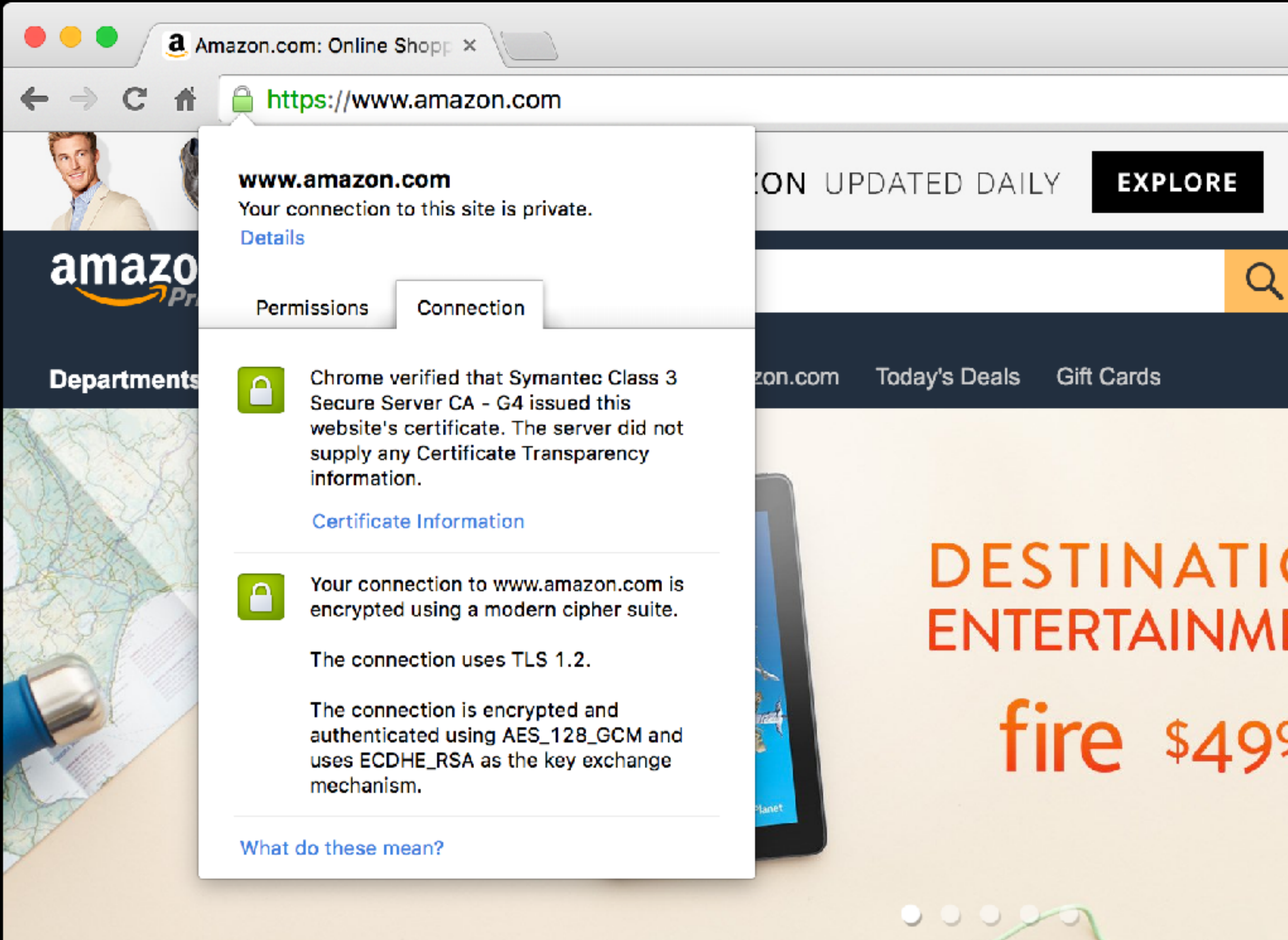
An Introduction to Cryptography

CMSC 23200/33250, Winter 2020, Lecture 3

David Cash & Blase Ur

University of Chicago



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
Today's Deals
Gift Cards


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

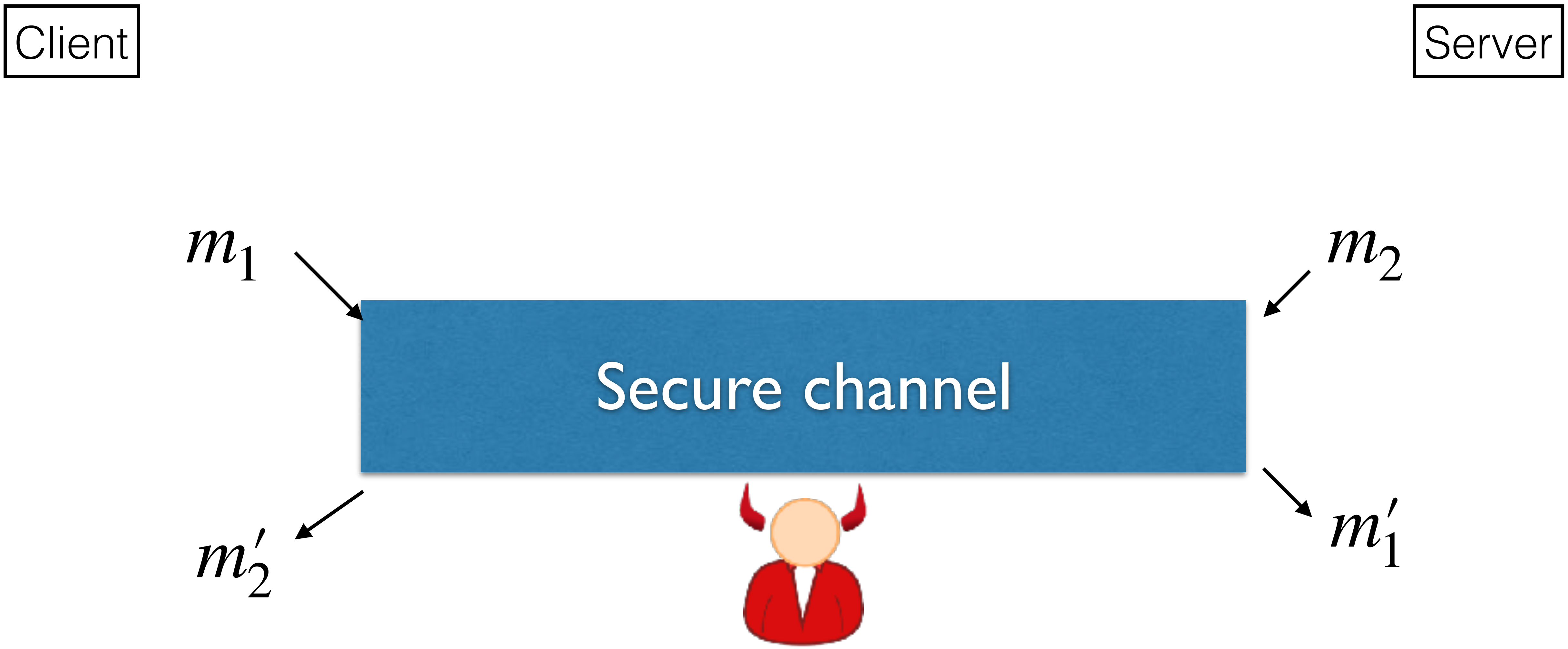

5100
21
5100

What is Cryptography?

Cryptography involves algorithms with security goals.

Cryptography involves using math to stop adversaries.

Common Security Goal: Secure Channel

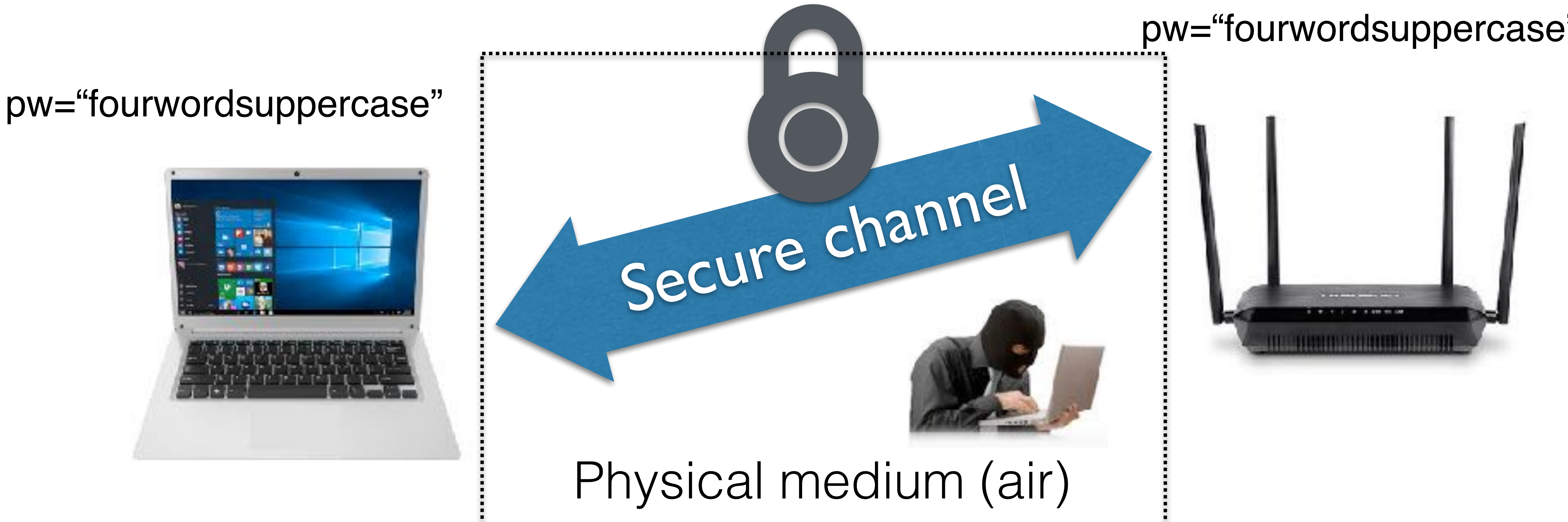


Confidentiality: Adversary does not learn anything about messages m_1, m_2

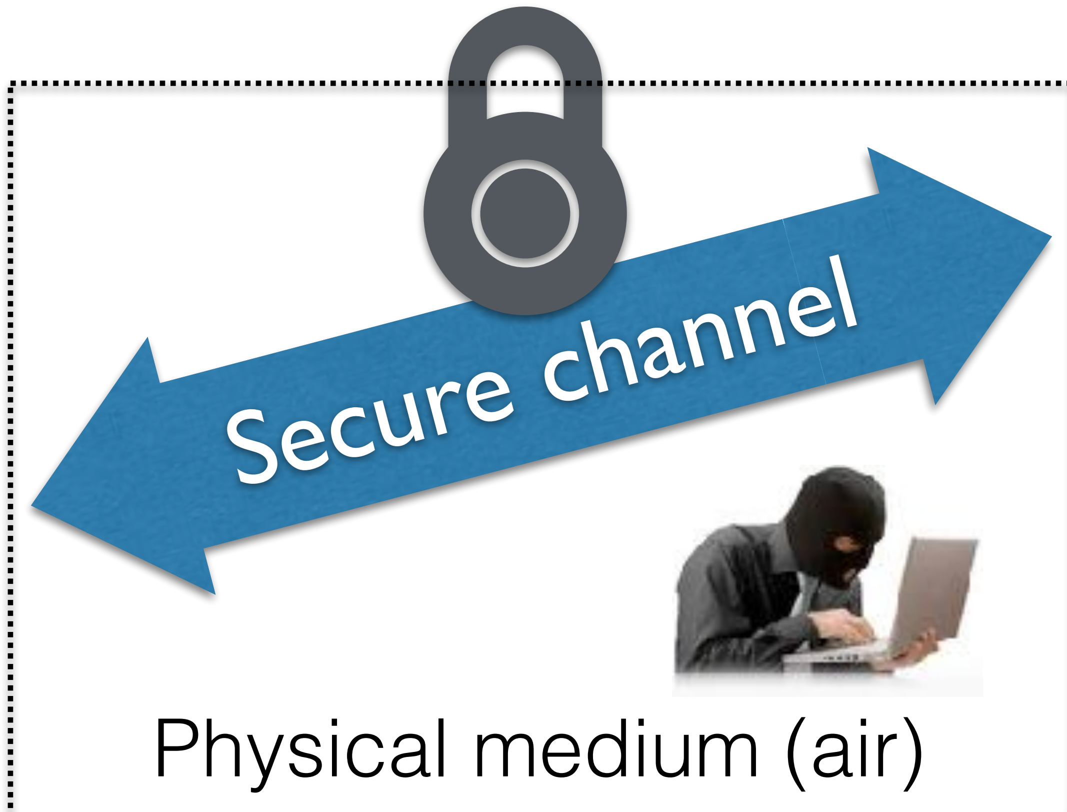
Authenticity: $m_1' = m_1$ and $m_2' = m_2$

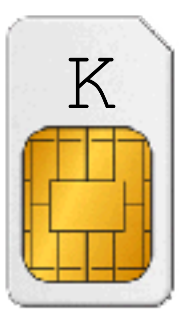
Warning: subtitles abound

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



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

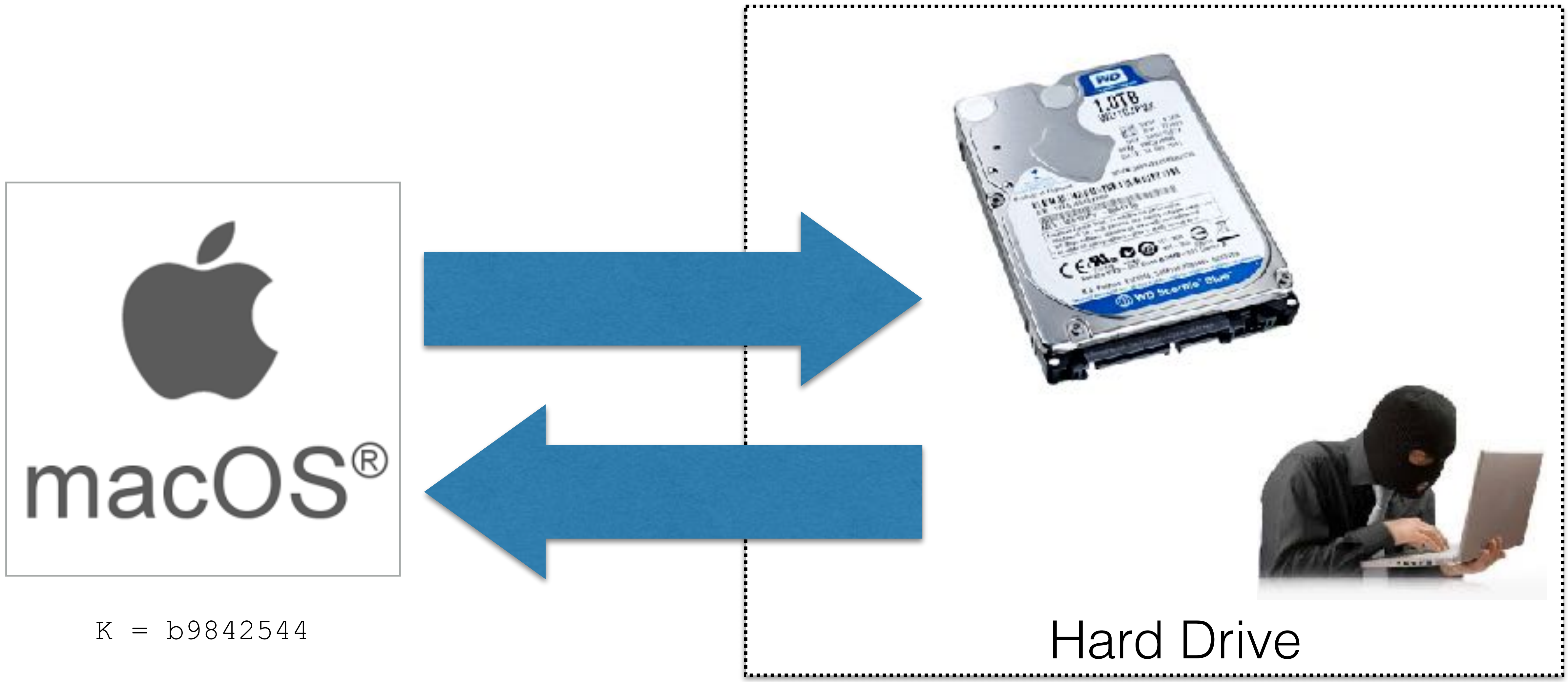




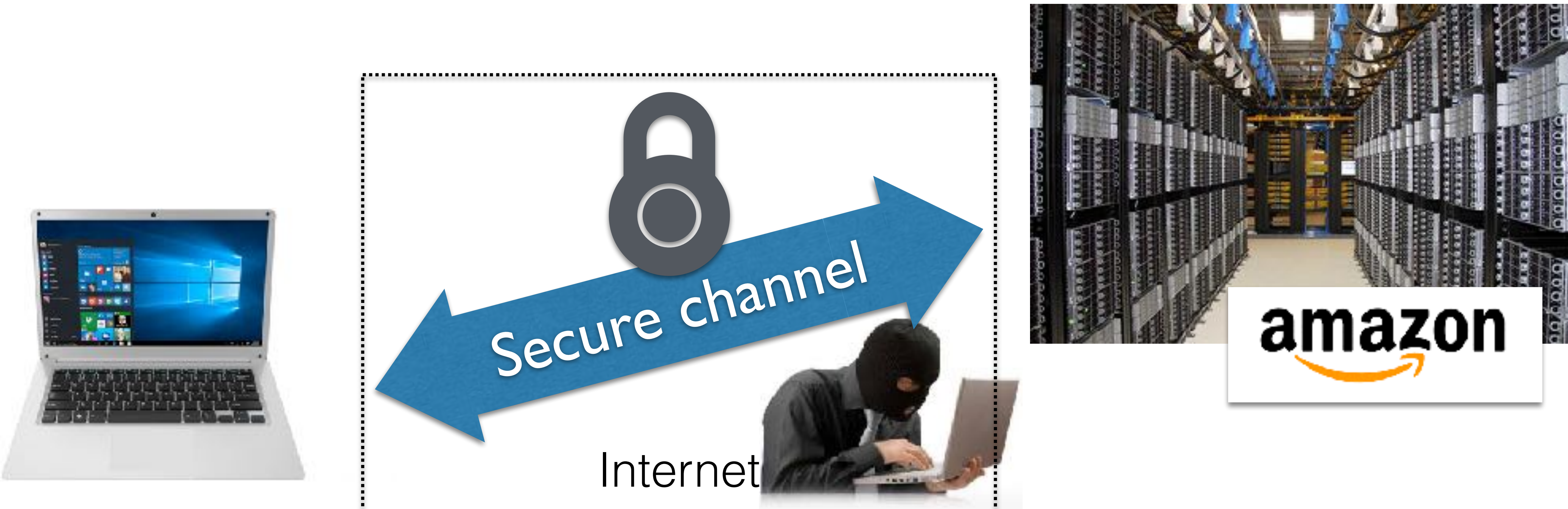
K = b9842544

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

Disk Encryption

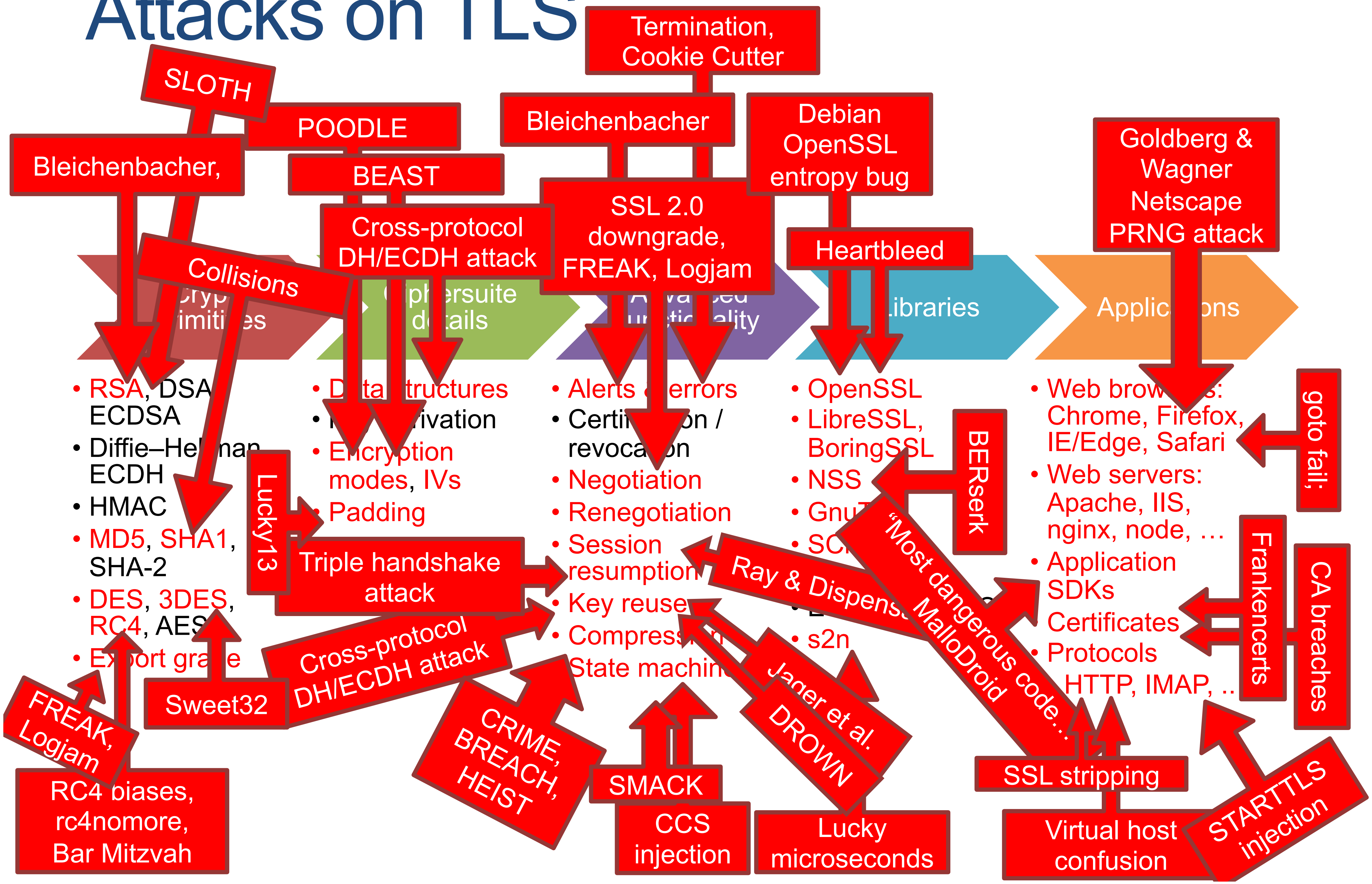


Crypto in your browser: TLS (Transport Layer Security)



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

Attacks on TLS



Rest of this lecture

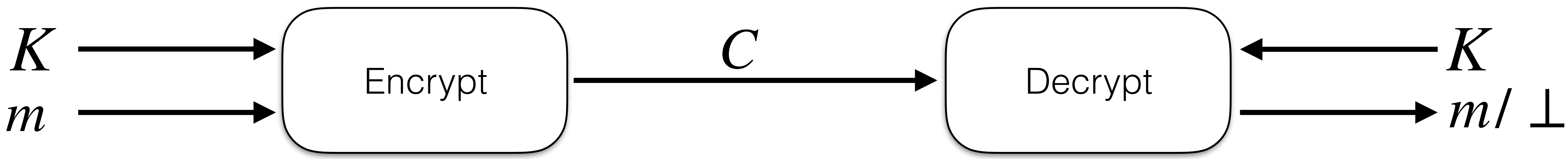
- Syntax of a cipher
- Some historical ciphers and how they were broken
- The One-Time Pad cipher and its security/insecurity
- Towards practice: Begin stream ciphers and blockciphers

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

Ciphers (a.k.a. Symmetric Encryption)

A <u>cipher</u> 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**

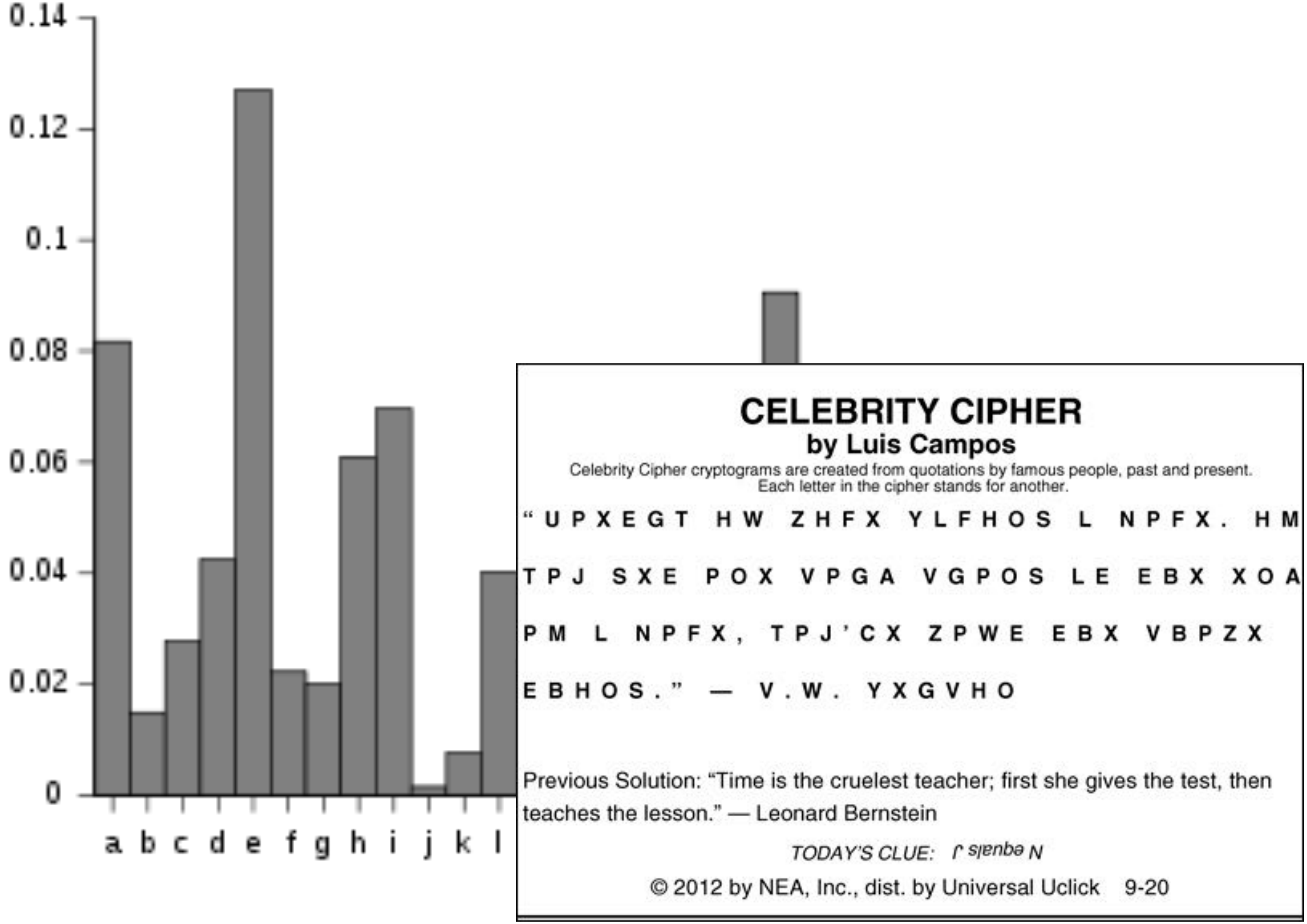
x	π(x)
A	Z
B	U
C	A
D	Y
E	R
F	E
G	X
H	B
I	D
J	C
K	M
L	Q
M	H
N	T
O	I
P	S
Q	V
R	N
S	P
T	K
U	O
V	F
W	G
X	W
Y	L
Z	J

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: J equals N

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

<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?

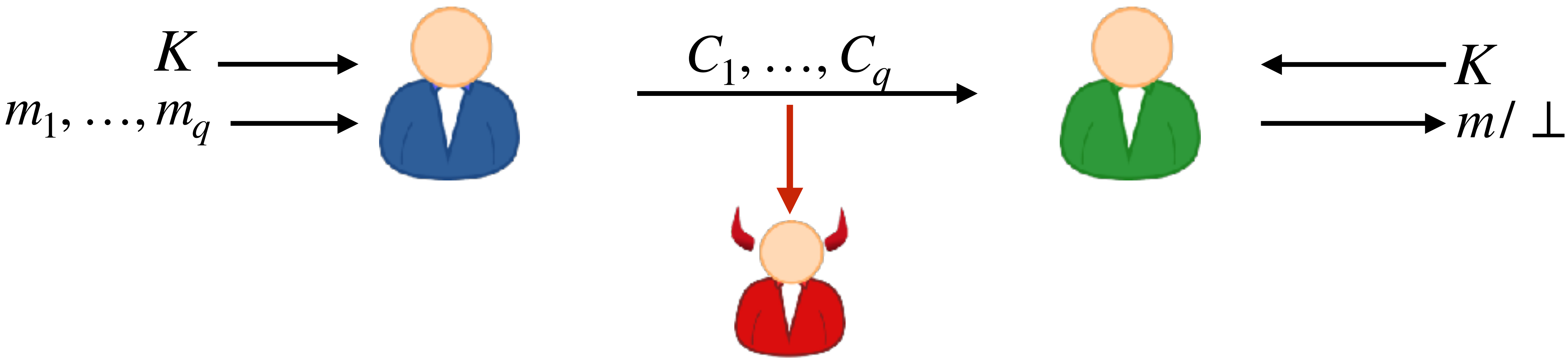
2. Quantify adversary capabilities

   View ciphertexts? Probe system with chosen inputs?

3. Quantify computational resources available to adversary

   Compute cycles? Memory?

Breaking Encryption - A Basic Game

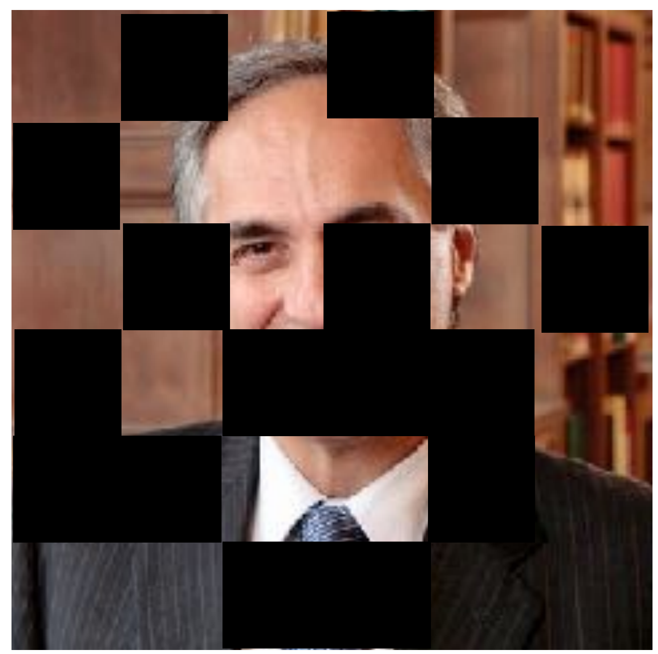


Ciphertext-only attack: The adversary sees ciphertexts and attempts to recover some useful information about plaintexts.

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.

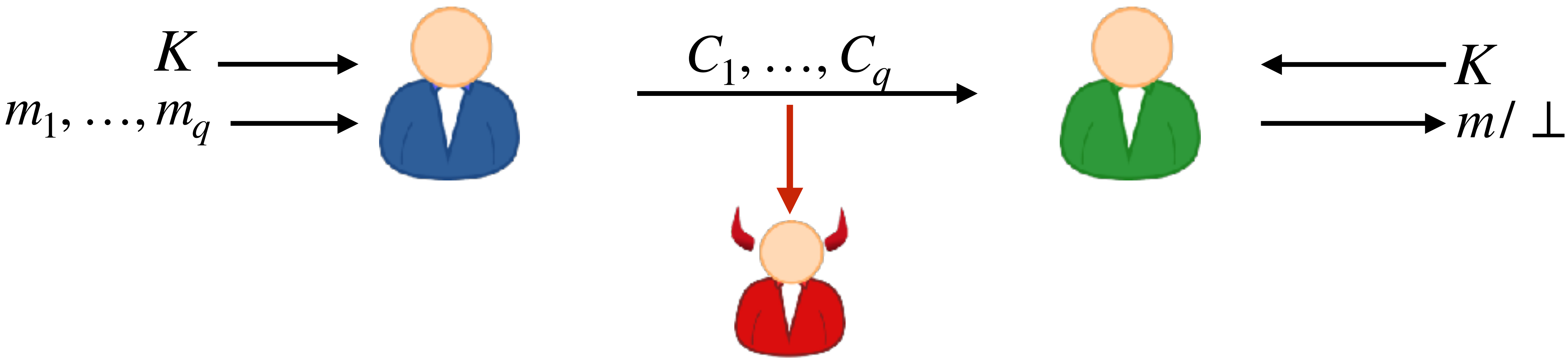
```
M = http://site.com?password=[redacted]
```

“Attacks” versus “Security”

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

Encryption should hide <u>all possible partial information</u> 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

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

Ciphertext observed: `10111`
Possible plaintext: `00101`
$\Rightarrow$ 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

$$\texttt{HELLOALICE} \oplus \texttt{Pad} = C_1 \qquad \texttt{PWDHAMSTER} \oplus \texttt{Pad} = C_2$$

$$C_1 \oplus \texttt{HELLOALICE} = \texttt{Pad} \qquad C_2 \oplus \texttt{Pad} = \texttt{PWDHAMSTER}$$

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)
- Secure routers caught doing this last fall! [link]

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

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

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

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 $\geq$ Plaintext-length

In practice: (covered here and next lecture):

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

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 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?

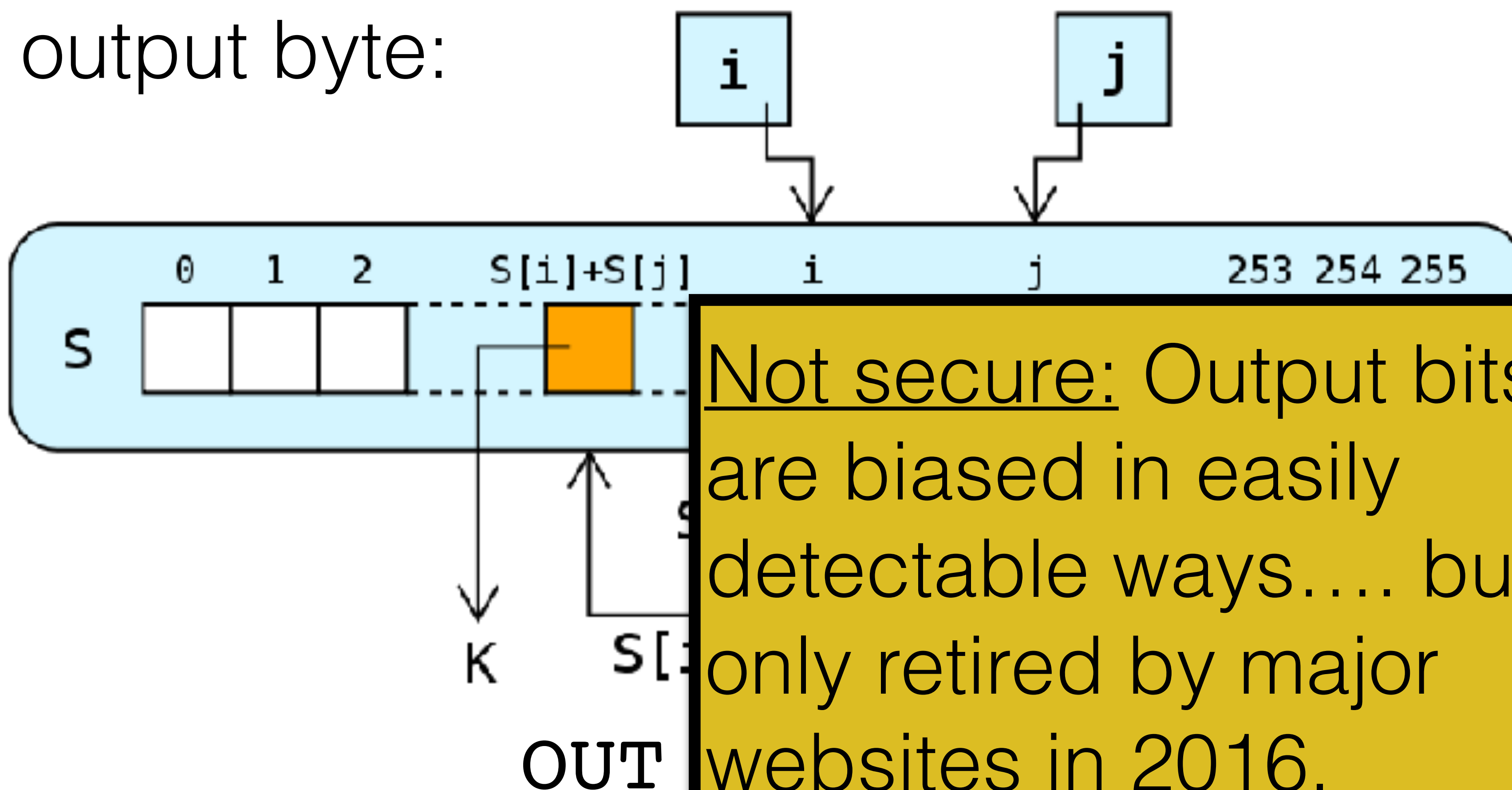
*Not directly comparable but this is an estimate of equivalent power. Quantum computers are most effective against public-key crypto, but they also speed up attacks on symmeric-key crypto. (More next week.)

Example Stream Cipher: RC4

Warning: Broken

<u>Internal state:</u> Array `S` of 256 bytes and ptrs `i, j`

To compute next output byte:



<u>Not secure:</u> Output bits are biased in easily detectable ways…. but only retired by major websites in 2016.

<u>Then:</u>

```
-i=i+1 mod256
-j=j+S[i] mod256
-swap S[i] and S[j]
```

<u>Replacement:</u> Salsa20/ChaCha, or AES-based methods to be discussed

Pad reuse can still happen with stream ciphers