CS 370: Introduction to Security

04.06: Cryptography basics

Tu/Th 4:00 – 5:50 PM (WNGR 149)

Sanghyun Hong

sanghyun.hong@oregonstate.edu

Oregon State University

SAIL

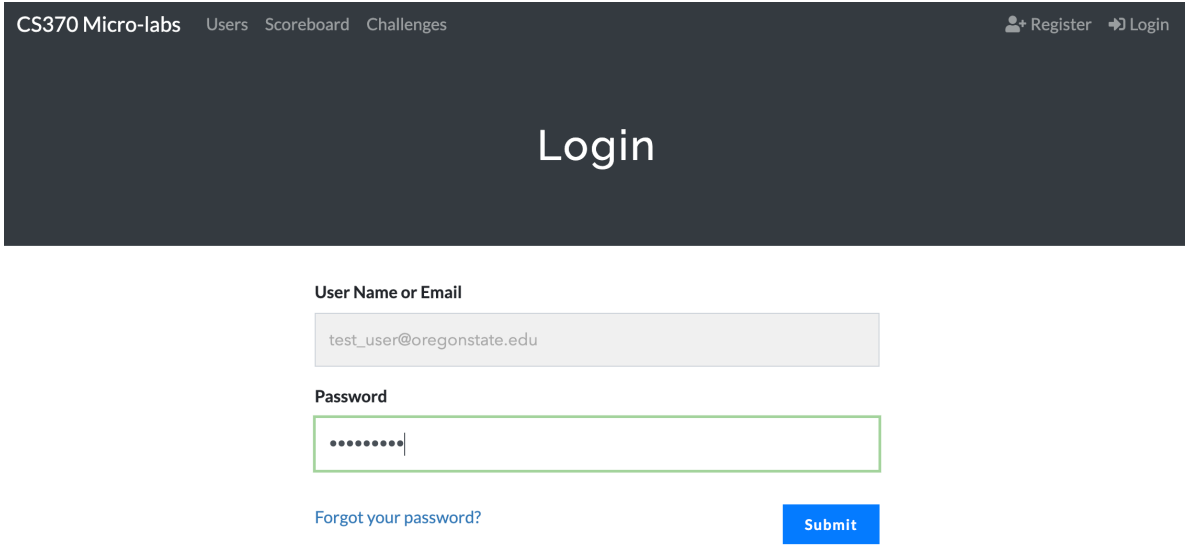
Secure AI Systems Lab

Topics for today

- Micro-lab instruction
- Crypto basics
  - Why do we need crypto?
  - What were the ancient crypto schemes?
  - What does it mean by perfectly secure?
  - What were the perfect crypto schemes so far?
  - What are the limitations of those above?
  - What were some practical solutions and their (also) limitations?

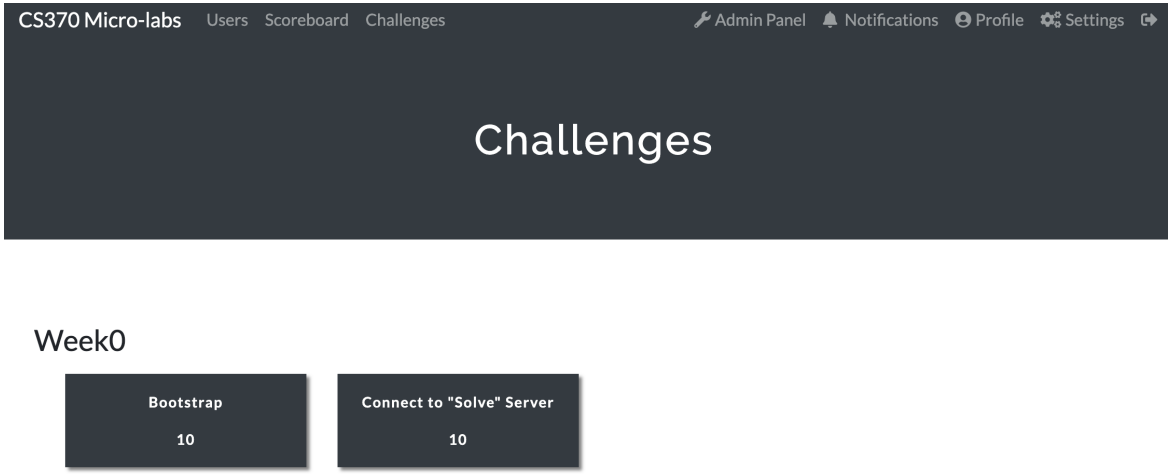
Micro-lab instruction

- Create an account on ctf.secure-ai.systems
  - Use OSU email address
  - Use some secure password
  - Once registered, check the inbox for the welcome email
  - Otherwise, lmk



Micro-lab instruction – cont'd

- Go to Challenges
  - You can find two challenges
  - More will come soon



Micro-lab instruction – cont'd

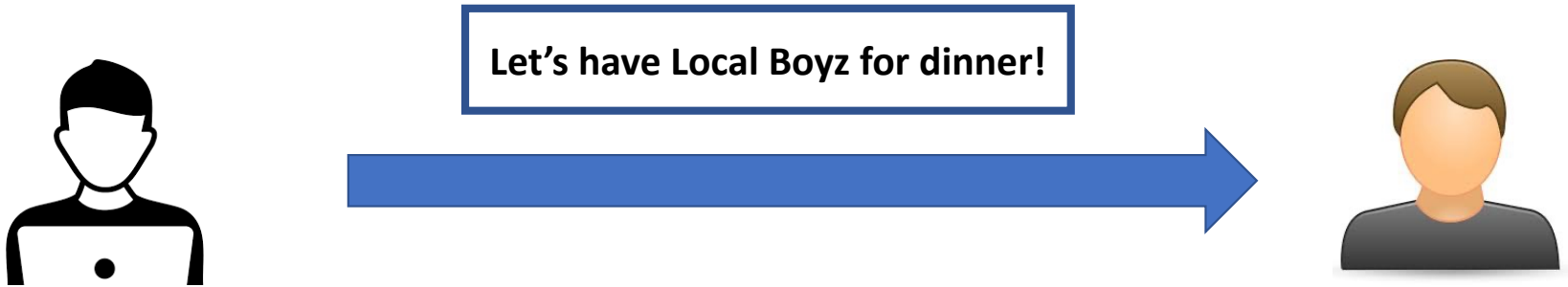
- Connect to the "Solve" server
  - This is the place where you can solve the challenges
  - Instruction can be found on here (the course website)
  - Your username is set to your ONID
  - Password for logging in can be found in the Canvas announcement
  - I will walk you through how to do it now…

Topics for today

- Micro-lab instruction
- Crypto basics
  - Why do we need crypto?
  - What were the ancient crypto schemes?
  - What does it mean by perfectly secure?
  - What were the perfect crypto schemes so far?
  - What are the limitations of those above?
  - What were some practical solutions and their (also) limitations?

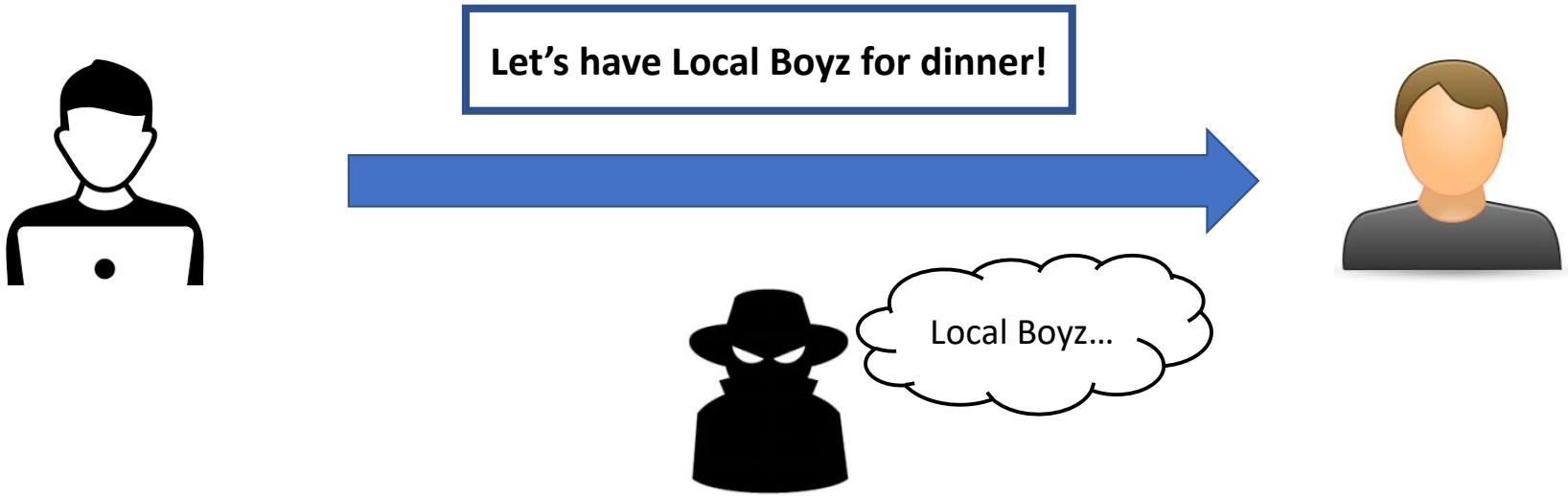
Why do we need crypto?

- Confidentiality
  - We want to communicate with others securely (and privately)



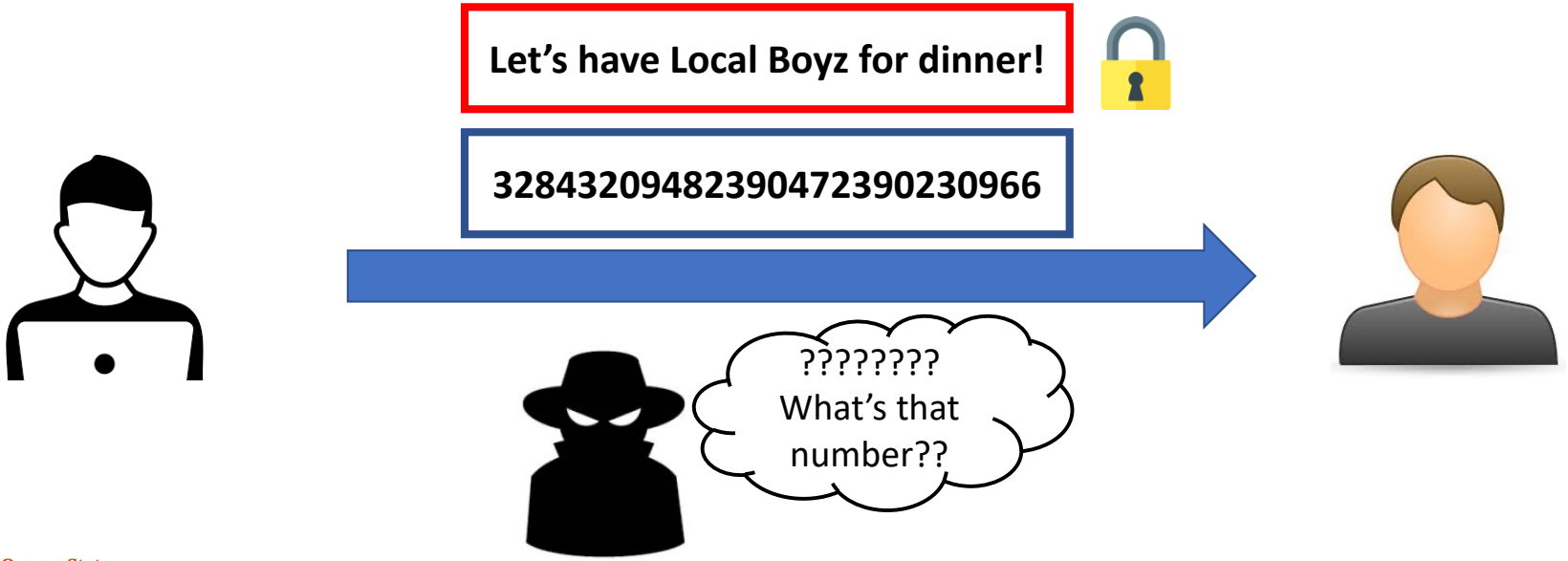
Why do we need crypto?

- Confidentiality
  - We want to communicate with others securely (and privately)
  - Plaintext communication can be eavesdropped by an adversary



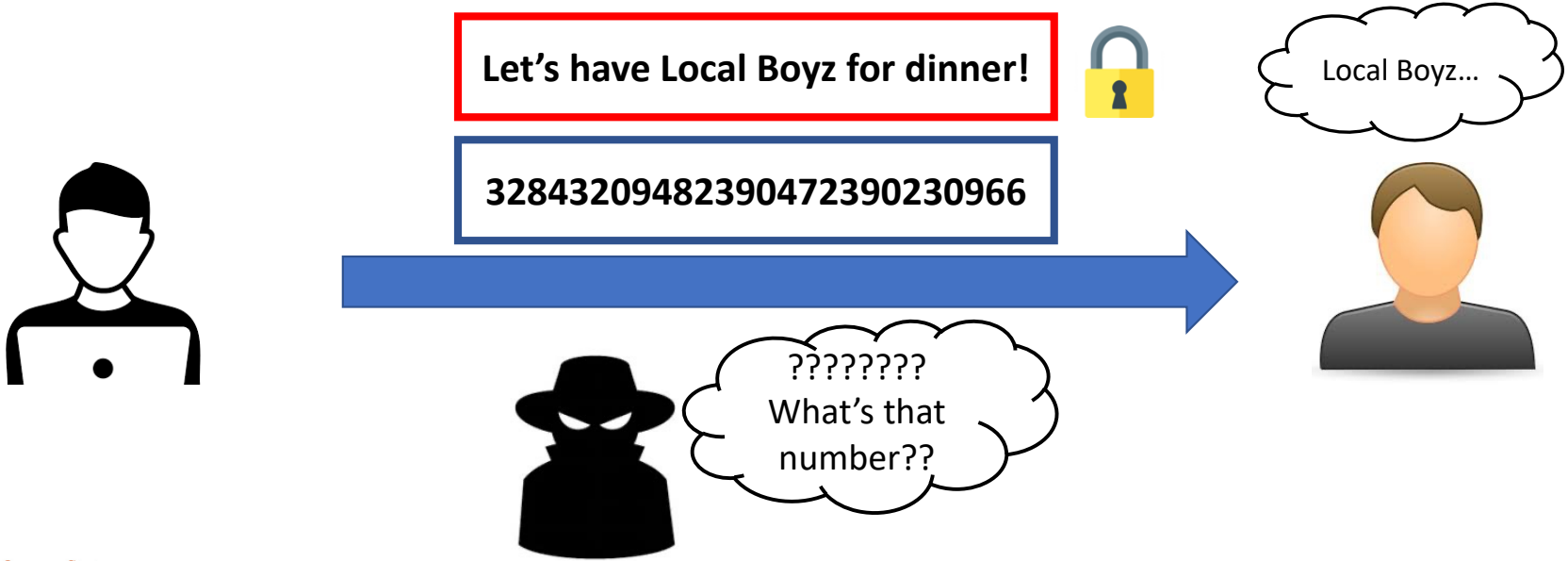
Why do we need crypto?

- Confidentiality
  - We want to communicate with others securely (and privately)
  - Plaintext communication can be eavesdropped by an adversary
  - Cryptography enables secure (and private) communication



Basic terminology

- Terms
  - Plaintext: readable text, before getting encrypted
  - Ciphertext: encrypted text, transformed plaintext using an encryption algorithm
  - Encryption/decryption: the act of encrypting (or decrypting)



Topics for today

- Micro-lab instruction
- Crypto basics
  - Why do we need crypto?
  - What were the ancient crypto schemes?
  - What does it mean by perfectly secure?
  - What were the perfect crypto schemes so far?
  - What are the limitations of those above?
  - What were some practical solutions and their (also) limitations?

Caesar cipher

- Crypto scheme in Roman empire
  - Encryption: shift each character by N
  - Example: shift by 3
    - Plaintext: ABCDEFGHIJKLMNOPQRSTUVWXYZ
    - Cipher text: DEFGHIJKLMNOPQRSTUVWXYZABC

    - Plaintext: HELLO
    - Cipher text: KHOOR

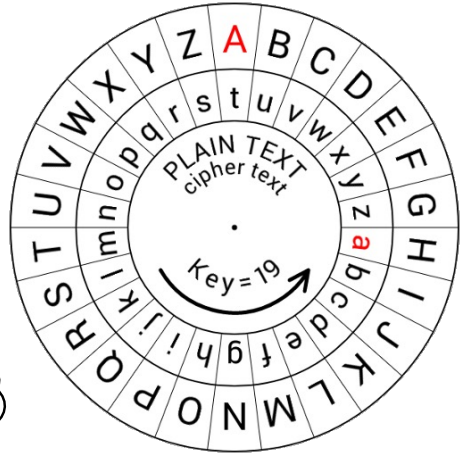
Are we safe now?

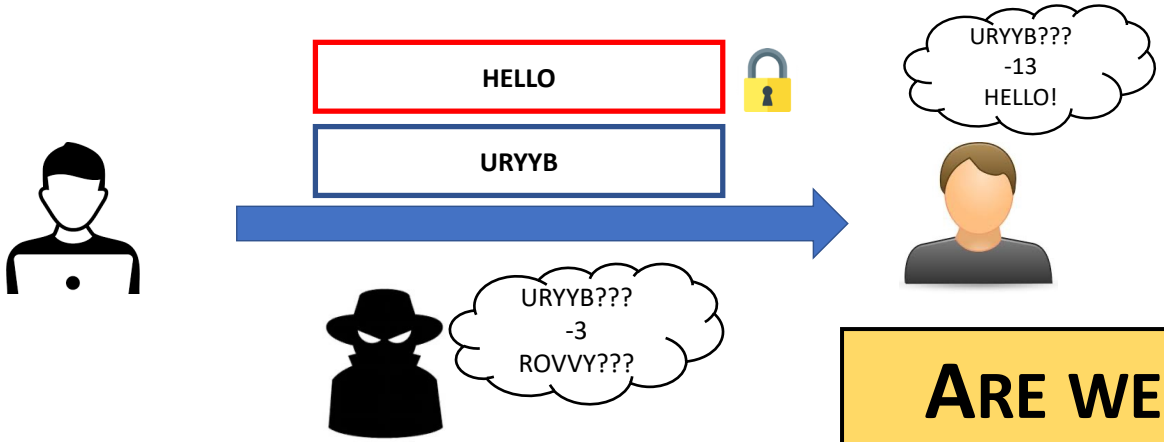
Problem(s) in Caesar Cipher

- What if:
  - An adversary knows the shift offset?

General form of Caesar cipher

- Rot-N cipher
  - Encryption: shift each character by N
  - More complex than Caesar cipher
    - We can set N and share between us
    - If an adversary doesn't know N, it's hard to decrypt it
  - Example: shift by 13



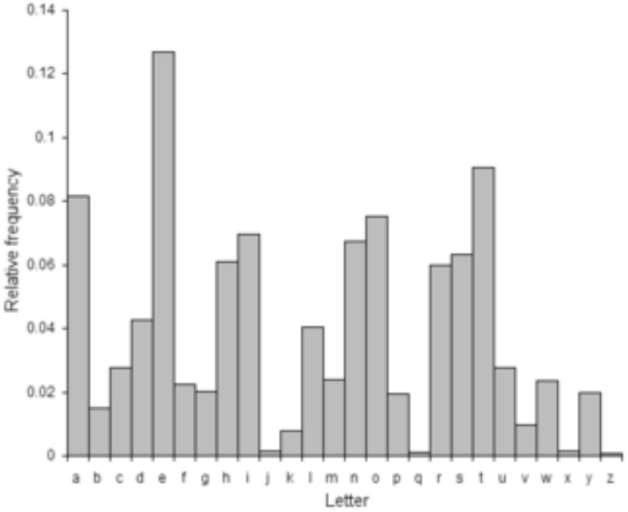


Are we safe now?

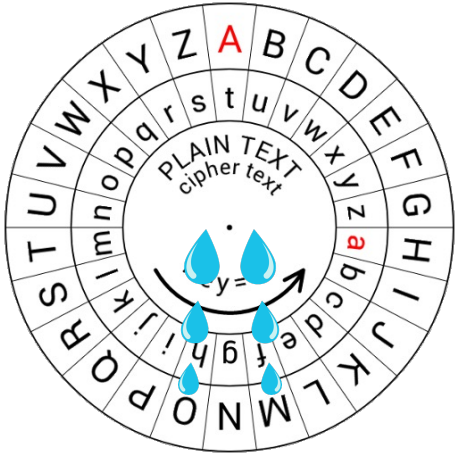
Problem(s) in ROT-N cipher

- What if:
  - ~~An adversary knows the shift offset?~~
  - An attacker finds the offset

E	11.1607%	56.88	M	3.0129%	15.36
A	8.4966%	43.31	H	3.0034%	15.31
R	7.5809%	38.64	G	2.4705%	12.59
I	7.5448%	38.45	B	2.0720%	10.56
O	7.1635%	36.51	F	1.8121%	9.24
T	6.9509%	35.43	Y	1.7779%	9.06
N	6.6544%	33.92	W	1.2899%	6.57
S	5.7351%	29.23	K	1.1016%	5.61
L	5.4893%	27.98	V	1.0074%	5.13
C	4.5388%	23.13	X	0.2902%	1.48
U	3.6308%	18.51	Z	0.2722%	1.39
D	3.3844%	17.25	J	0.1965%	1.00
P	3.1671%	16.14	Q	0.1962%	(1)



Letter frequency in English

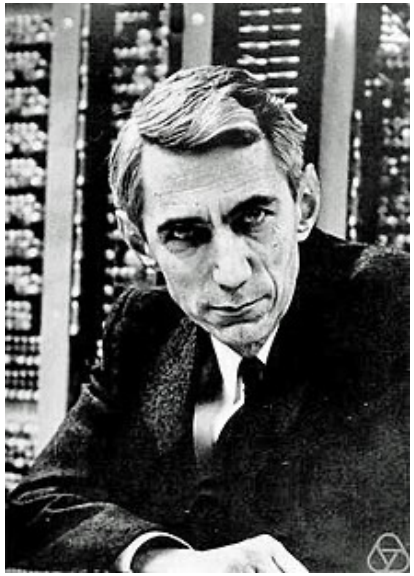


Topics for today

- Micro-lab instruction
- Crypto basics
  - Why do we need crypto?
  - What were the ancient crypto schemes?
  - What does it mean by perfectly secure?
  - What were the perfect crypto schemes so far?
  - What are the limitations of those above?
  - What were some practical solutions and their (also) limitations?

Perfect security

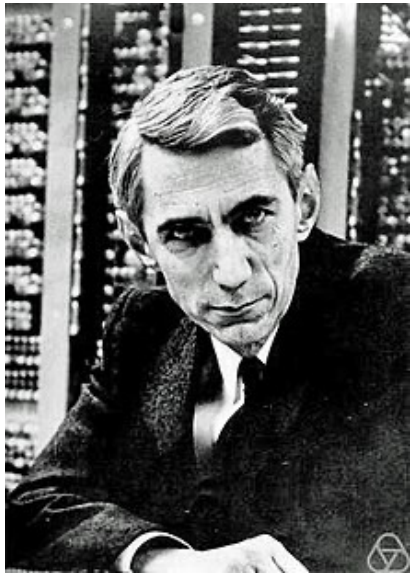
- Shannon's intuition
  - An adversary should not distinguish a message M from a random text R



Claude Shannon (1916 ~ 2001)
A Father of Information Theory
and Modern Cryptography

Perfect security

- Shannon's intuition
  - An adversary should not distinguish a message M from a random text R
  - Formally:
    - Pr[M = m | C = c] = Pr[M = m]
    - where
      - m is a message (from a set M)
      - c is a ciphertext (from a set of all ciphertexts C)
    - Pr[C = c | M = m] = Pr[C = c]
  - It means:
    - Ciphertext provides no additional information
    - Observing c does not help with guessing M = m
    - c is independent of the message m



Claude Shannon (1916 ~ 2001)
A Father of Information Theory and Modern Cryptography

Topics for today

- Micro-lab instruction
- Crypto basics
  - Why do we need crypto?
  - What were the ancient crypto schemes?
  - What does it mean by perfectly secure?
  - What were the perfect crypto schemes so far?
  - What are the limitations of those above?
  - What were some practical solutions and their (also) limitations?

XOR CIPHER

- Crypto scheme with perfect secrecy
  - Encryption:
    - Given a message *m* and a random key *k*
    - Ciphertext $c = m \oplus k$
  - Example:
    - Message: HELLO
    - Key : ABCDE

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

Message	H (0x48)	E (0x45)	L (0x4c)	L (0x4c)	O (0x4f)
Key	A (0x41)	B (0x42)	C (0x43)	D (0x44)	E (0x45)
Ciphertext	**0x9**	**0x7**	**0xf**	**0x8**	**0xa**

XOR CIPHER

- Crypto scheme with perfect secrecy
  - Encryption:
    - Given a message *m* and a random key *k*
    - Plaintext $m = k \oplus c$
  - Example:
    - Message: HELLO
    - Key : ABCDE

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

Key	A (0x41)	B (0x42)	C (0x43)	D (0x44)	E (0x45)
Ciphertext	0x9	0x7	0xf	0x8	0xa
Decrypt	**H**	**E**	**L**	**L**	**O**

XOR CIPHER: IN BITWISE OPERATION

- Example from Wikipedia[1]

The string "Wiki" (01010111 01101001 01101011 01101001 in 8-bit ASCII) can be encrypted with the repeating key 11110011 as follows:

```
  01010111 01101001 01101011 01101001
⊕ 11110011 11110011 11110011 11110011
------------------------------------
= 10100100 10011010 10011000 10011010
```

And conversely, for decryption:

```
  10100100 10011010 10011000 10011010
⊕ 11110011 11110011 11110011 11110011
------------------------------------
= 01010111 01101001 01101011 01101001
```

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

[1]Image from: https://en.wikipedia.org/wiki/XOR_cipher

Topics for today

- Micro-lab instruction
- Crypto basics
  - Why do we need crypto?
  - What were the ancient crypto schemes?
  - What does it mean by perfectly secure?
  - What were the perfect crypto schemes so far?
  - What are the limitations of those above?
  - What were some practical solutions and their (also) limitations?

Problem(s) in XOR cipher

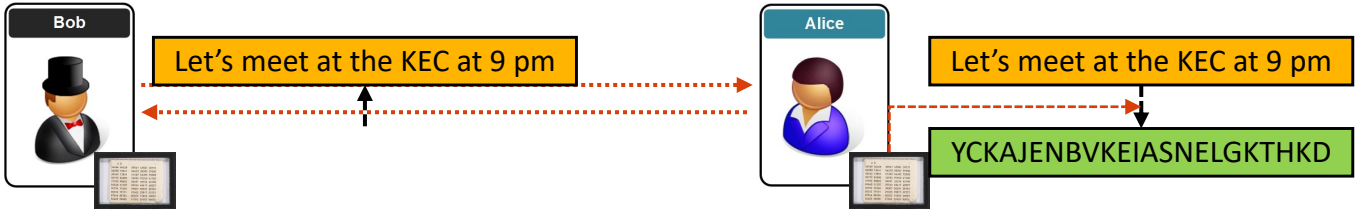
- What if:
  - An adversary accidently knows a pair of *m* and *c*
    - Key $k = m \oplus c$

Message	H (0x48)	E (0x45)	L (0x4c)	L (0x4c)	O (0x4f)
Ciphertext	0x9	0x7	0xf	0x8	0xa
Key	**A (0x41)**	**B (0x42)**	**C (0x43)**	**D (0x44)**	**E (0x45)**

General form of XOR cipher

- OTP
  - **O**ne-**T**ime **P**ads (**OTP**) is an encryption mechanism
- How it works?
  - Alice and Bob want to communicate *securely*
  - Alice and Bob share the same OTP
  - Alice encrypts a message to send with the OTP
  - Alice sends the encrypted message to Bob
  - Bob decrypts the received message with the OTP

An Example OTP



Problem(s) in XOR cipher

- What if:
  - An adversary accidently knows a pair of *m* and *c*
    - Key $k = m \oplus c$

Message	H (0x48)	E (0x45)	L (0x4c)	L (0x4c)	O (0x4f)
Ciphertext	0x9	0x7	0xf	0x8	0xa
Key	**A (0x41)**	**B (0x42)**	**C (0x43)**	**D (0x44)**	**E (0x45)**

- Practical limitations:
  - What if we want to encrypt a 1GB video file?
  - How can we share keys with others (OTP)?

Topics for today

- Micro-lab instruction
- Crypto basics
  - Why do we need crypto?
  - What were the ancient crypto schemes?
  - What does it mean by perfectly secure?
  - What were the perfect crypto schemes so far?
  - What are the limitations of those above?
  - What were some practical solutions and their (also) limitations?

Stream cipher

- Reduce key generation and exchange overheads
  - Encryption:
    - Given a message *m* and a random key *k*
    - Ciphertext $c = m \oplus k$
    - and:
      - The key stream is generated by the same mechanism for a sender and a receiver
      - The key stream is a byte stream (0xAB129dB…)
      - It performs XOR encryption over this byte stream

Stream cipher

- Stream cipher
  - Example:

Encrypt message 1 with 0x1b395a46
Encrypt message 2 with 0xf1737202
Encrypt message 3 with 0xccf0de05…

1: 0x1b395a46
2: 0xf1737202
3: 0xccf0de05
4: 0x908b0feb
5: 0x9d4c9add
…

A random number generator

Decrypt message 1 with 0x1b395a46
Decrypt message 2 with 0xf1737202
Decrypt message 3 with 0xccf0de05…

1: 0x1b395a46
2: 0xf1737202
3: 0xccf0de05
4: 0x908b0feb
5: 0x9d4c9add
…

A random number generator

Stream cipher

- Stream cipher
  - Example: RC4/RC5

Encrypt message 1 with 0x1b395a46
Encrypt message 2 with 0xf1737202
Encrypt message 3 with 0xccf0de05…

Decrypt message 1 with 0x1b395a46
Decrypt message 2 with 0xf1737202
Decrypt message 3 with 0xccf0de05…

Physically meet and share
40~2048 bits of keys
(max 512 bytes, short!)

1: 0x1b395a46
2: 0xf1737202
3: 0xccf0de05
4: 0x908b0feb
5: 0x9d4c9add
…

RC4 or
RC5
Algorithm

Key: 0xd7fe6798a7c6a9859b289ce

1: 0x1b395a46
2: 0xf1737202
3: 0xccf0de05
4: 0x908b0feb
5: 0x9d4c9add
…

RC4 or
RC5
Algorithm

Key: 0xd7fe6798a7c6a9859b289ce

Problem(s) in RC4/RC5

- See the Wikipedia sections
  - Bit-flipping attacks
  - Reused key attacks
  - Differential attacks
  - …

Topics for today

- Micro-lab instruction
- Crypto basics
  - Why do we need crypto?
  - What were the ancient crypto schemes?
  - What does it mean by perfectly secure?
  - What were the perfect crypto schemes so far?
  - What are the limitations of those above?
  - What were some practical solutions and their (also) limitations?

Thank You!

Tu/Th 4:00 – 5:50 PM (WNGR 149)

Sanghyun Hong

sanghyun.hong@oregonstate.edu

Oregon State University

SAIL

Secure AI Systems Lab