Cryptography Part 1

CMSC 23200/33250, Winter 2021, Lecture 7

David Cash & Blase Ur

University of Chicago

Brief Pause: Computer Security Ethics

- Never explore vulnerabilities in someone’s else’s system without their permission!
    - … even if they are easy/obvious
    - … even if you mean no harm.
    - At best it is rude; Usually it is harmful.
    - It is almost always illegal. Trouble with the University as well.

Trying out a vulnerability on your VM is okay.

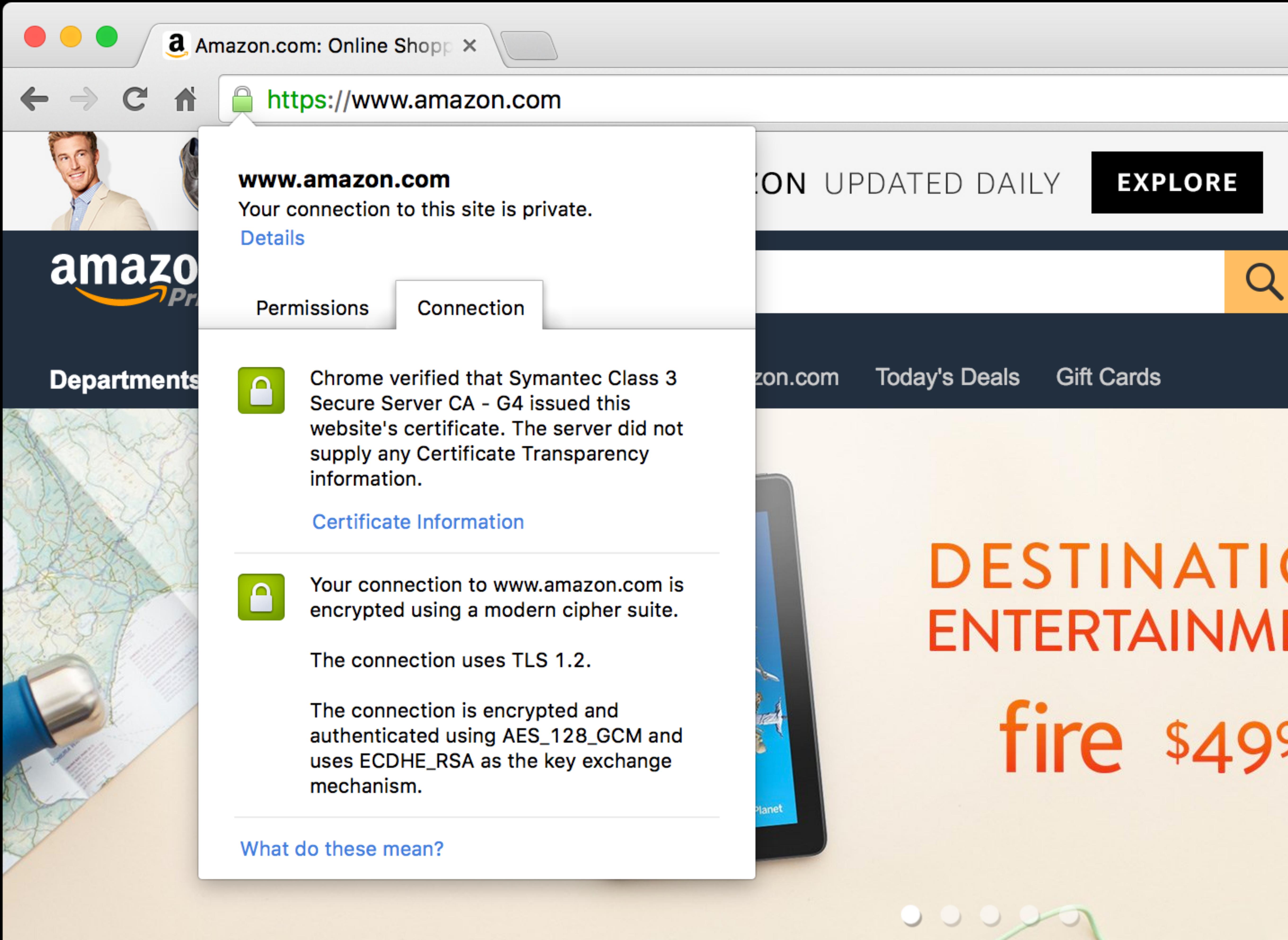
Brief Pause: Computer Security Ethics

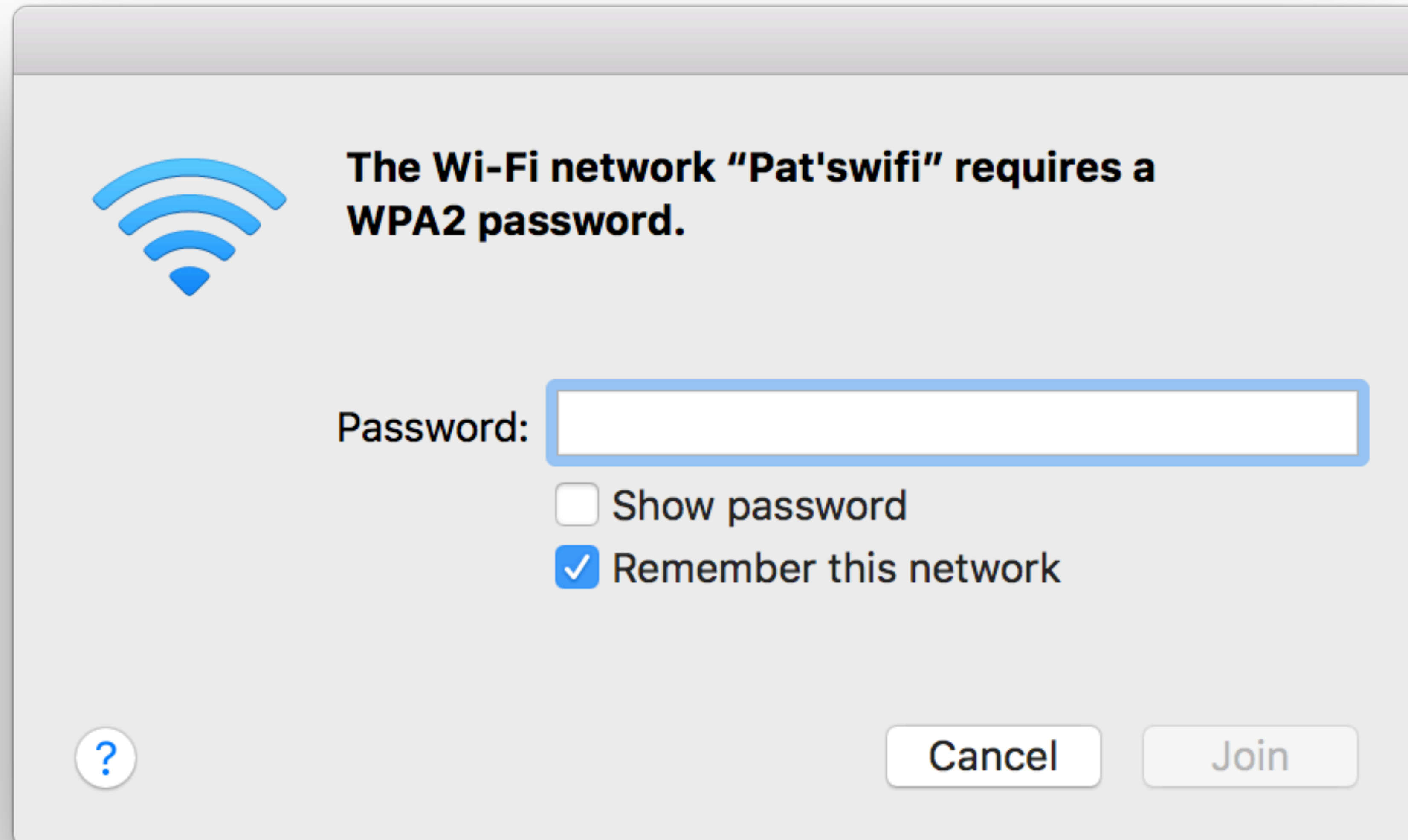
- If you do find a novel vulnerability, do not make it public!
  - … even if it is easy/obvious
  - … even if you mean no harm.
  - It is almost always illegal.
  - Legal gray area: Selling it… please don't.

"**Responsible disclosure"** is the term of art for

  - Privately notifying the vendor and possibly victims,
  - Filing for a CVE,
  - Waiting until it is patched to discuss your finding.

- Sometimes conflicts arise (e.g. vendor won't fix).


Amazon.com: Online Shopp
https://www.amazon.com
www.amazon.com
Your connection to this site is private.
Details
Permissions
Connection
Chrome verified that Symantec Class 3 Secure Server CA - G4 issued this website's certificate. The server did not supply any Certificate Transparency information.
Certificate Information
Your connection to www.amazon.com is encrypted using a modern cipher suite.
The connection uses TLS 1.2.
The connection is encrypted and authenticated using AES_128_GCM and uses ECDHE_RSA as the key exchange mechanism.
What do these mean?
UPDATED DAILY
EXPLORE
Departments
Today's Deals
Gift Cards
DESTINATI
ENTERTAINM
fire
$49


The Wi-Fi network “Pat'swifi” requires a WPA2 password.
Password:
Show password
Remember this network
?
Cancel
Join


nothing
Today 11:11
Can you please come over asap to help me move the couch?
I need to be out of here by 3pm
I guess you forgot your phone at home or something
Delivered
iMessage
Send

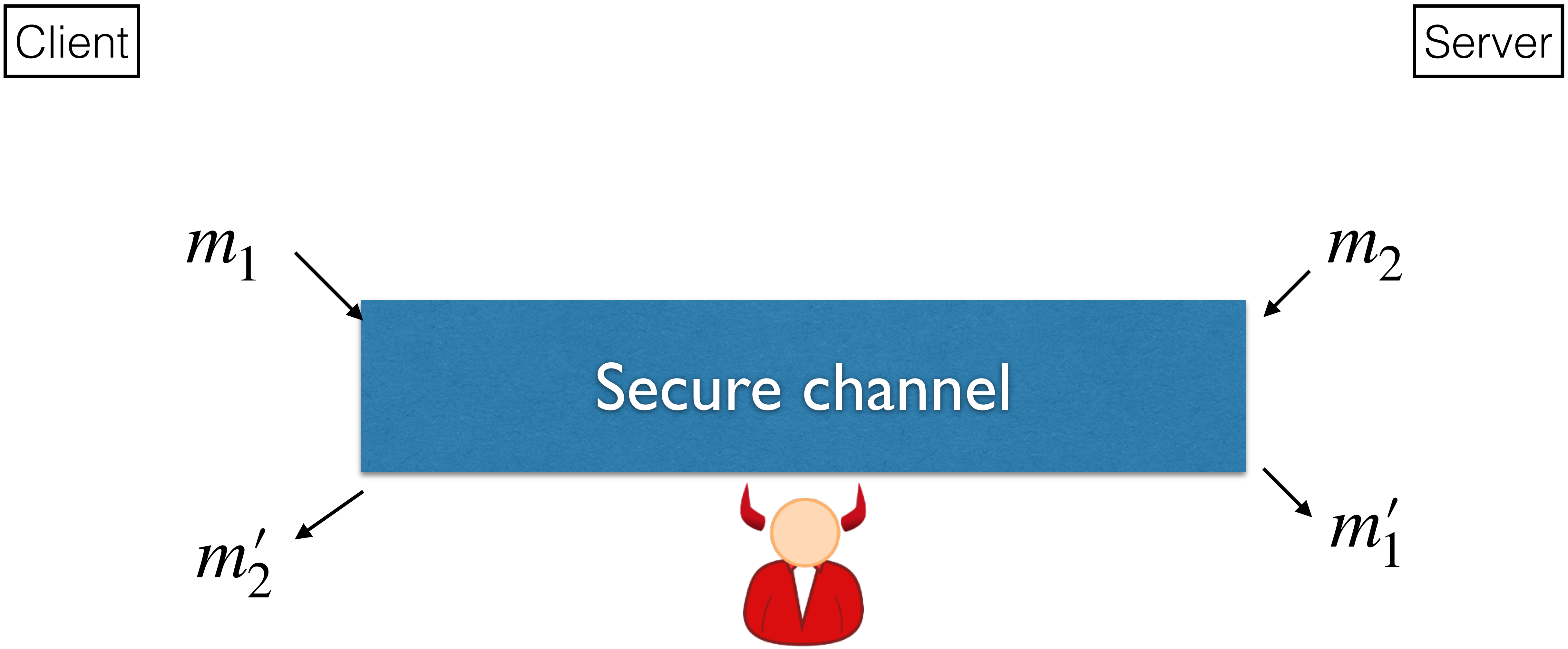

5100
5100

What is Cryptography?

Cryptography involves algorithms with security goals.

Cryptography involves using math to stop adversaries.

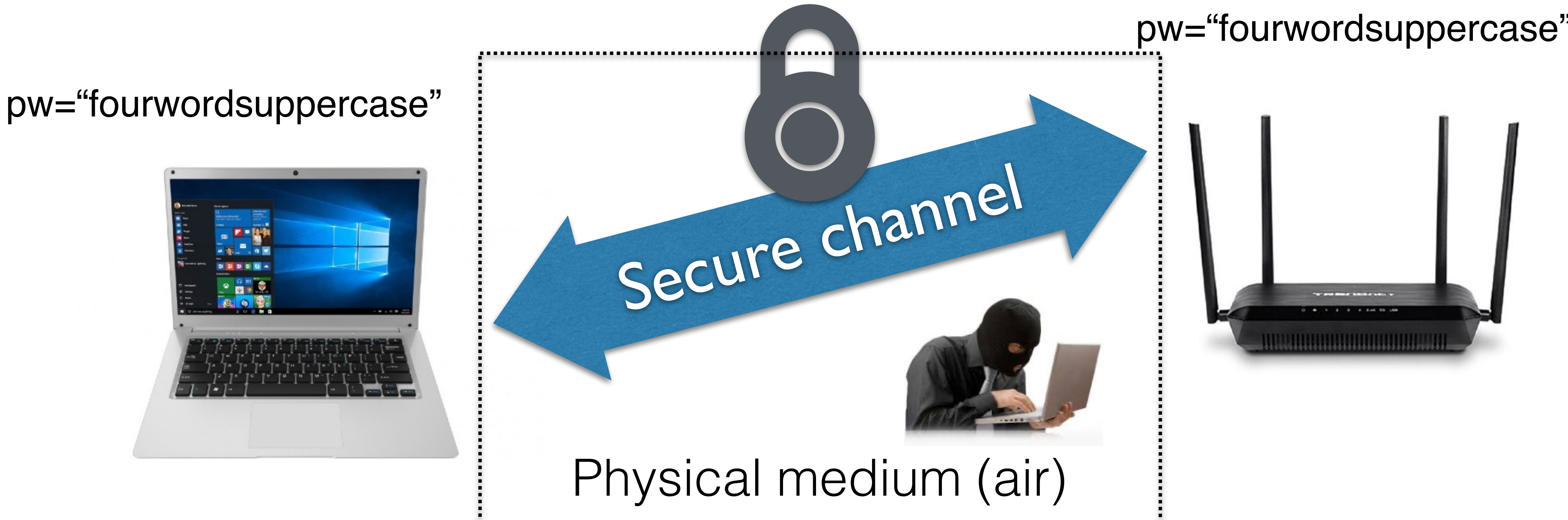
Common Security Goal: Secure Channel



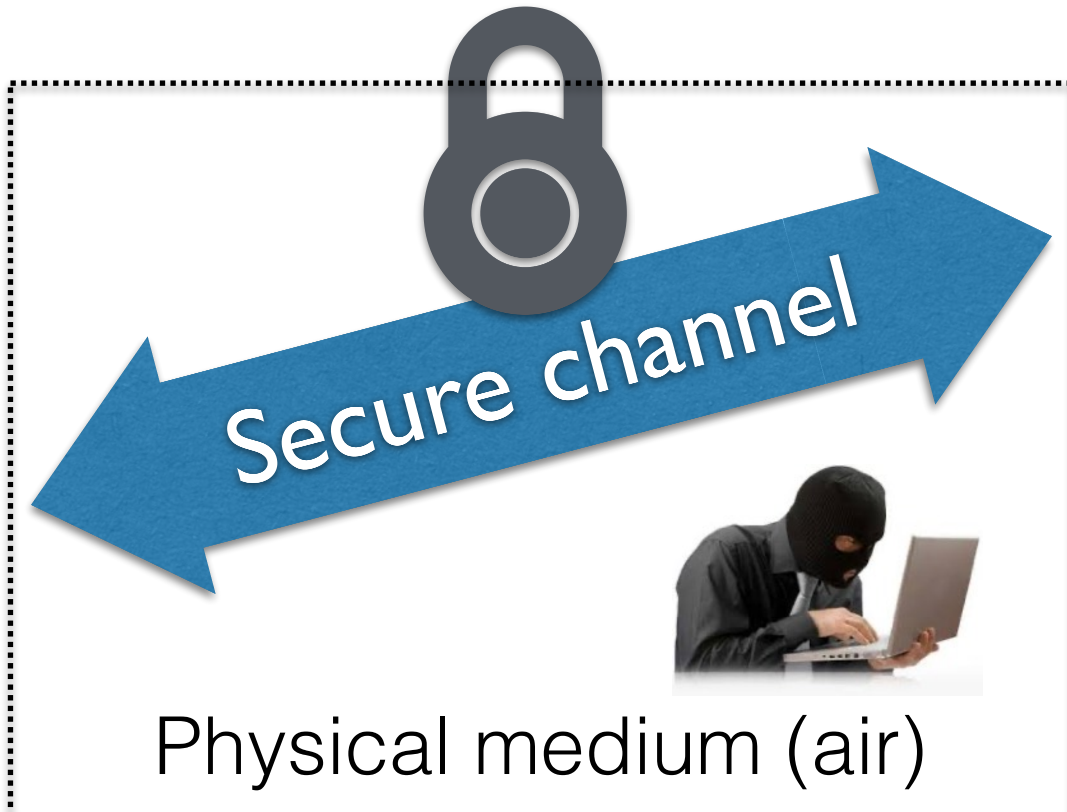
Confidentiality: Adversary does not learn anything about messages m_1, m_2

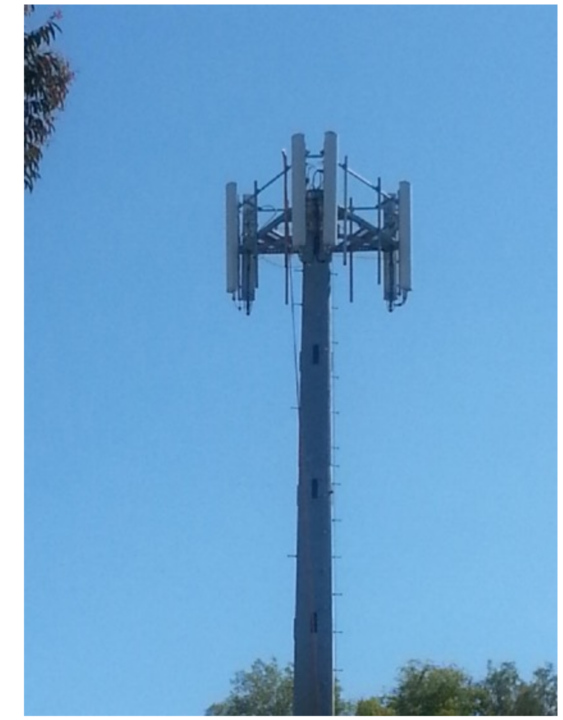
Authenticity: $m_1' = m_1$ and $m_2' = m_2$

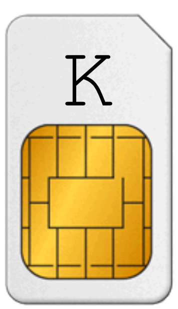
WPA2 (Wi-Fi Protected Access 2): Secure WiFi



GSM Cell Phone Encryption (A5/1, A5/3)



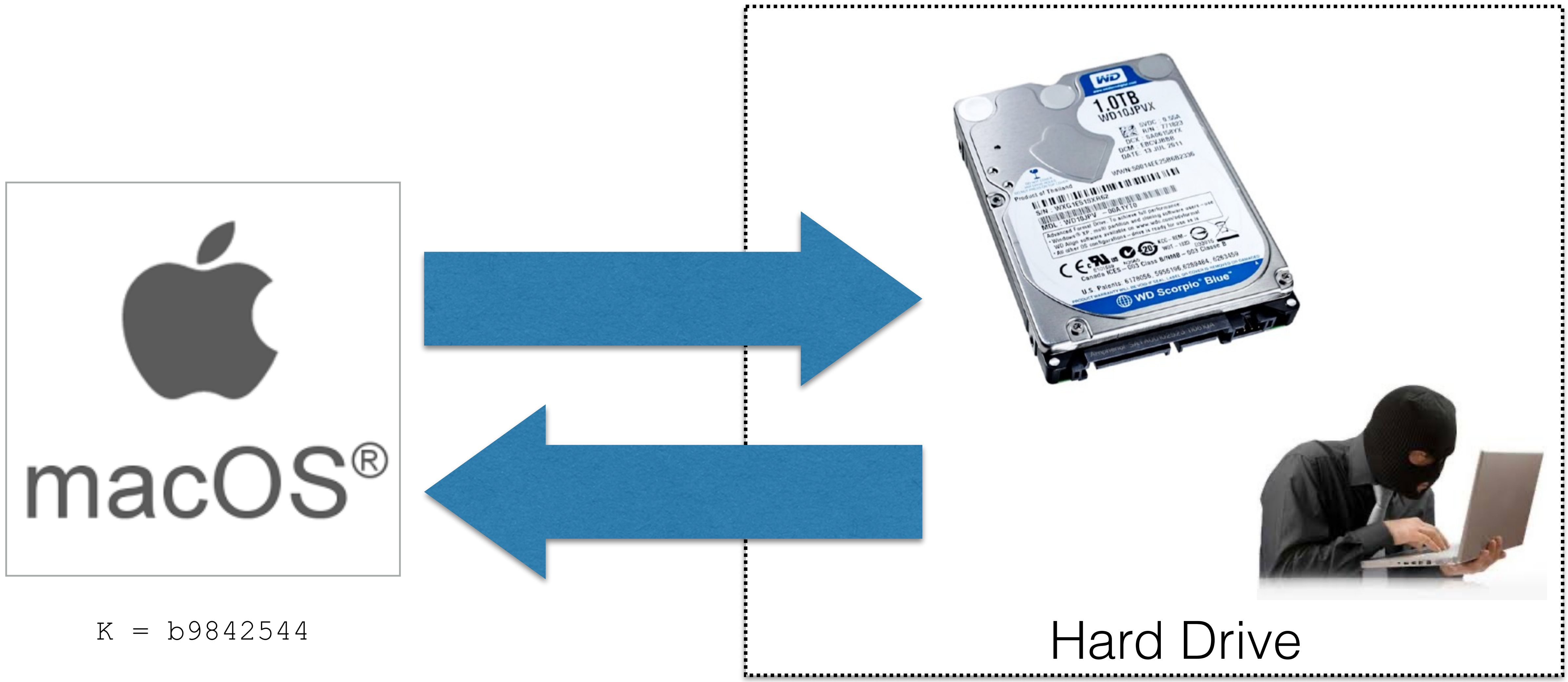




K = b9842544

User	Key
Alice Doe	340934c3
Betty Lee	b9842544
Cheryl Zang	93d94520
Pat Dobbs	2ea0f48d
...	...

Disk Encryption



Crypto in your browser: TLS (Transport Layer Security)

No pre-shared key, yet “guarantees” secret & authenticated communication with amazon.com.

Crypto in CS23200/33250

- A brief overview of major concepts and tools
- Cover (some of) big “gotchas” in crypto deployments
- Cover background for networking and authentication later

Not going to cover math, proofs, or many details.
Consider taking CS284 (Cryptography)!

Four settings for cryptography

Pre-shared key? \ Security Goal	Confidentiality	Authenticity/Integrity
Yes (“Symmetric”)	Symmetric Encryption (aka Secret-key Encryption)	Message Authentication Code (MAC)
No (“Asymmetric”)	Public-Key Encryption	Digital Signatures

Rest of this lecture

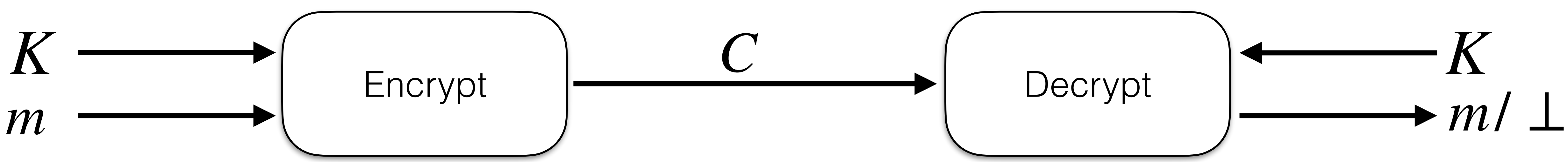
- Symmetric Encryption Basics
- Stream Ciphers
- Block Ciphers

Rest of this lecture

- **Symmetric Encryption Basics**
- Stream Ciphers
- Block Ciphers

Ciphers (a.k.a. Symmetric Encryption)

A cipher is a pair of algorithms Encrypt, Decrypt:



Require that decryption recovers the same message.

Historical Cipher: ROT13 ("Caesar cipher")

Encrypt(K,m): shift each letter of plaintext forward by K positions in alphabet (wrap from Z to A).

Plaintext: **DEFGH**
Key (shift): 3
Ciphertext: FGHKL

Plaintext: **ATTACKATDAWN**
Key (shift): 13
Ciphertext: NGGNPXNGQNJA

Historical Cipher: Substitution Cipher

Encrypt(K,m): Parse key K as a permutation π on {A,… Z}.
Apply π to each character of m.

P: **ATTACKATDAWN**

K: π →

C: ZKKZAMZKYZGT

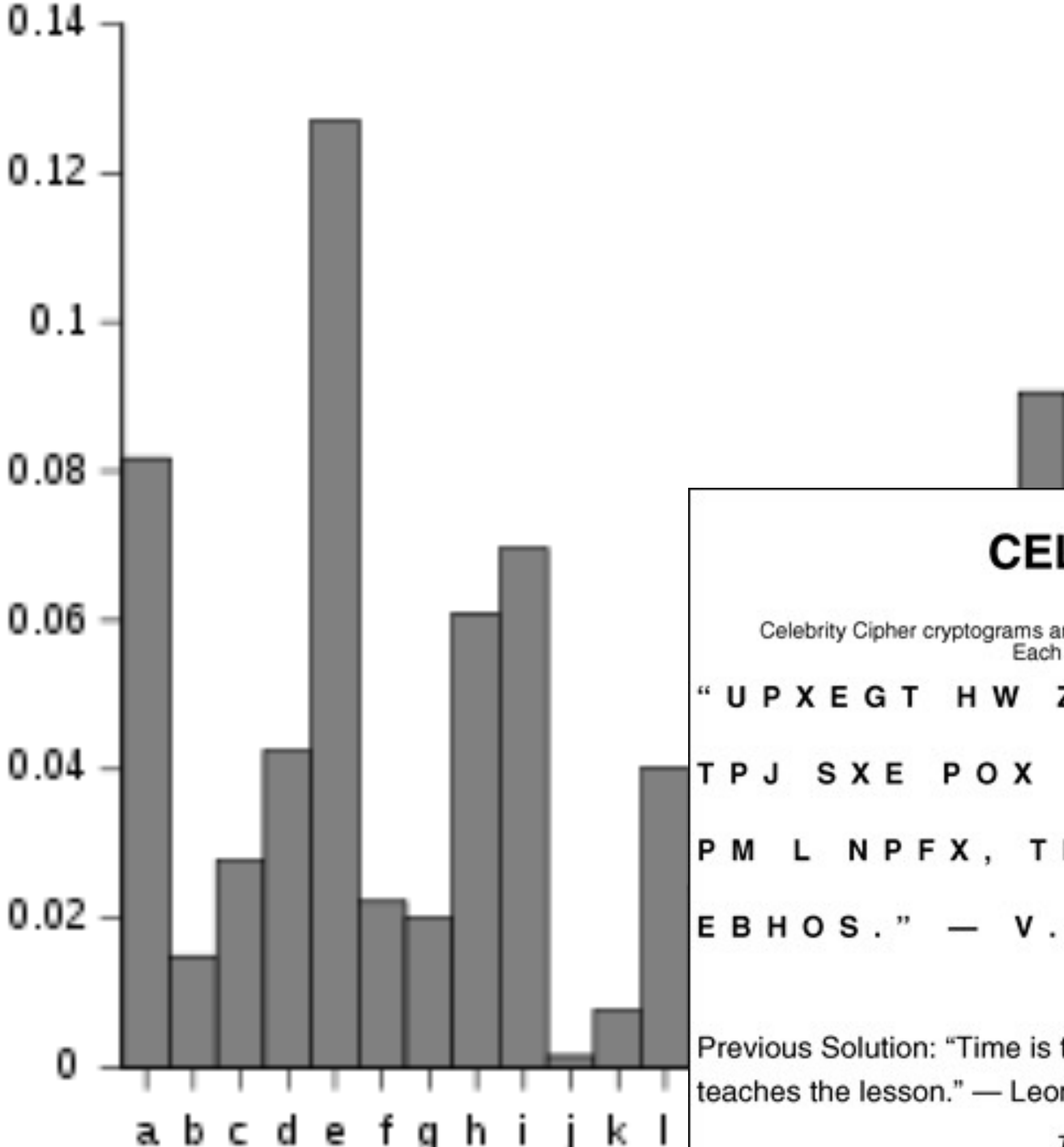
x	π(x)
A	Z
B	U
C	A
D	Y
E	R
F	E
G	X
H	B
I	D
J	C
K	M
L	Q
M	H
N	T
O	I
P	S
Q	V
R	N
S	P
T	K
U	O
V	F
W	G
X	W
Y	L
Z	J

How many keys?

$26! \approx 2^{88}$

9 million years to try all keys at rate of 1 trillion/sec

Cryptanalysis of Substitution Cipher



CELEBRITY CIPHER

by Luis Campos

Celebrity Cipher cryptograms are created from quotations by famous people, past and present. Each letter in the cipher stands for another.

" U P X E G T H W Z H F X Y L F H O S L N P F X . H M T P J S X E P O X V P G A V G P O S L E E B X X O A P M L N P F X , T P J ' C X Z P W E E B X V B P Z X E B H O S . " — V . W . Y X G V H O

Previous Solution: "Time is the cruelest teacher; first she gives the test, then teaches the lesson." — Leonard Bernstein

TODAY'S CLUE: N equals J

Quick recall: Bitwise-XOR operation

We will use bit-wise XOR:

$$\begin{array}{r} 0101 \\ \oplus\ 1100 \\ \hline 1001 \end{array}$$

Some Properties:

- $X \oplus Y = Y \oplus X$
- $X \oplus X = 000\ldots0$
- $X \oplus Y \oplus X = Y$

Cipher Example: One-Time Pad

Key K: Bitstring of length L

Plaintext M: Bitstring of length L

$\boxed{\text{Encrypt(K,M): Output } K \oplus M}$

$\boxed{\text{Decrypt(K,C): Output } K \oplus C}$

Example:

$$\begin{array}{r} 0101 \\ \oplus\ 1100 \\ \hline 1001 \end{array}$$

Correctly decrypts because

$$K \oplus C = K \oplus (K \oplus M) = (K \oplus K) \oplus M = M$$

Q: Is the one-time pad secure?

Bigger Q: What does "secure" even mean?

Evaluating Security of Crypto Algorithms

<u>Kerckhoff's Principle</u>: Assume adversary knows your algorithms and implementation. The only thing it doesn't know is the key.

1. Quantify adversary goals

   Learn something about plaintext? Spoof a message?

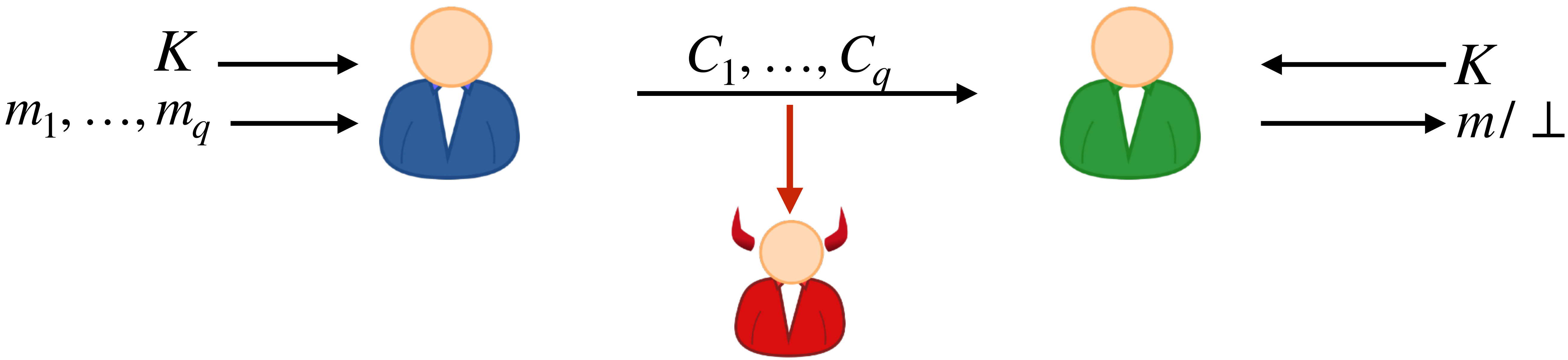
2. Quantify adversary capabilities

   View ciphertexts? Probe system with chosen inputs?

3. Quantify computational resources available to adversary

   Compute cycles? Memory?

Breaking Encryption - A Basic Game

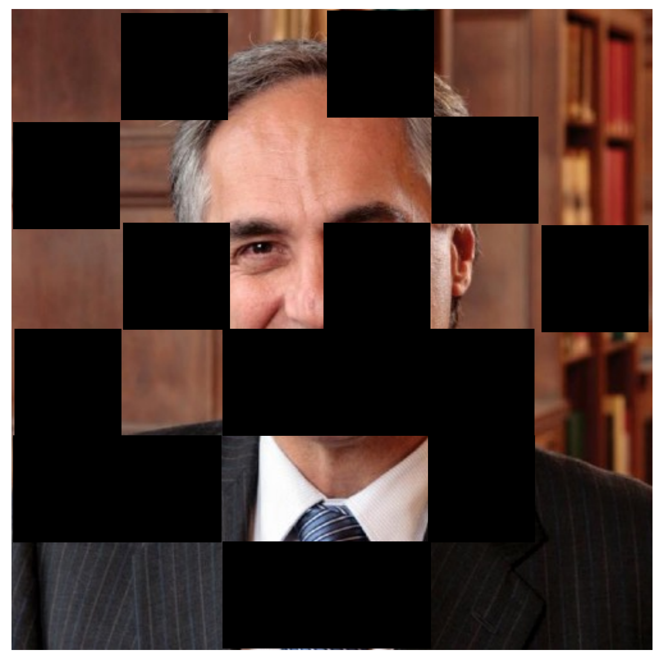


Ciphertext-only attack: The adversary sees ciphertexts and attempts to recover some useful information about plaintexts.

More attack settings later.

Recovering Partial Information; Partial Knowledge

- Recovering entire messages is useful
- But recovering **partial information** is also be useful



A lot of information is missing here.

But can we say who this is?

- Attacker may know large parts of plaintext already (e.g. formatting strings or application content). The attacker tries to obtain something it doesn't already know.

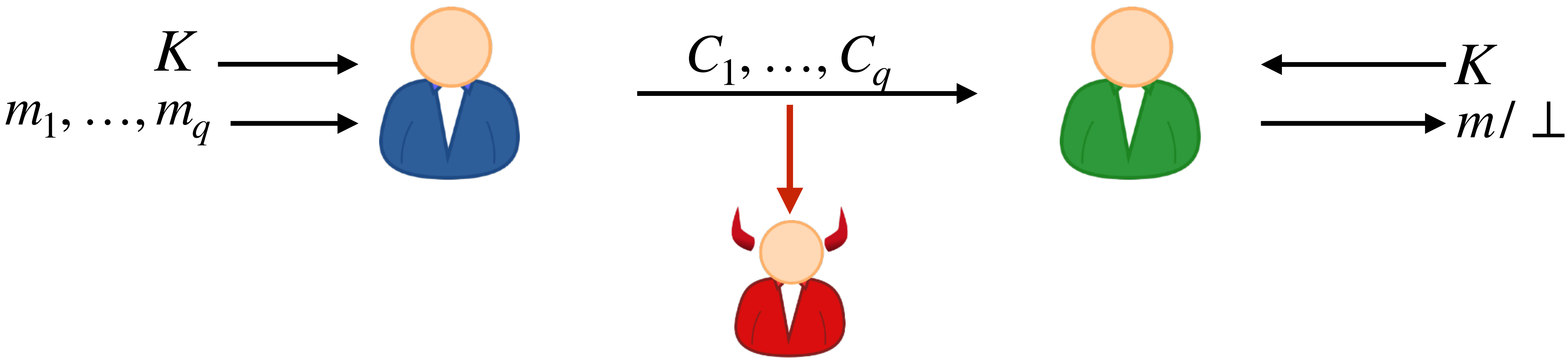
```
M = http://site.com?password=[redacted]
```

“Attacks” versus “Security”

An **attack** is successful as long as it recovers some info about plaintext that is useful to adversary.

Encryption should hide all possible partial information about plaintexts, since what is useful is situation-dependent.

Attacks can succeed without recovering the key



Full break: Adversary recovers K, decrypts all ciphertexts.

However: Clever attacker may compromise encryption without recovering the key.

Security of One-Time Pad

<u>Claim</u>: If adversary sees **only one** ciphertext under a random key, then any plaintext is equally likely, so it cannot recover any partial information <u>besides plaintext length</u>.

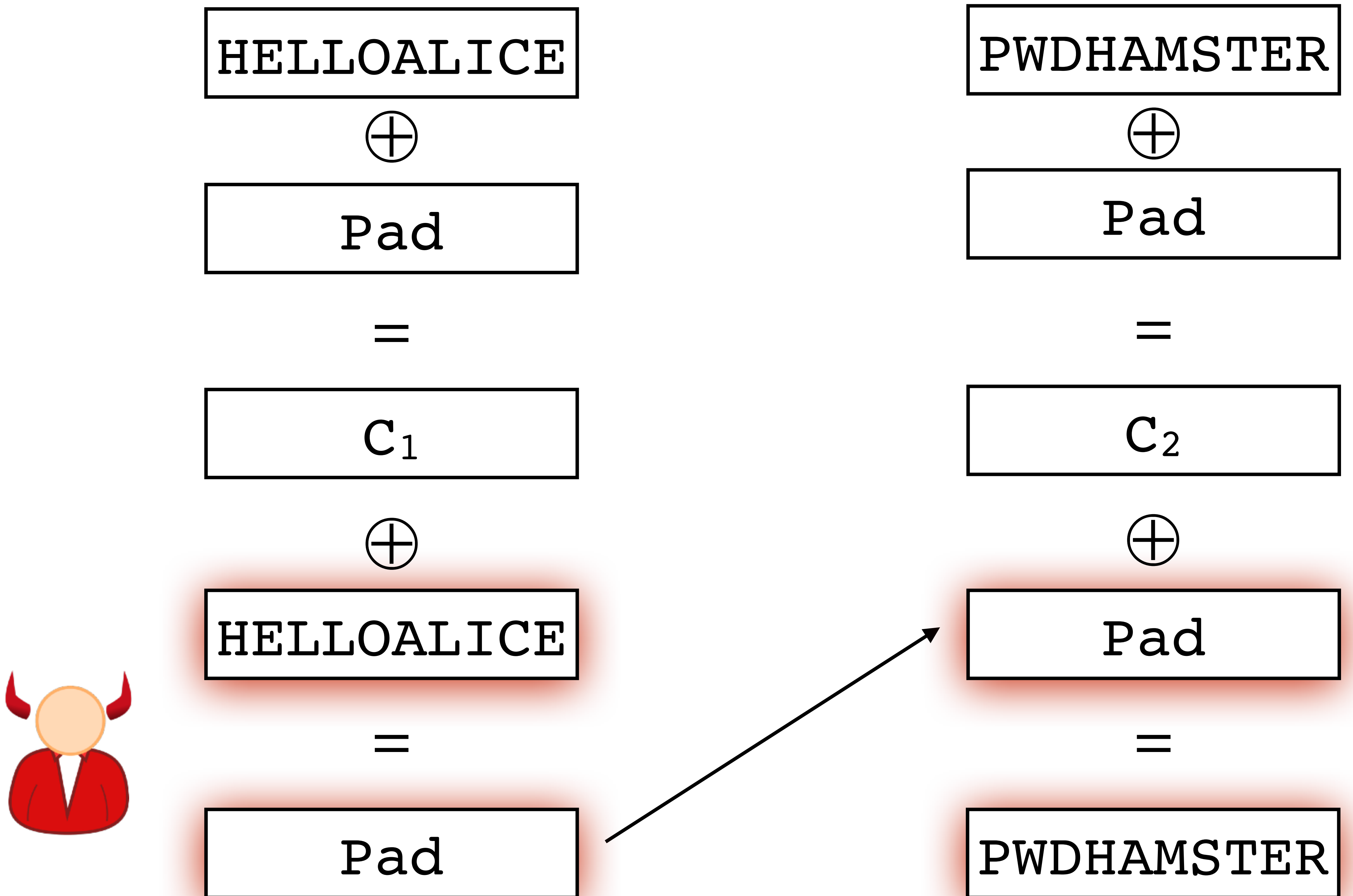
Ciphertext observed:	`10111`
Possible plaintext:	`00101`
⇒ Possible key:	`10010`

1. Adversary goal: Learn partial information from plaintext
2. Adversary capability: Observe a single ciphertext
3. Adversary compute resources: Unlimited time/memory (!)

Issues with One-Time Pad

1. Reusing a pad is insecure
2. One-Time Pad is *malleable*
3. One-Time Pad has a long key

Issue #1: Reusing a One-Time Pad is Insecure



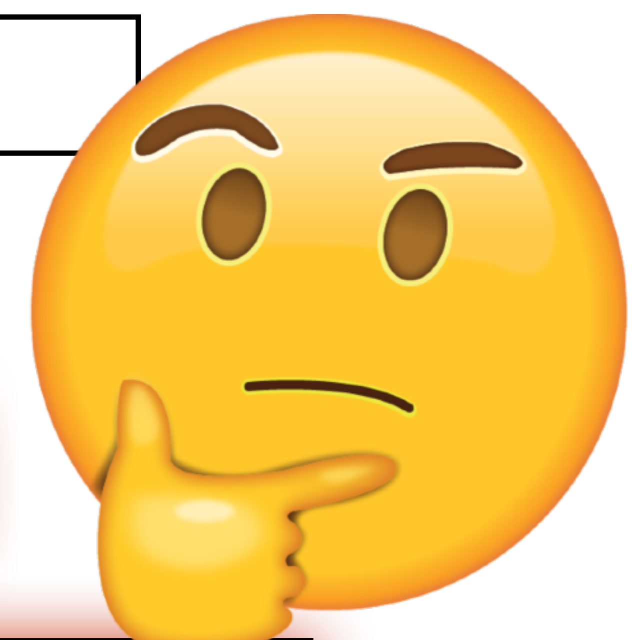
Issue #1: Reusing a One-Time Pad is Insecure

Has led to real attacks:

- Project Venona (1940s) attack by US on Soviet encryption
- MS Windows NT protocol PPTP
- WEP (old WiFi encryption protocol)
- Fortiguard routers! [link]

$=$ $\quad$ $=$

$\boxed{c_1}$ $\quad$ $\boxed{c_2}$



$\boxed{c_1} \oplus \boxed{c_2}$

$= \boxed{\texttt{S3CR3T1234}} \oplus \boxed{\texttt{3L33THXRRR}}$

Issue #2: One-Time Pad is *Malleable*

PAYALICE$1

$\oplus$

Pad

=

C

$\oplus$

000ALICE00

$\oplus$

000DAVID00

=

C′

Decrypt(Pad, C') = PAYDAVID$1

Issue #3: One-Time Pad Needs a Long Key

Can prove: Any cipher as secure as the OTP must have:

Key-length ≥ Plaintext-length

In practice:

- Use *stream cipher*: Encrypt(K,m) = G(K)⊕m
- Add *authentication tag*
- Use *nonces* to encrypt multiple messages

Outline

- Symmetric Encryption Basics
- **Stream Ciphers**
- Block Ciphers

Tool to address key-length of OTP: Stream Ciphers

<u>Stream cipher syntax:</u> Algorithm G that takes one input and produces an very long bit-string as output.

Usually very, very large
(petabytes if needed)

<u>Key/Seed k:</u> `1100..11` Typically 16 or 32 bytes.

G

<u>G(k):</u> `11111010001000111010100101000101100100111100…`

$\oplus$ `DONUTSDONUTSDONUTSDONUTSDONUTSDONUTSDONUTSDON`

Use G(seed) in place of pad.
Still malleable and still one-time, but key is shorter.

Stream Cipher Security Goal (Sketch)

<u>Security goal:</u> When **`k`** is random and unknown, **`G(k)`** should “look” random.

… even to an adversary spending a lot of computation.

Much stronger requirement that “passes statistical tests”.

<u>Brute force attack:</u> Given **`y=G(k)`**, try all possible **`k`** and see if you get the string **`y`**.

<u>Clarified goal:</u> When **`k`** is random and unknown, **`G(k)`** should “look” random to anyone with less computational power needed for a brute force attack.

(keylength = 256 is considered strong now)

Aside: Fundamental Physical Property of the Universe*

There exist (1-to-1) functions (say on bitstrings) that are:
1) Very fast to evaluate
2) Computationally infeasible to reverse

The disparity can be almost arbitrarily large!

Evaluating y = f(x) may only take a few cycles….

… and finding x from y within the lifetime of the universe may not be possible, even with a computer made up of every particle in the universe.

**conjectured, but unproven property*

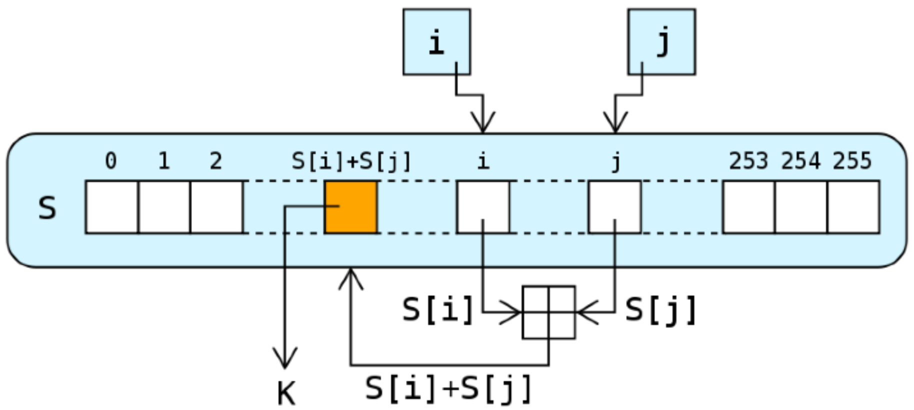
Computational Strength

# Steps	Who can do that many?
2^{56}	Strong computer with GPUs
2^{80}	All computers on Bitcoin network in 4.5 hours
2^{128}	Very large quantum computer? (Ask Fred+Bill)*
2^{192}	Nobody?
2^{256}	Nobody?

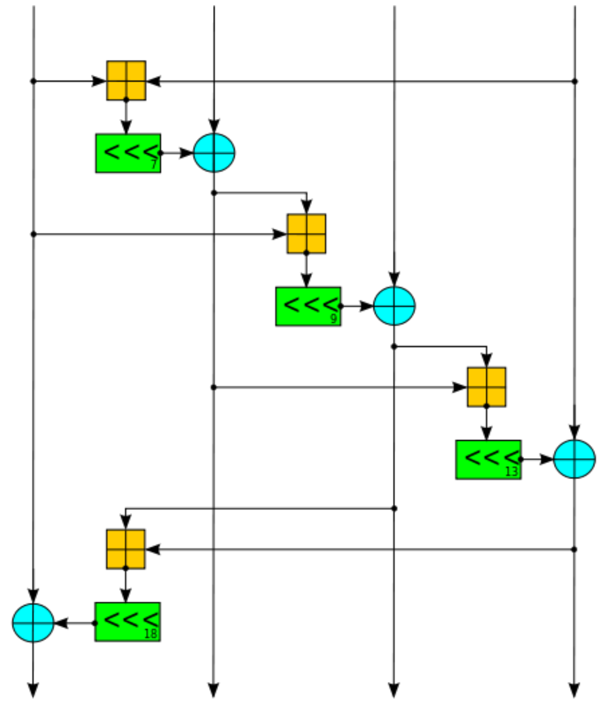
*Not directly comparable but this is an estimate of equivalent power. Quantum computers are most effective against public-key crypto, but they also speed up attacks on symmeric-key crypto. (More next week.)

Practical Stream Ciphers

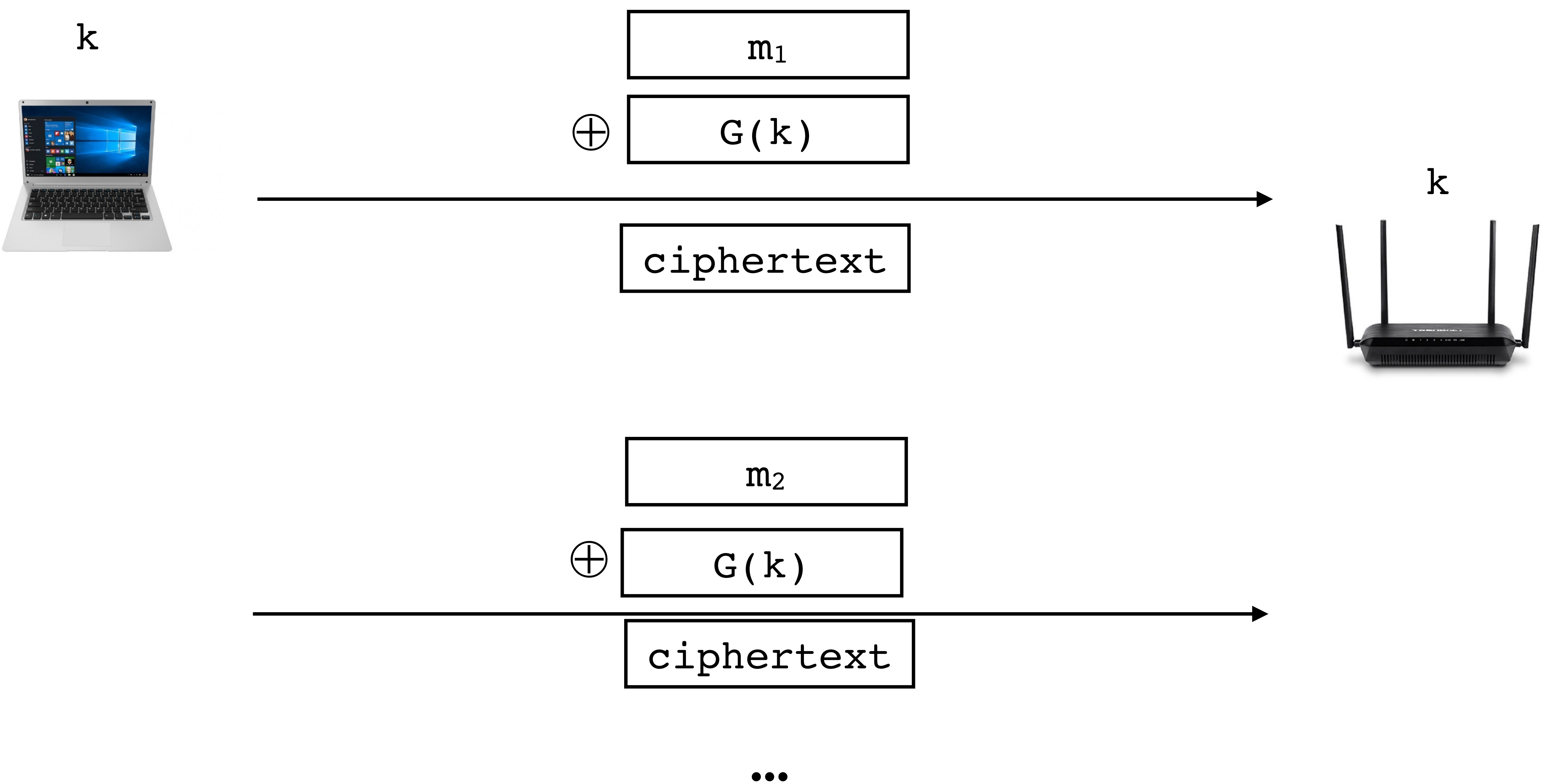
RC4 (1987)**:** "Ron's Cipher #4". Mostly retired by 2016.



ChaCha20 (2007)**:** Successfully deployed replacement. Supports *nonces*.



Pad reuse can still happen with stream ciphers



Addressing pad reuse: Stream cipher with a nonce

<u>Stream cipher with a nonce:</u> Algorithm G that takes **two inputs** and produces a very long bit-string as output.

<u>Nonce IV:</u>	<u>Key/Seed k:</u>
`1100..11`	`1100..11`

<u>G(IV,k):</u> `11111010001000111010100101000101100100111100…`

- “nonce” = “number once”.
- Usually denoted IV = “initialization vector”

<u>Security goal:</u> When **`k`** is random and unknown, **`G(IV,k)`** should “look” random and independent for each value of **`IV`**.

Solution 1: Stream cipher with a nonce

k

IV←0

m_1

$\oplus$ G(IV,k)

IV | ciphertext

k

IV←IV+1

m_2

$\oplus$ G(IV,k)

IV | ciphertext

...

- If nonce repeats, then pad repeats

Example of Pad Re-use: WEP

Warning: Broken

IEEE 802.11b WEP: WiFi security standard ’97-‘03

IV

IV is 24-bit wide counter

- Repeats after 2^{24} frames ($\approx$16 million)
- IV is often set to zero on power cycle

Solutions: (WPA2 replacement)
- Larger IV space, or force rekeying more often
- Set IV to combination of packet number, address, etc

Example of Pad Re-use: WEP

Warning: Broken

IEEE 802.11b WEP: WiFi security standard ’97-‘03

BIZ & IT —

Serious flaw in WPA2 protocol lets attackers intercept passwords and much more

KRACK attack is especially bad news for Android and Linux users.

DAN GOODIN - 10/15/2017, 11:37 PM

- Re
- Of

parameters to their initial values. KRACK forces the nonce reuse in a way that allows the encryption to be bypassed. Ars Technica IT editor Sean Gallagher has much more about KRACK here.

Solutions: (W

- Larger IV sp
- Set IV to combination of packet number, address, etc

Issues with One-Time Pad

1. Reusing a pad is insecure ✓ *Use unique nonces*
2. One-Time Pad is *malleable*
3. One-Time Pad has a long key ✓ *Use stream cipher with short key*

More difficult to address; We will return to this later.

Rest of this lecture

- Symmetric Encryption Basics
- Stream Ciphers
- **Block Ciphers**

Next Up: Blockciphers

Blockciphers are a ubiquitous crypto tool applied to many different problems.

Informal definition: A blockcipher is essentially a substitution cipher with a very large alphabet and a very compact key. Require that efficient algorithms for forward and backward directions.

Typical parameters:
Alphabet = $\{0,1\}^{128}$
Key length = 16 bytes.

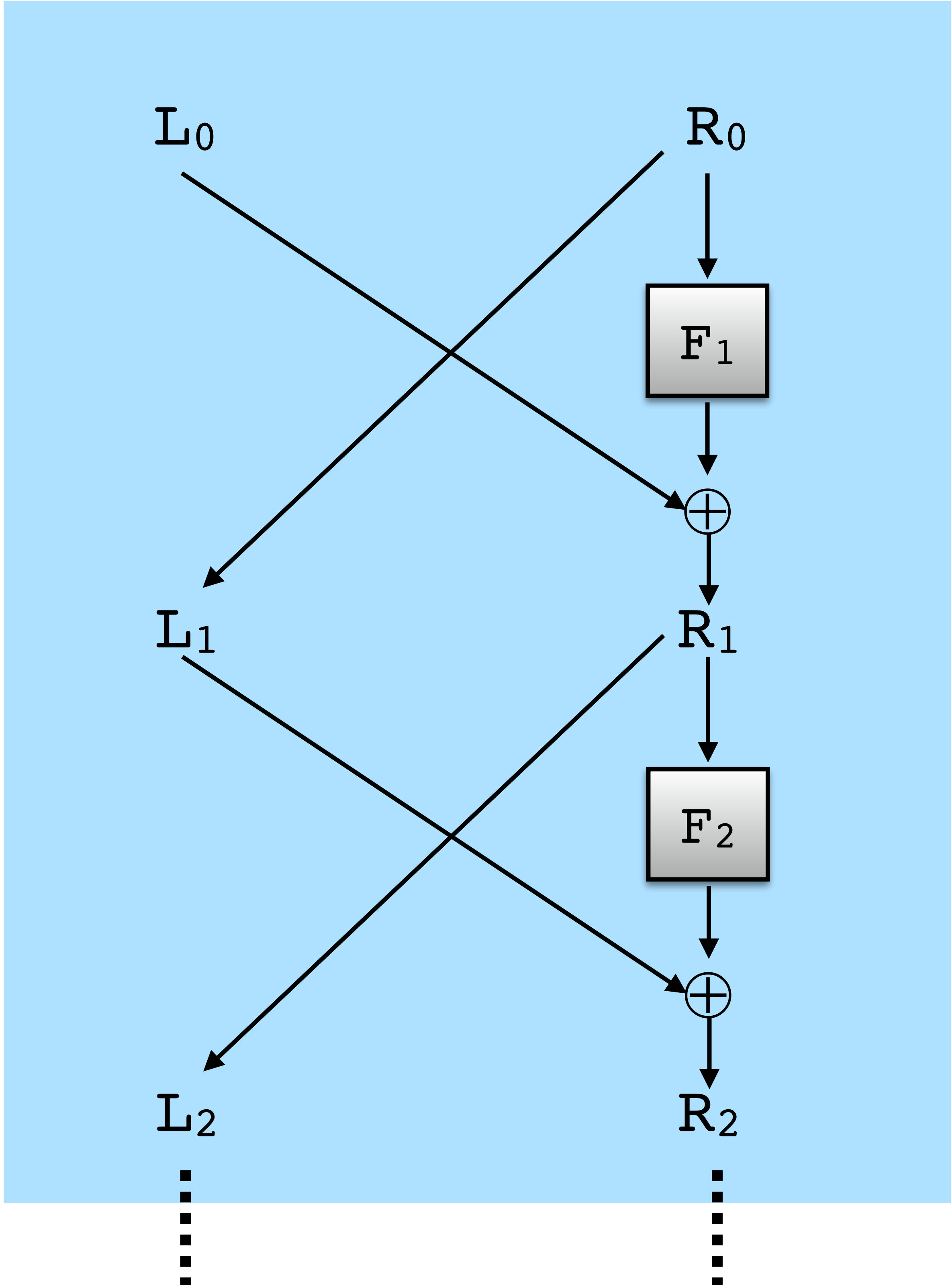
Plan: Build many higher-level protocols from a good blockcipher.

Now: Two example blockciphers, DES and AES.

Data Encryption Standard (DES)

- Originally a designed by IBM
- Parameters adjusted by NSA
- NIST Standard in 1976
  - Block length n = 64
  - Key length k = 56

Parses input block into 32-bit chunks and applies 16 rounds of a “Feistel Network”



DES is Broken

Warning: Broken

Attack	Complexity	Year
Biham&Shamir	2^{47} encrypted blocks	1992
DESCHALL	41 days	1997
EFF Deepcrack	4.5 days	1998
EFF Deepcrack	22 hours	1999

- 3DES (“Triple DES”) is still used by banks
- 3DES encrypts three times (so key length is 118)
- 3DES is not known to be broken but should be avoided

crack.sh

GET CRACKING

NOTE: There are currently extremely high wait times.
We're in the process of adding capacity to speed things up.

QUEUE WAIT TIME:
Standard 46.2 Days, ASAP 1.0 Days

These are the types of DES cracking jobs that we support:

Windows LM/NTLMv1 Authentication
PPTP VPNs WPA-Enterprise
des_crypt() Hashes
DES Kerberos5 Known Plaintext DES

SUBMIT A JOB!

Token:

Priority: Enter Token For Pricing

PAY WITH CARD

WARNING: Charges will show up on your credit card statement as from "crack.sh" and processed through Stripe. We've experienced a high number of our charges being reported as fraudulent, so we'll be blacklisting any accounts that contest charges for jobs submitted. If you wish to cancel a job or have any issues, please email david@toorcon.org and we'll be happy to cancel and refund any charges.

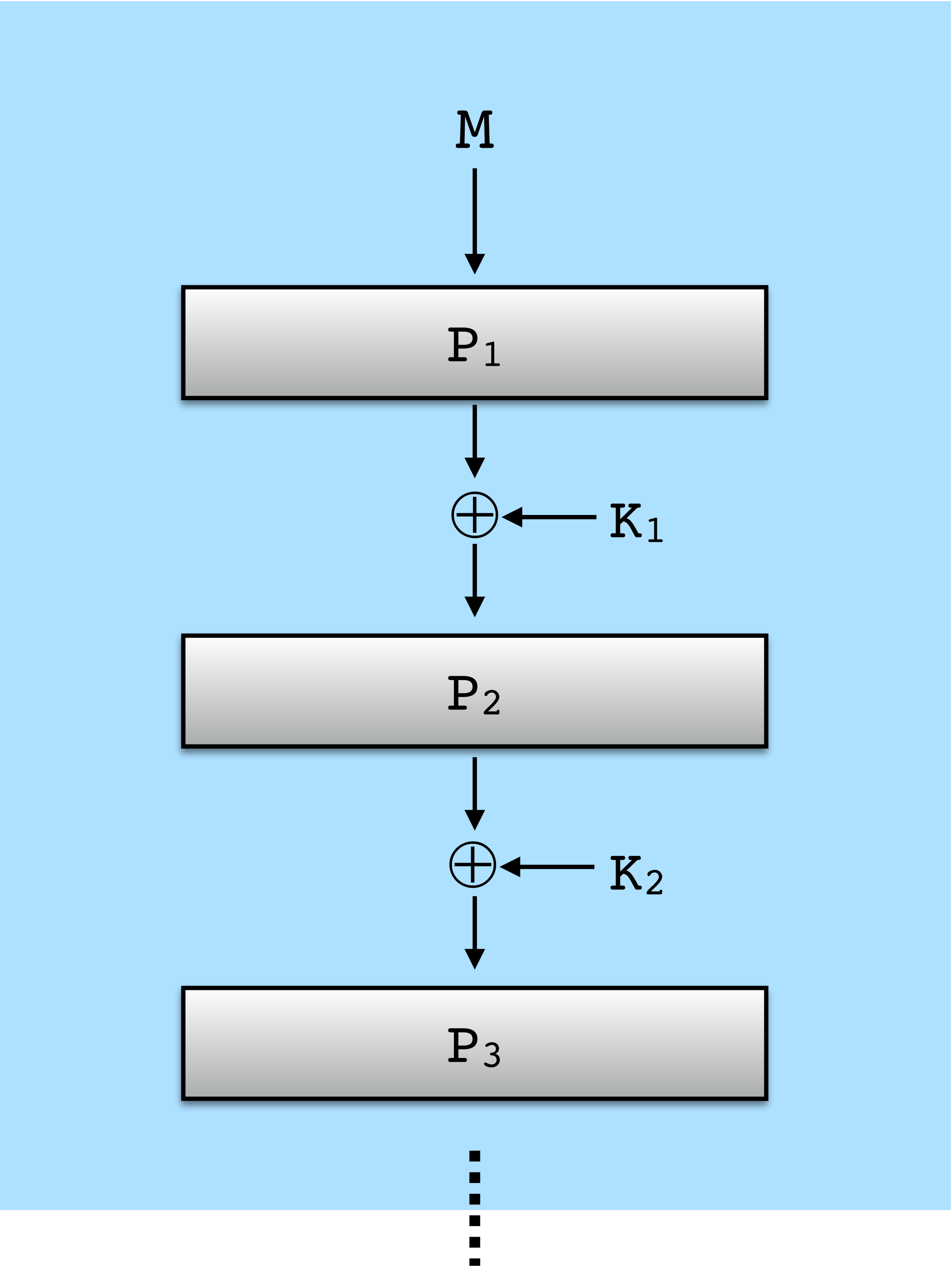
Advanced Encryption Standard (AES)

- NIST ran competition to replace DES starting in 1997
- Several submissions, *Rijndael* chosen and standardized
- AES is now the gold standard blockcipher
- Very fast; Intel chips even have AES instructions

Advanced Encryption Standard (AES)

- Due to Rijmen and Daemen
  - Block length n = 128
  - Key length k = 128,192,256

- Different structure from DES.
- 10 rounds of "substitution-permutation network"



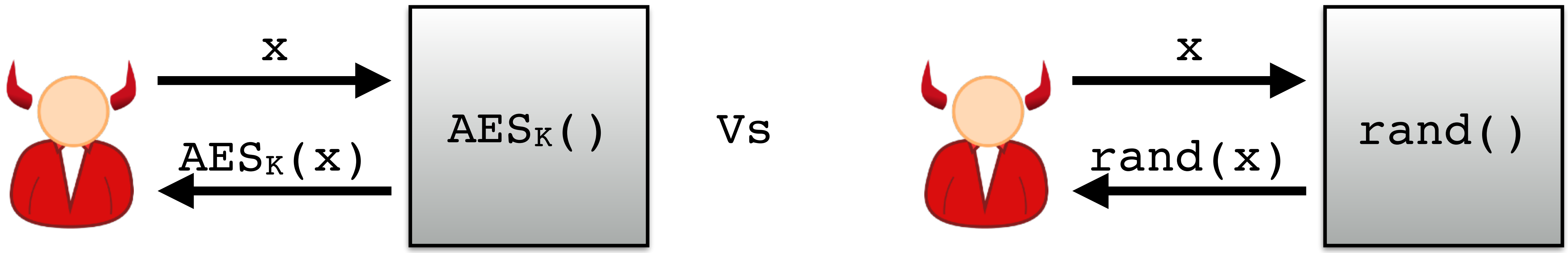
AES is not (know to be) broken

Attack	Complexity	Year
Bogdanov et al.	$\approx 2^{126.1}$	2011

- Compare to trying all keys: $2^{126.1} \approx 2^{128}/4$
- Always prefer AES for a blockcipher if setting can support it (i.e. everything except low-power hardware)

Blockcipher Security

- AES is thought to be a good "Pseudorandom Permutation"



- Outputs all look random and independent, even when inputs are maliciously controlled.
- Formal definition in CS284.

Example - AES Input/Outputs

- Keys and inputs are 16 bytes = 128 bits

- K1: 9500924ad9d1b7a28391887d95fcfbd5
- K2: 9500924ad9d1b7a28391887d95fcfbd<u>6</u>

AES_{K1}(00..00)=8b805ddb39f3eee72b43bf95c9ce410f
AES_{K1}(00..01)=9918e60f2a20b1b81674646dceebdb51
AES_{K2}(00..00)=1303270be48ce8b8dd8316fdba38eb04
AES_{K2}(00..01)=96ba598a55873ec1286af646073e36f6

So we have a blockcipher…

- Now what?

  It only processes 16 bytes at a time, and I have a whole lot more data than that.

  This next step is where everything flies off the rails in implementations…

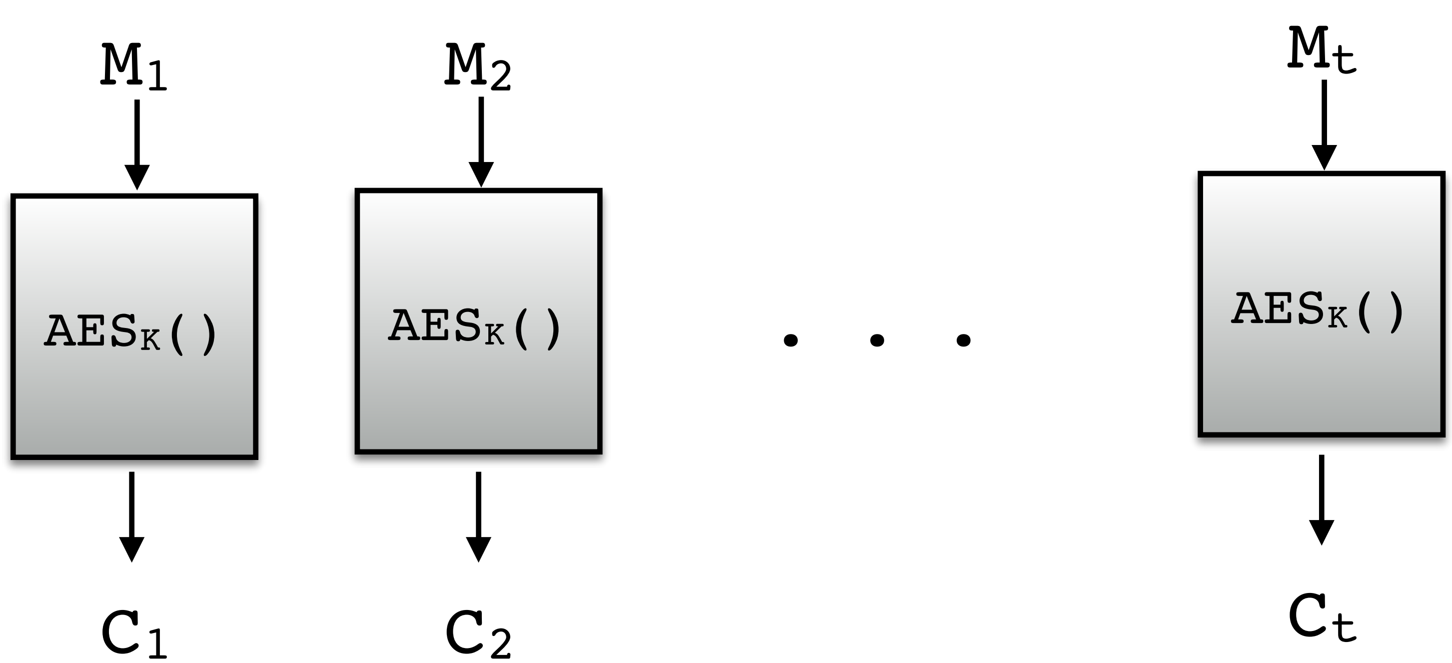
Encrypting large files: ECB

Warning: Broken

- ECB = "Electronic Code Book"

<u>AES-ECB$_k$(M)</u>

- Parse M into blocks $M_1, M_2, \ldots, M_t$
  // all blocks except M_t are 16 bytes
- Pad M_t up to 16 bytes
- For $i=1\ldots t$:
  - $C_i \leftarrow AES_k(M_i)$
- Return $C_1, \ldots, C_t$



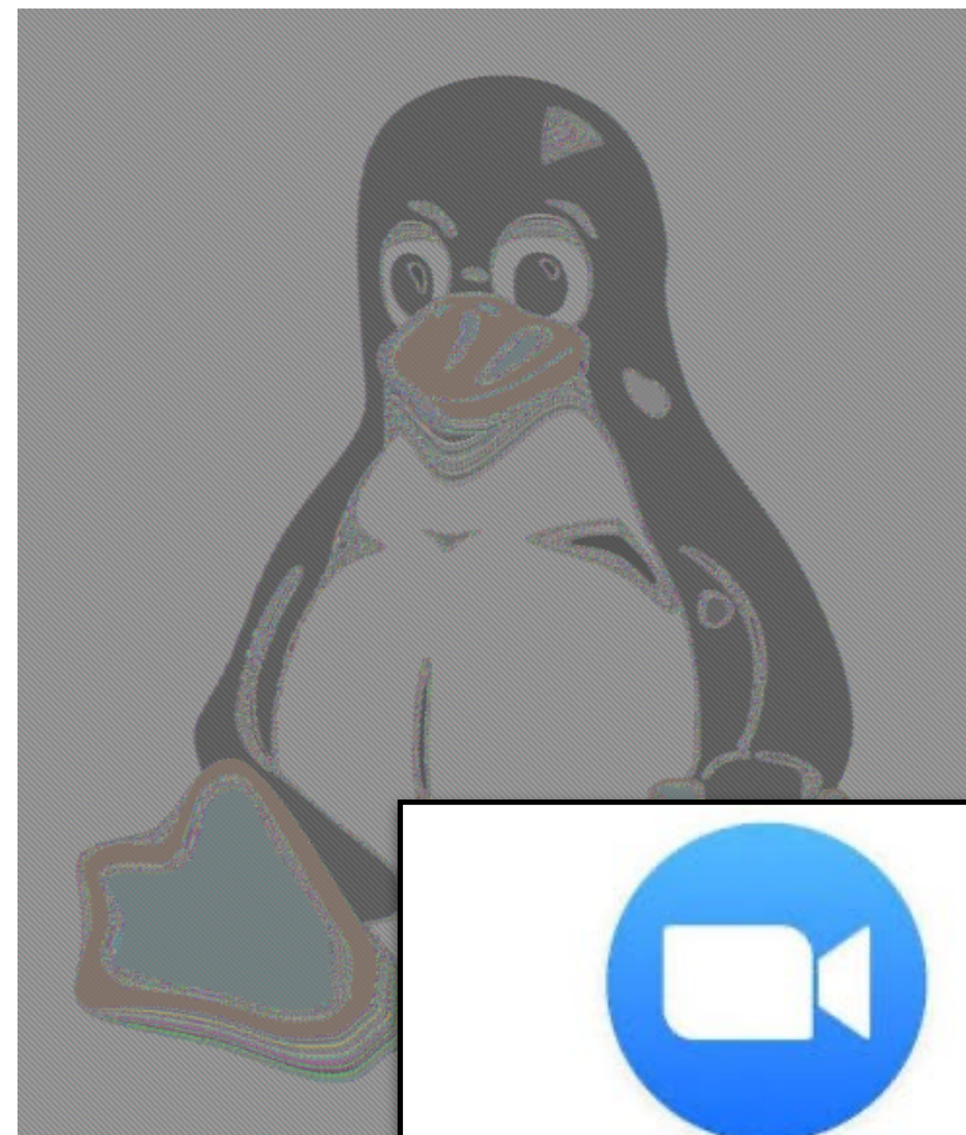
The ECB Penguin

Warning: Broken

- 16 byte chunks are consecutive pixels

Plaintext

ECB Ciphertext



- It gets even worse…

Encrypting large files, Attempt #2: CTR

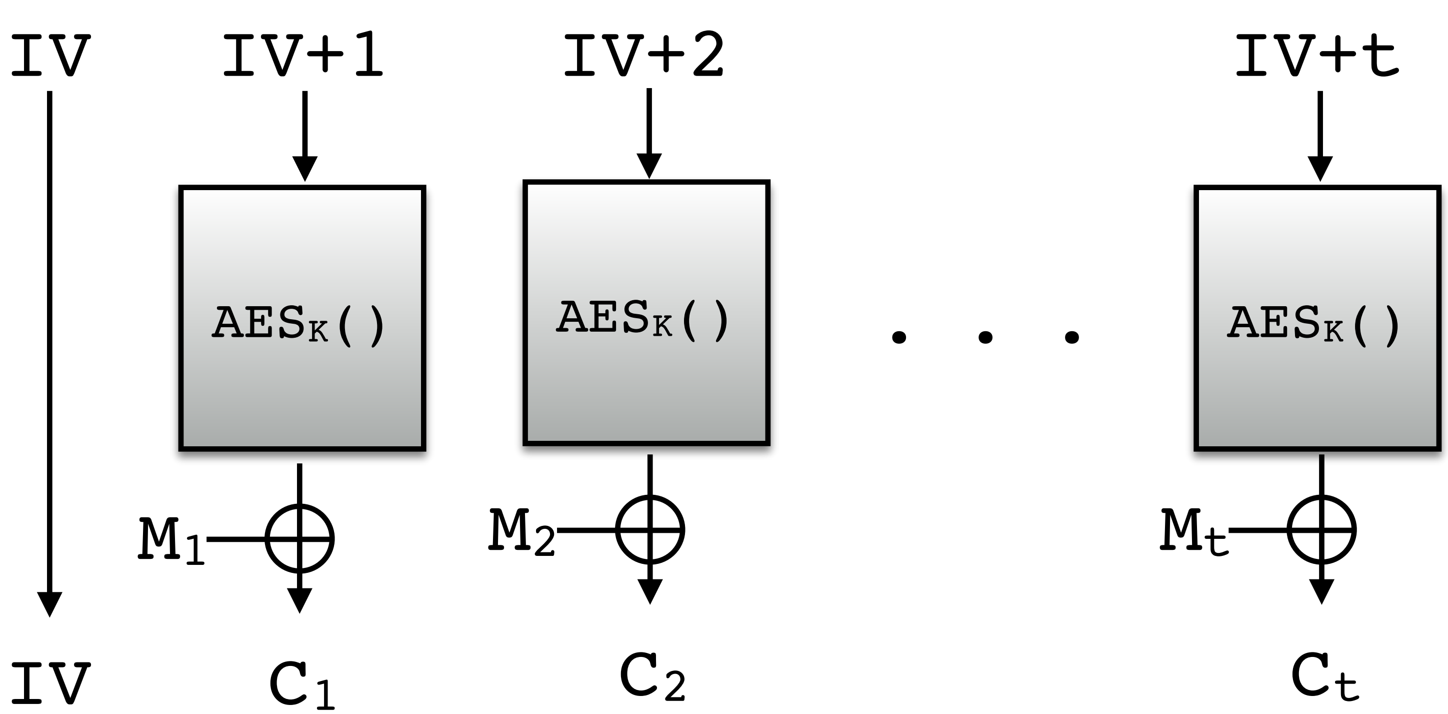
- CTR = “Counter Mode”
- Idea: Build a nonce-based stream cipher from AES

<u>$\texttt{AES-CTR}_{\texttt{k}}\texttt{(IV,M)}$</u>

- Parse $\texttt{M}$ into blocks $\texttt{M}_1$, $\texttt{M}_2$, …, $\texttt{M}_t$
  // all blocks except M_t are 16 bytes
- For $\texttt{i=1…t:}$
  - $\texttt{C}_i \leftarrow \texttt{M}_i \oplus \texttt{AES}_k\texttt{(IV+i)}$
- Return $\texttt{IV}$, $\texttt{C}_1$,…, $\texttt{C}_t$

Notes:

- No need to pad last block
- Must avoid reusing part of stream

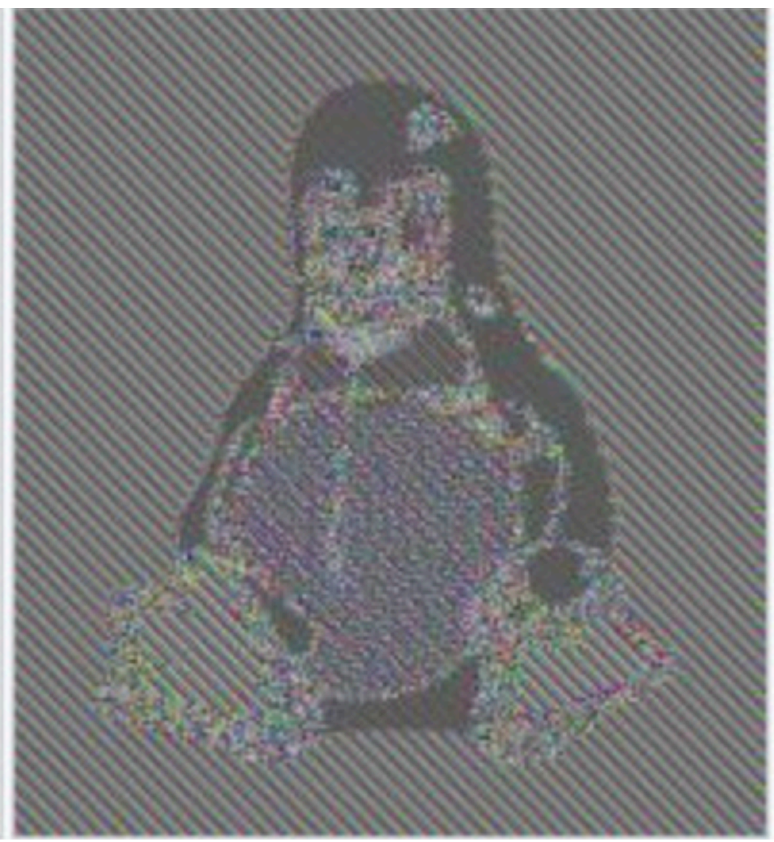


<u>When combined with authentication</u>, CTR is a good cipher.

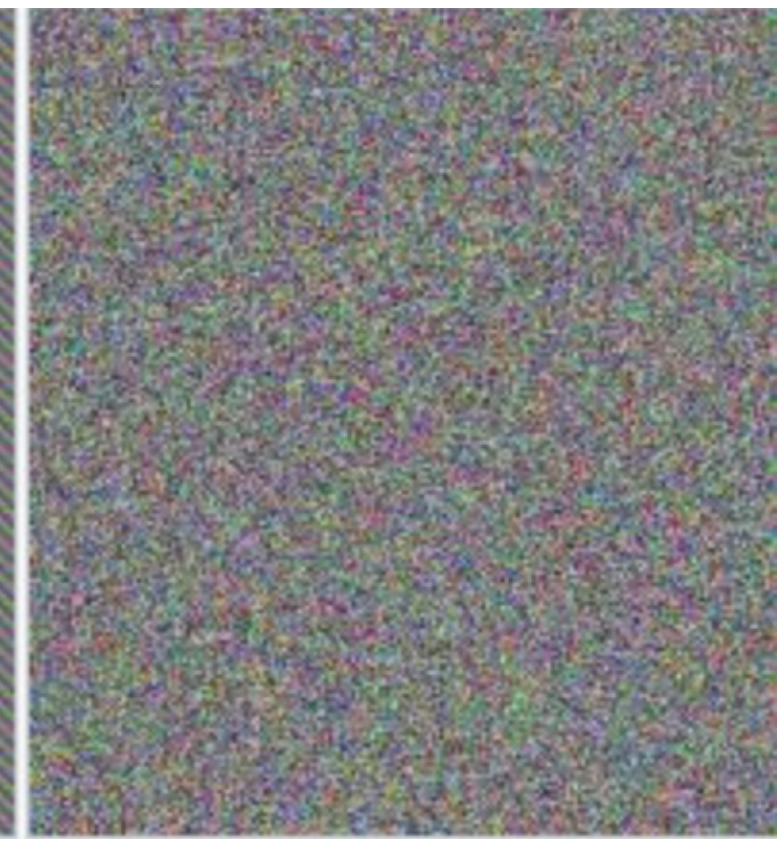
Penguin Sanity Check

Plaintext

ECB Ciphertext



CTR Ciphertext



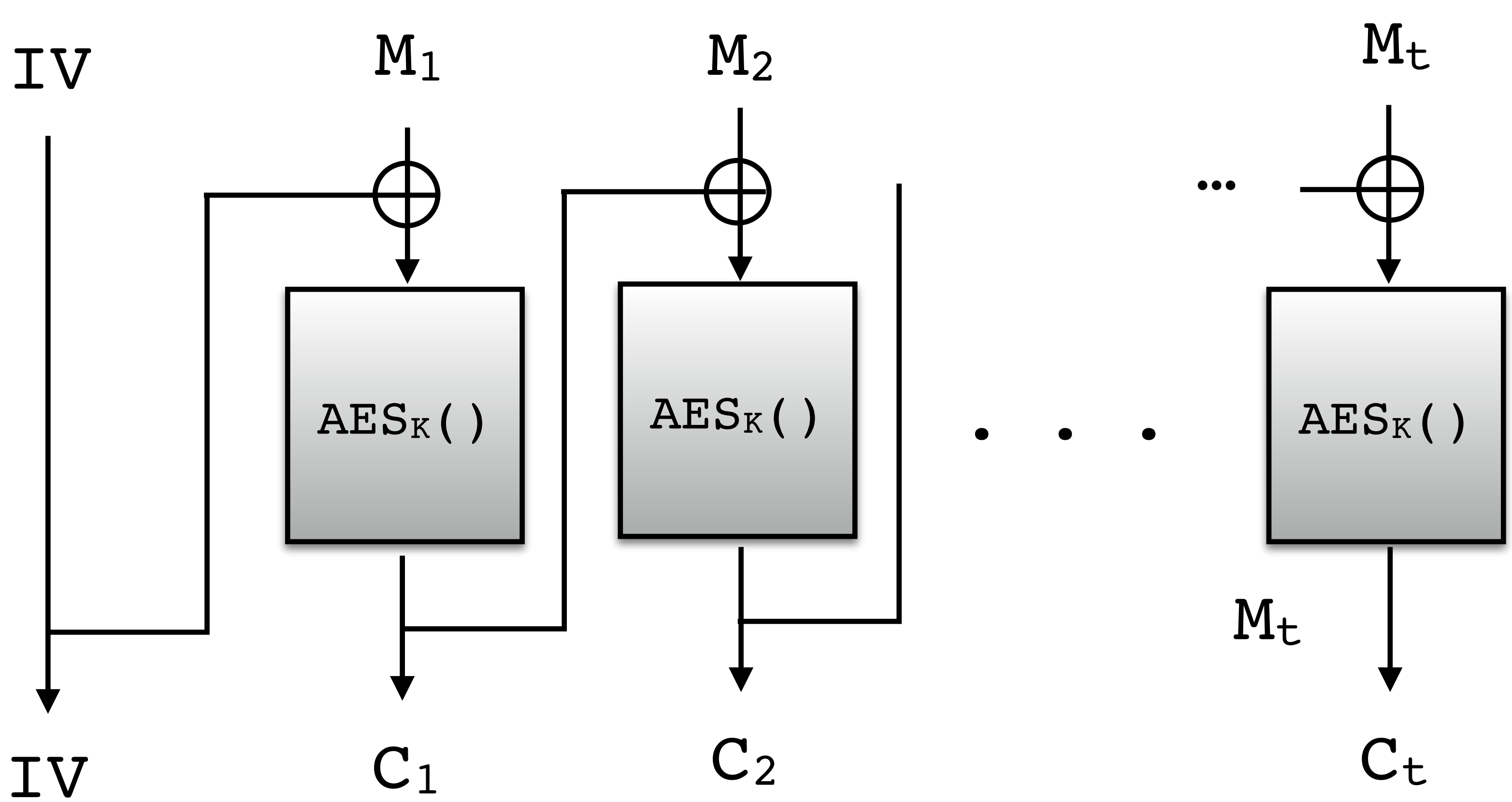
Looks random

Encrypting large files, Attempt #3: CBC

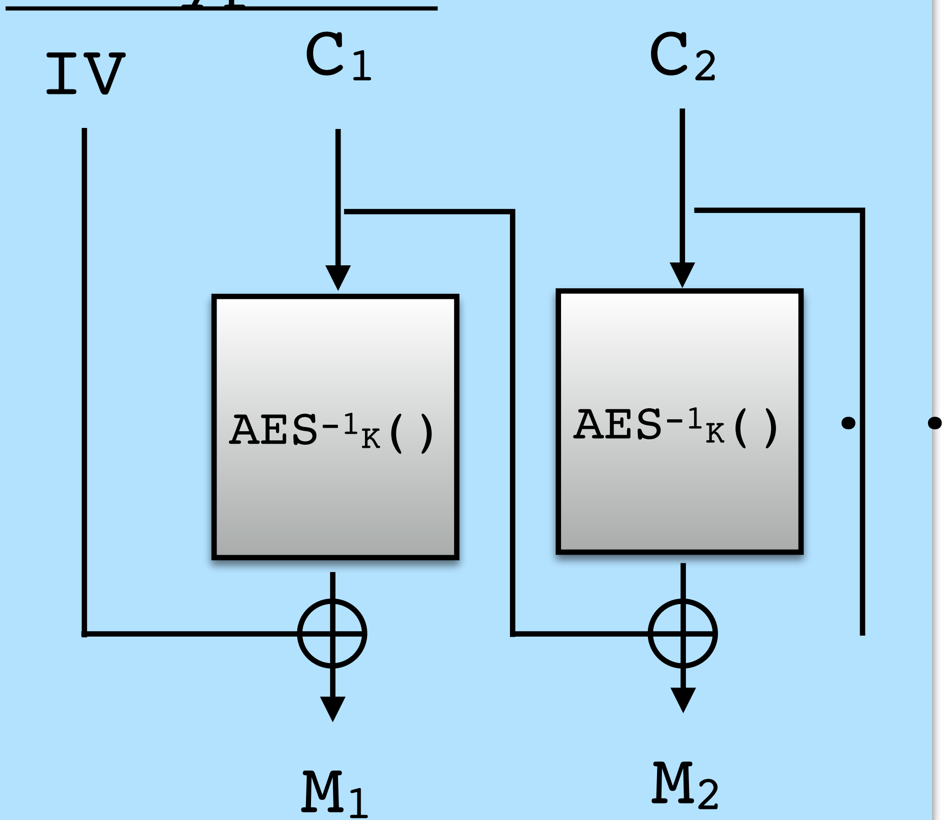
- CBC = “Cipher Block Chaining”
- Nonce-based, but not a stream cipher
- Historical option (sometimes used without nonce)

$\underline{\texttt{AES-CBC}_k\texttt{(IV,M)}}$

- Parse $\texttt{M}$ into blocks $\texttt{M}_1, \texttt{M}_2, \ldots, \texttt{M}_t$
  // all blocks except $\texttt{M}_t$ are 16 bytes
- Pad $\texttt{M}_t$ up to 16 bytes
- $\texttt{C}_0 \leftarrow \texttt{IV}$
- For $\texttt{i=1...t:}$
  - $\texttt{C}_i \leftarrow \texttt{AES}_k(\texttt{M}_i \oplus \texttt{C}_{i-1})$
- Return $\texttt{C}_0, \texttt{C}_1, \ldots, \texttt{C}_t$



Decryption

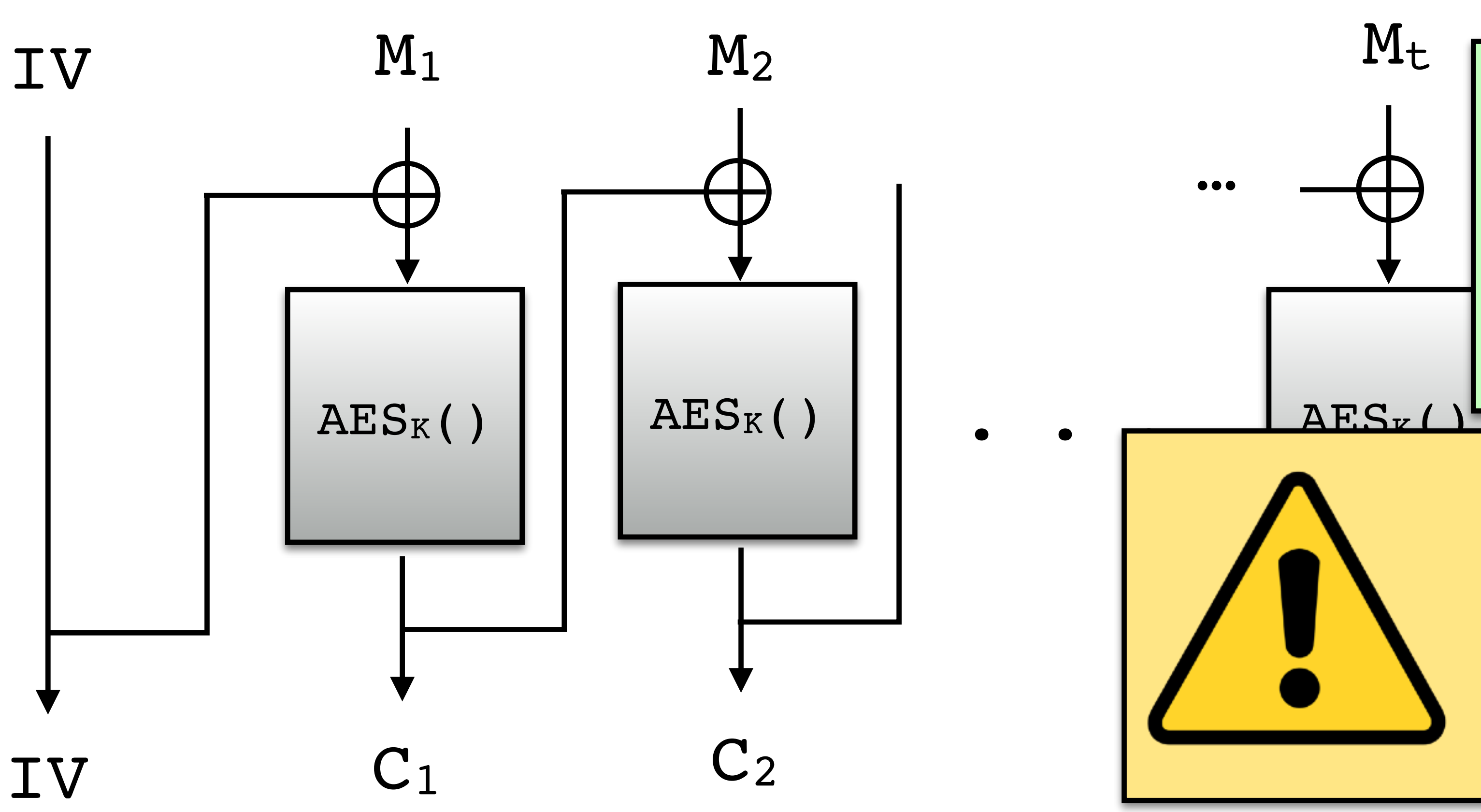


Encrypting large files, Attempt #3: CBC

- CBC = “Cipher Block Chaining”
- Nonce-based, but not a stream cipher
- Historical option (sometimes used without nonce)

<u>$\texttt{AES-CBC}_\texttt{k}\texttt{(IV,M)}$</u>

- Parse $\texttt{M}$ into blocks $\texttt{M}_1, \texttt{M}_2, \ldots, \texttt{M}_t$
  // all blocks except $\texttt{M}_t$ are 16 bytes
- Pad $\texttt{M}_t$ up to 16 bytes
- $\texttt{C}_0 \leftarrow \texttt{IV}$
- For $\texttt{i=1…t:}$
  - $\texttt{C}_\texttt{i} \leftarrow \texttt{AES}_\texttt{k}(\texttt{M}_\texttt{i} \oplus \texttt{C}_{\texttt{i-1}})$
- Return $\texttt{C}_0, \texttt{C}_1, \ldots, \texttt{C}_t$



<u>When combined with authentication</u>, CBC is a good cipher.

Warning: Padding creates havoc with authentication. Very difficult to implement.

Blockcipher Encryption Summary

- AES is unbroken
- AES-CTR is most robust construction for confidentiality
- AES-CTR/AES-CBC do not provide authenticity/integrity and should almost never be used alone.

The End