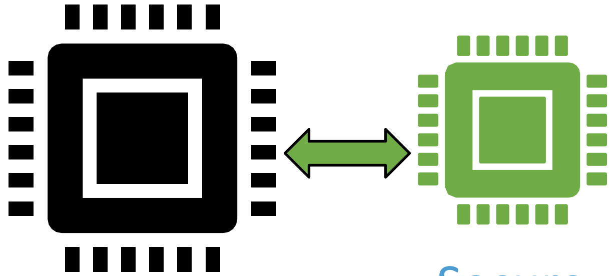


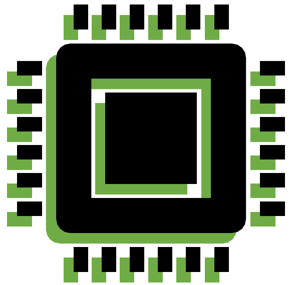
Encrypt enclave data and only decrypt at the memory protection engine

- Only run secure enclave functionality, no user code
- Block vulnerabilities due to software bugs (running L4 microkernel)
- Block uarch side channels

From Apple Platform Security (Page 5-96)

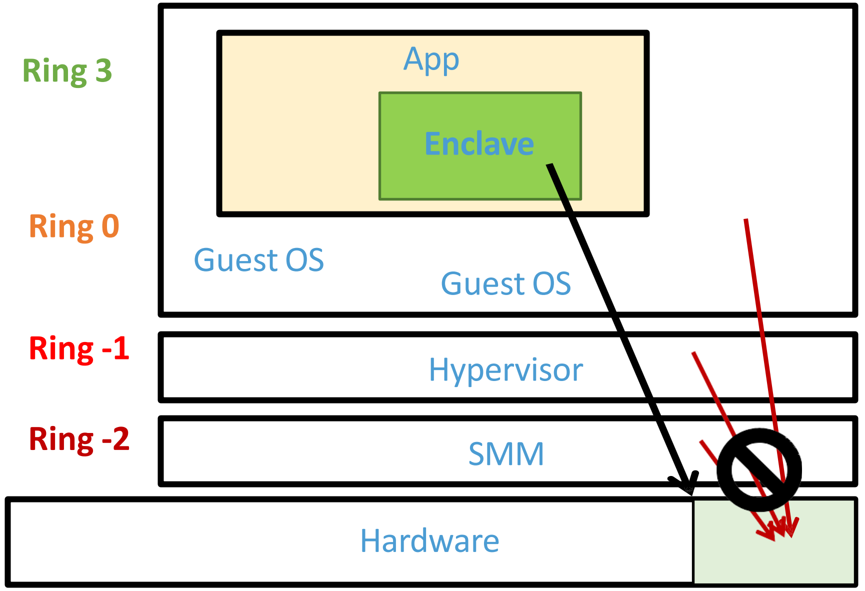
Make Physical Isolation More Flexible?



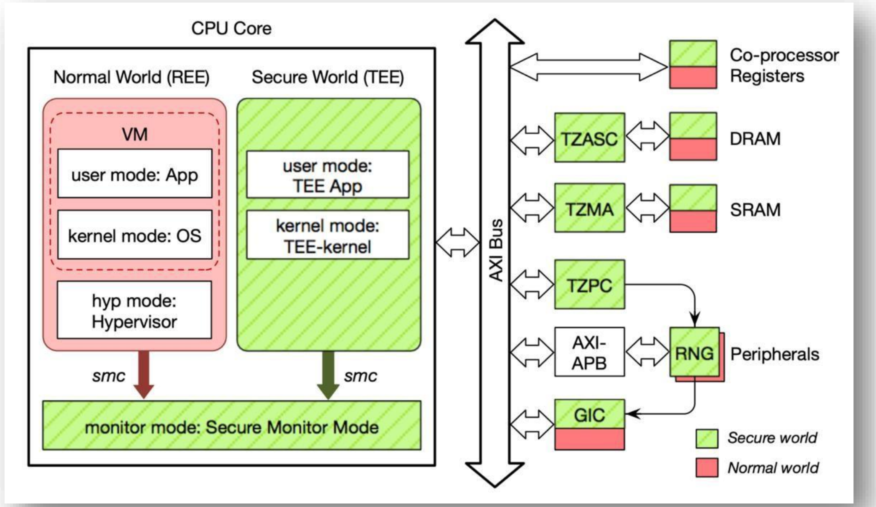


A Processor can switch between Normal and Secure Modes

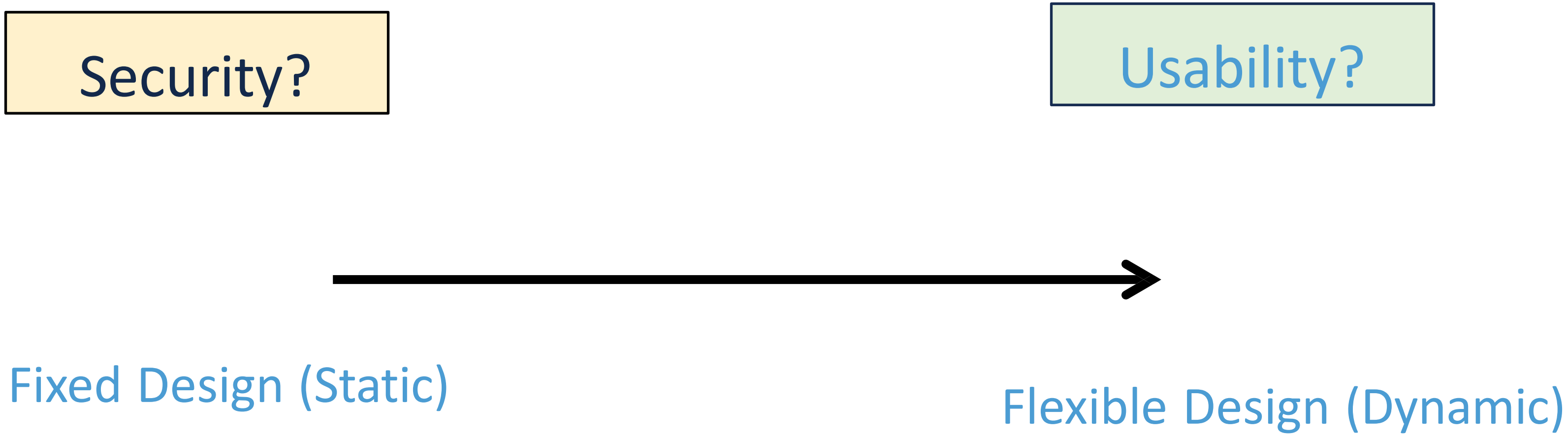
The Trends (isolation with some sharing?)



Intel SGX model



ARM TrustZone


Security?
Usability?
Fixed Design (Static)
Flexible Design (Dynamic)

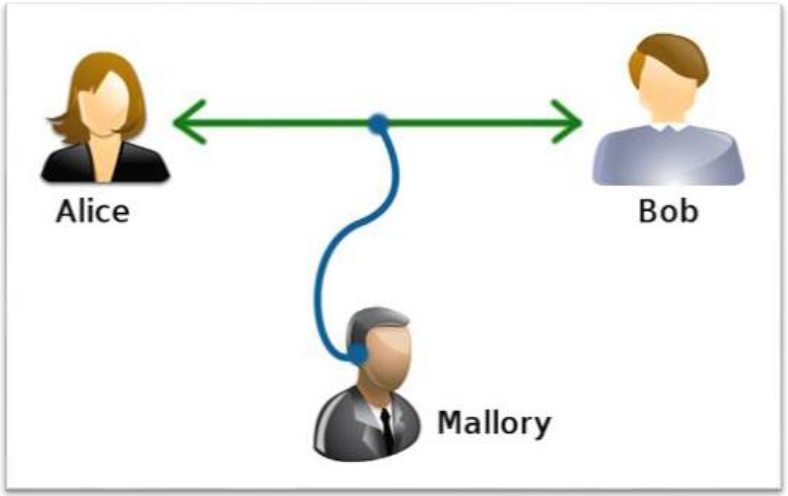
Security Context #2

Lost your device?

- Data leakage => confidentiality
- Rootkits => integrity

Security Properties and Crypto Primitives

- Confidentiality
  - Symmetric
  - Asymmetric
- Integrity
- Freshness



Cryptography

- Is:
  - ➢ A tremendous tool
  - ➢ The basis for many security mechanisms
- Is not:
  - ➢ Reliable unless implemented and used properly
  - ➢ Something you should try to invent yourself
  - ➢ Another word for blockchain/bitcoin

- Do not roll your own crypto*

  * Exceptions: You are Daniel J. Bernstein, Joan Daemen, Neal Koblitz, Dan Boneh, or similar, or you have finished your PhD in cryptography under an advisor of that caliber, and your work has been accepted at Crypto, Eurocrypt, Asiacrypt, FSE, or PKC and/or NIST is running another competition, and then wait several years for full standardization and community vetting.

Terms

- **cryptosystem**: method of disguising (encrypting) plaintext messages so that only select parties can decipher (decrypt) the ciphertext
- **cryptography**: the art/science of developing and using cryptosystems
- **cryptanalysis**: the art/science of breaking cryptosystems
- **cryptology**: the combined study of cryptography and cryptanalysis

Types

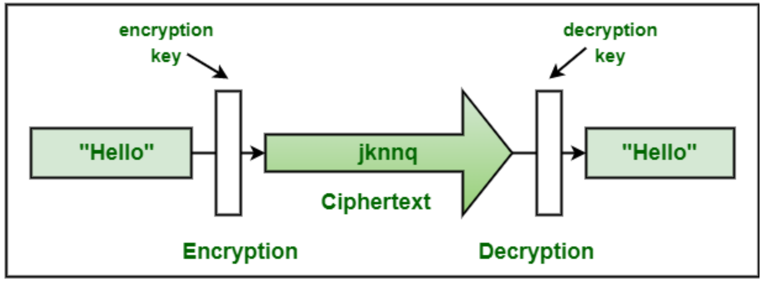
- Symmetric cryptography
  - Encryption and decryption keys are the same
- Asymmetric cryptography
  - Encryption and decryption keys differ!
- We'll start with symmetric key

Cryptosystem

- A cryptosystem is a 6-tuple consisting of

  (E, D, G, M, C, K)

- Where,
  - **M** is the set of *plaintexts*
  - **C** is the set of *ciphertexts*
  - **E** is an *encryption* algorithm
  - **D** is a *decryption* algorithm
  - **K** is the set of *keys*
  - **G** is a *key generation* algorithm



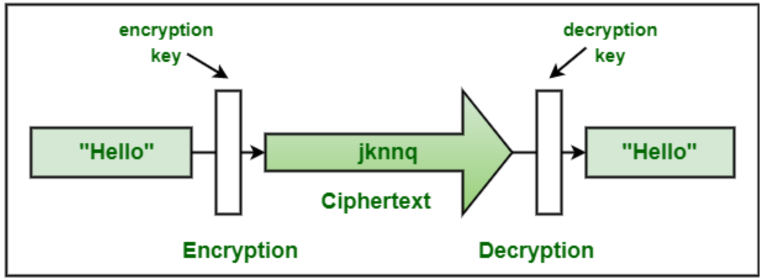
Caesar Cipher

- Shift each letter in the alphabet by a fixed mount
- Key is the shift

```
Plaintext:  THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
Ciphertext: QEB NRFZH YOLTK CLU GRJMP LSBO QEB IXWV ALD
```

What is secure encryption?

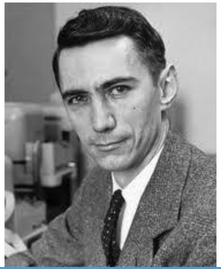
- Should be
  - Impossible for attacker to recover plaintext from ciphertext
  - Impossible for attacker to recover any character of the plaintext from ciphertext
  - Impossible for attacker to recover the key
  - Developed such that ciphertext leaks no additional info about plaintext regardless of info attacker already has



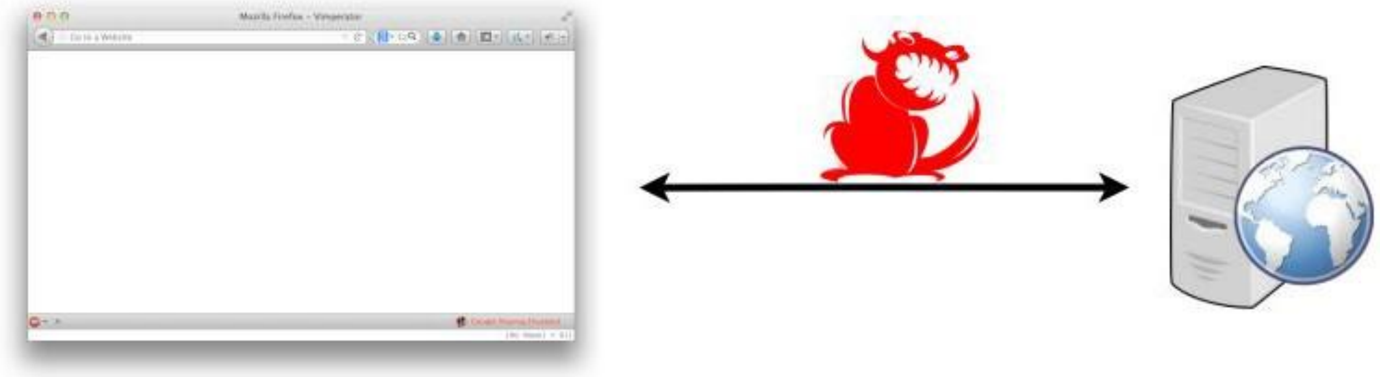
Kerckhoff’s Principle

“The cipher method must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience”

“The enemy knows the system”



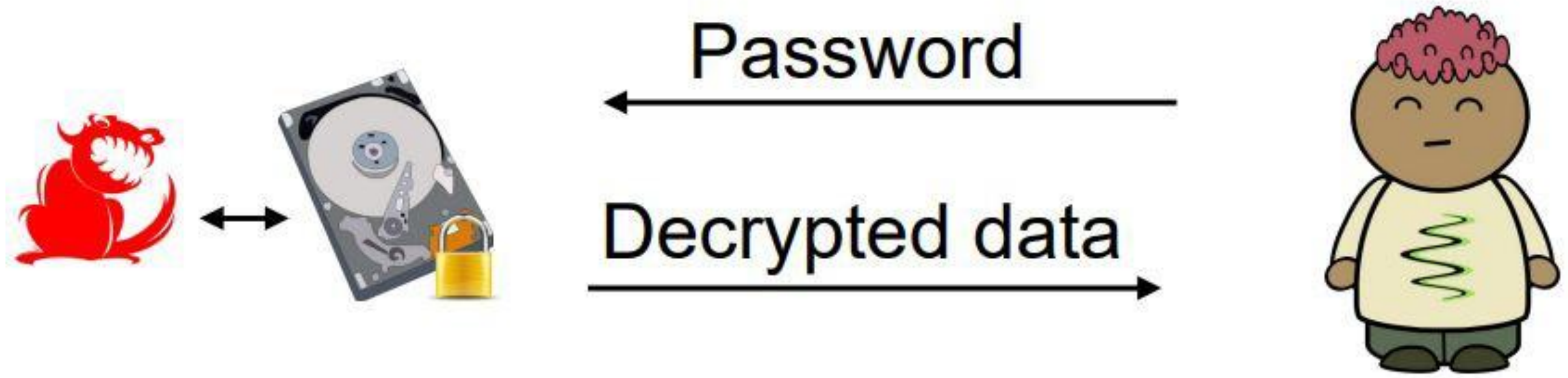
Real-world crypto: SSL/TLS



1. **Browser and web server run "handshake protocol":**
   - ➤ Establishes shared secret key using public-key cryptography

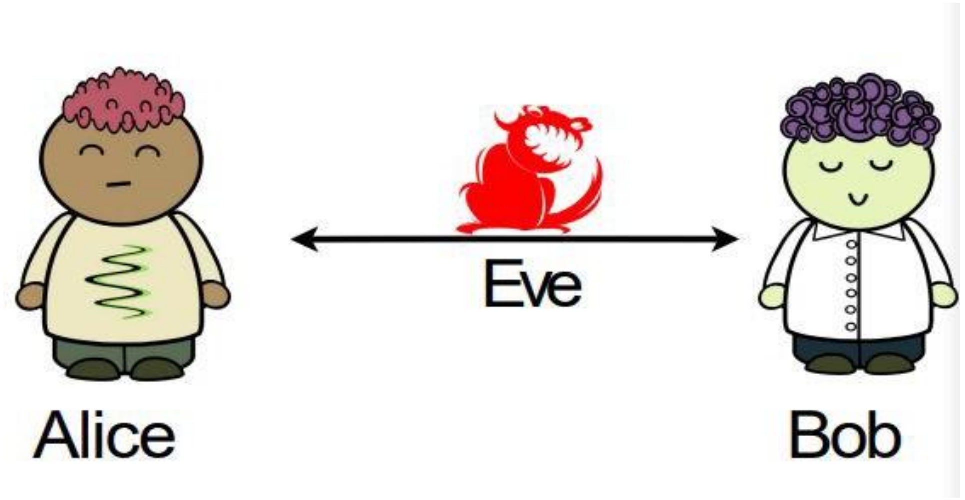
Browser and web server use negotiated key to encrypt all traffic

Real-world crypto: File encryption



- Files are encrypted with a secret key
- The key is stored encrypted or in tamperproof hardware.
- The password is used to unlock the key so the data can be decrypted.

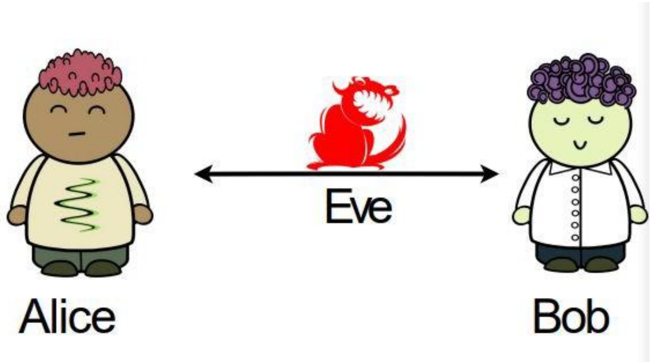
secure communication



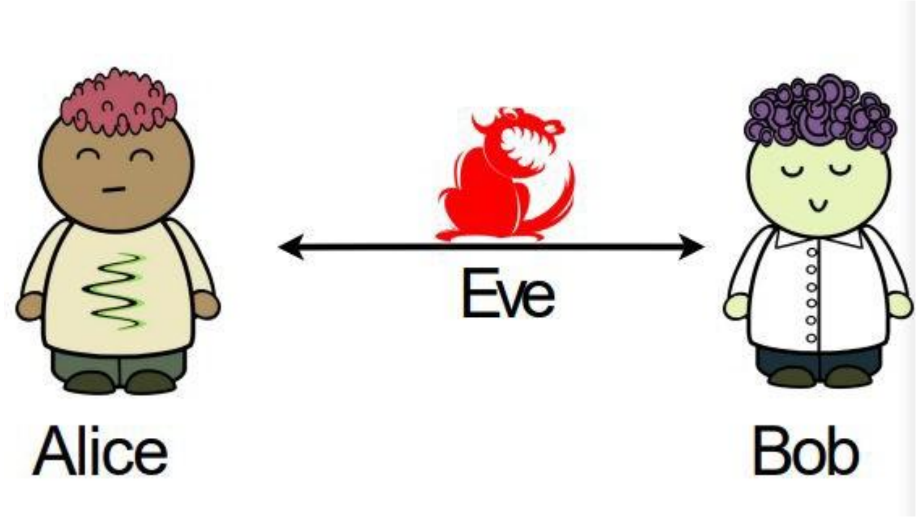
- Authenticity: Parties cannot be impersonated
- Secrecy: No one else can read messages

secure communication

- **Confidentiality**
  - Keep data and communication secret
  - Encryption / decryption
- **Integrity**
  - Protect reliability of data against tampering
  - “Was this the original message that was sent?”
- **Authenticity**
  - Provide evidence that data/messages are from their purported originators
  - “Did Alice really send this message?”



Attacker models

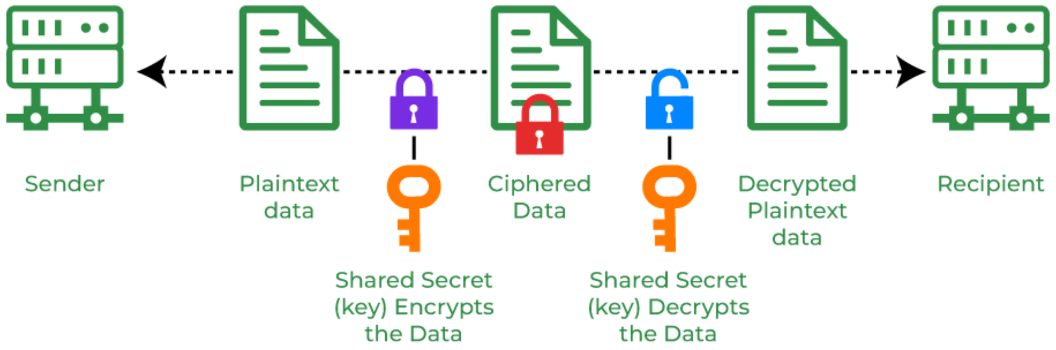


Passive attacker: Eve only snoops on channel

Active attacker: Eve can snoop, inject, block, tamper, etc.

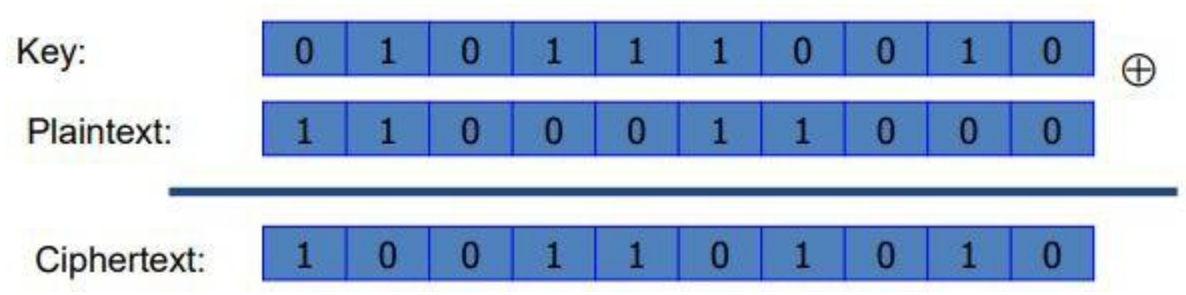
Symmetric Cryptography

- Encryption key and decryption key are the same


Sender
Plaintext data
Shared Secret (key) Encrypts the Data
Ciphered Data
Shared Secret (key) Decrypts the Data
Decrypted Plaintext data
Recipient

Example: One Time Pad

Vernam (1917)

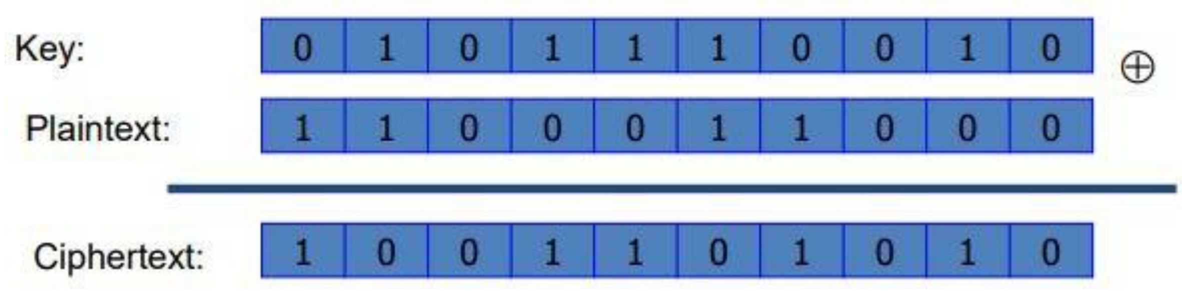


Key:	0	1	0	1	1	1	0	0	1	0
Plaintext:	1	1	0	0	0	1	1	0	0	0
Ciphertext:	1	0	0	1	1	0	1	0	1	0

- **Encryption:**
- **Decryption:**

Example: One Time Pad

Vernam (1917)



Key:	0	1	0	1	1	1	0	0	1	0	$\oplus$
Plaintext:	1	1	0	0	0	1	1	0	0	0	
Ciphertext:	1	0	0	1	1	0	1	0	1	0	

- **Encryption:** $c = E_k(m) = m \oplus k$
- **Decryption:** $D_k(c) = c \oplus k = (m \oplus k) \oplus k = m$

OTP security

- Shannon (1949)
  - ➤ Information-theoretic security: without key, ciphertext reveals no “information” about plaintext
- Problems with OTP
  - Can only use key once
  - Key is as long as the message
  - No integrity protection

Confidentiality

- *Unconditional security* : cryptosystem offers provable guarantees, irrespective of computational abilities of an attacker
- *Conditional* or *computational security*: cryptosystem is secure assuming a computationally bounded adversary, or under certain hardness assumptions (e.g., P<>NP)
  - E.g., DES, 3DES, AES, RSA, DSA, ECC, DH, MD5, SHA
  - Key sizes are much smaller (~128 bits)
- Almost all deployed modern cryptosystems are conditionally secure

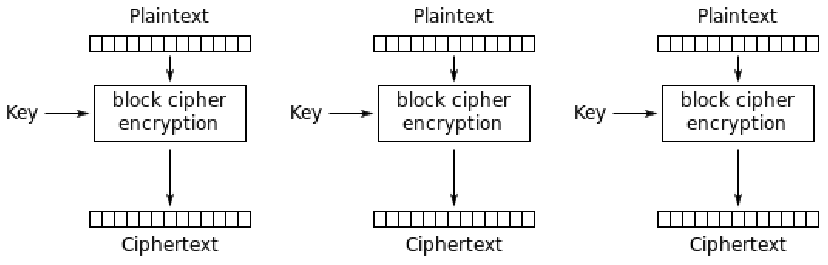
Computational cryptography

- Want to encrypt with shorter keys
  - ➢ Problem: information-theoretic security is impossible if key space is smaller than message space.
- Solution: Use a more practical security notion
  - ➢ It should be infeasible for a computationally bounded attacker to violate security
  - ➢ In practice: attacks should take at least e.g., 2^{128} time

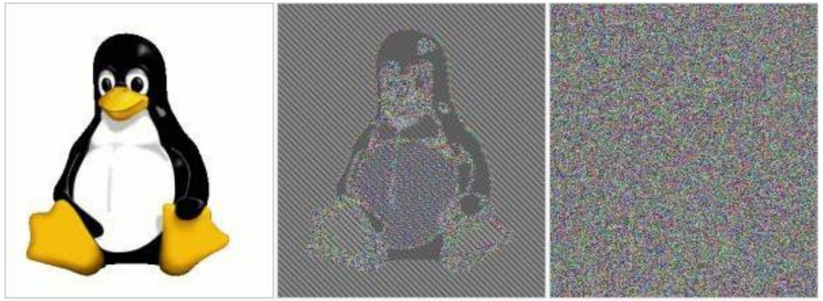
- Landauer Limit: there is a minimum amount of energy required to flip a bit
- ~ 2^{250} atoms in the universe

Block ciphers (e.g., DES, AES)

- Divide data in blocks and encrypt/decrypt each block
- Block ciphers are constructed using **one-way function**
- Fixed length input/output (e.g. 256bit input and 256bit output
- **ECB IS NOT RECOMMENDED**



Electronic Codebook (ECB) mode encryption

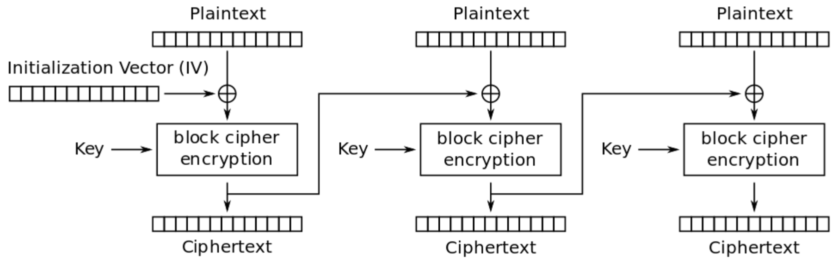


Original image

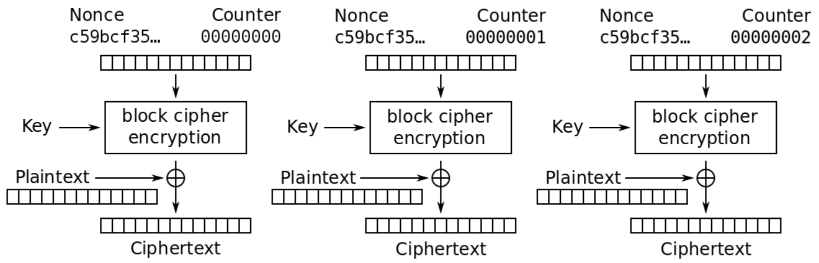
Encrypted using ECB mode

Modes other than ECB result in pseudo-randomness

Other Block cipher Modes



Cipher Block Chaining (CBC) mode encryption



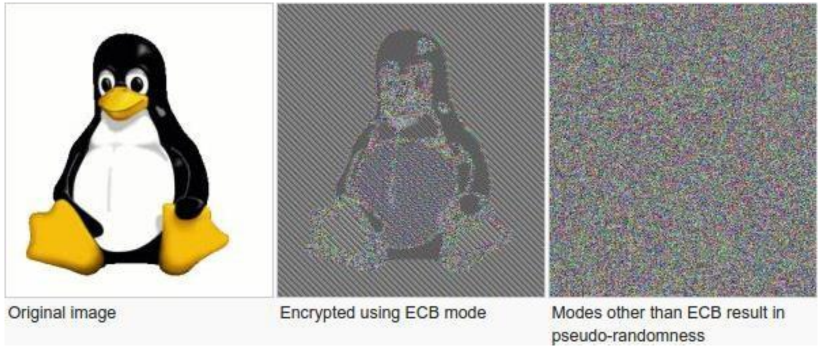
Counter (CTR) mode encryption

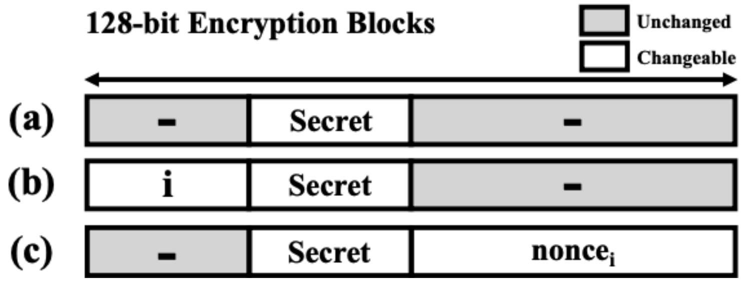
`IV` can be public, but need to ensure to not reuse `IV` for the same key.

Use cases: file/disk encryption and memory encryption.

Use Correct Crypto Primitives

- Ciphertext Side Channels on AMD SEV
- SEV's memory encryption engine uses an XOR-Encrypt-XOR (XEX) mode -> deterministic encryption during the lifetime of a VM



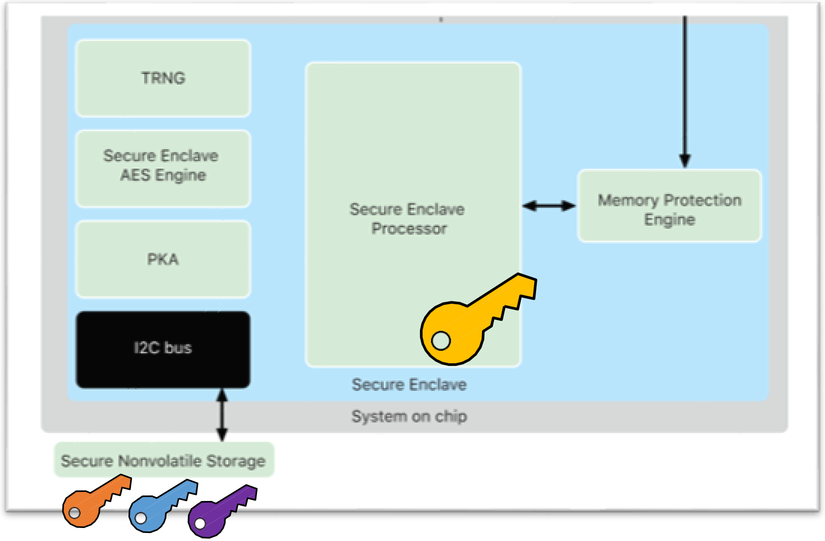


Li et al, CIPHERLEAKS: Breaking Constant-time Cryptography on AMD SEV via the Ciphertext Side Channel, USENIX'21
Li et al, A Systematic Look at Ciphertext Side Channels on AMD SEV-SNP, S&P'22

Encrypt using Short Passcode

- How many attempts do we need to brute-force 6-digit passcode?
- How to mitigate brute-force?
- How to deal with attacks who can copy the data across devices and brute-force in parallel?

Bind Crypto Keys to Device



User data encryption **keys**

A unique ID (**UID**) root cryptographic key.

- Unique to each device
- Randomly generated
- Fused into the SoC at manufacturing time
- Not visible outside the device
- BIG!

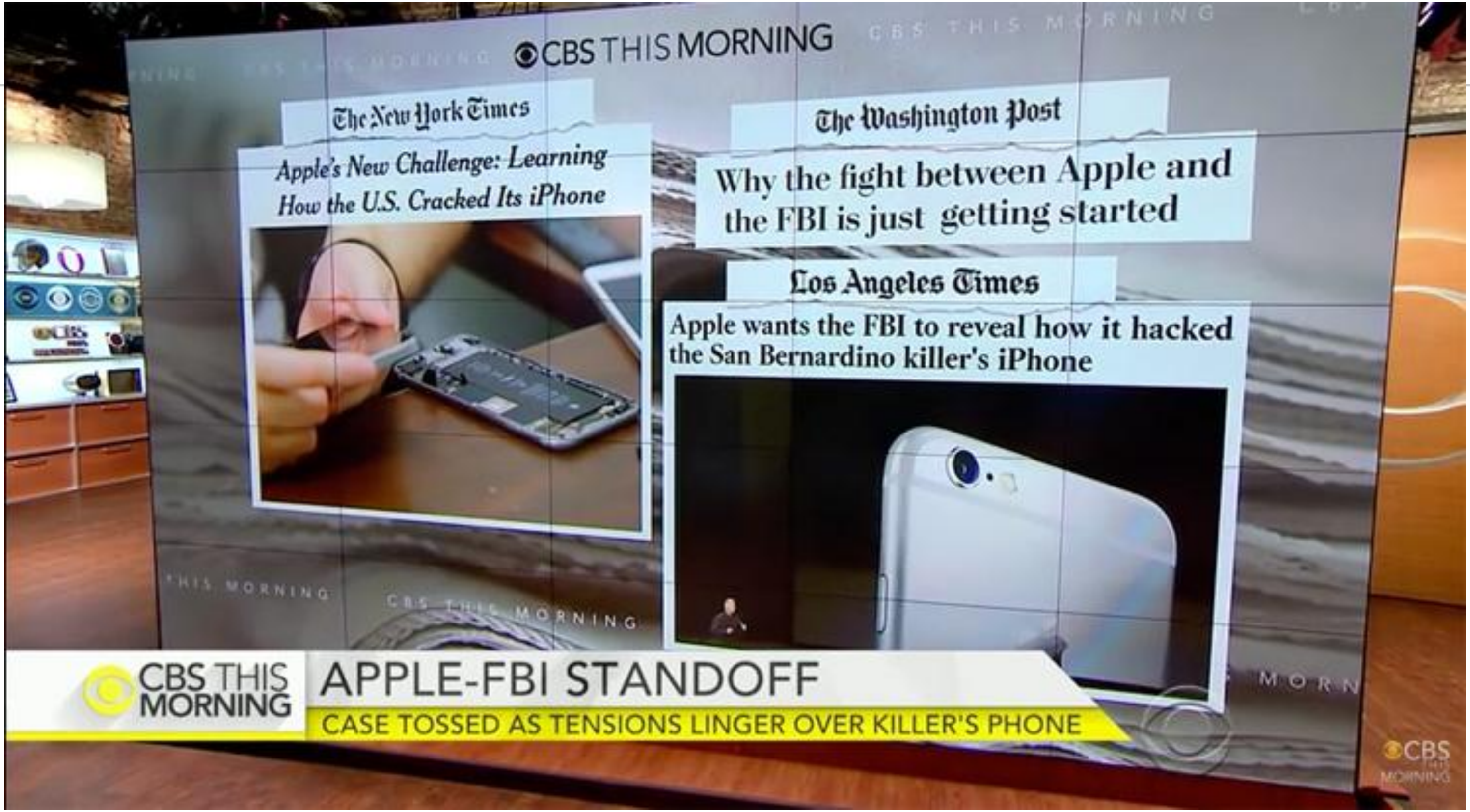
```
Passcode + UID -> encryption entropy
```

Brute-force has to be performed on the device under attack

Combine with other mitigations:

- Escalating time delays
- Erase data when exceeding attempt count

Real-world use case


CBS THIS MORNING
The New York Times
Apple's New Challenge: Learning How the U.S. Cracked Its iPhone
The Washington Post
Why the fight between Apple and the FBI is just getting started
Los Angeles Times
Apple wants the FBI to reveal how it hacked the San Bernardino killer's iPhone
CBS THIS MORNING
APPLE-FBI STANDOFF
CASE TOSSED AS TENSIONS LINGER OVER KILLER'S PHONE

Next time

- Trusted boot and TEE

THE UNIVERSITY
of NORTH CAROLINA
at CHAPEL HILL