Introduction to Cryptography

The hardest arithmetic to master is that which enables us to count our blessings.

-Eric Hoffer

Cybersecurity

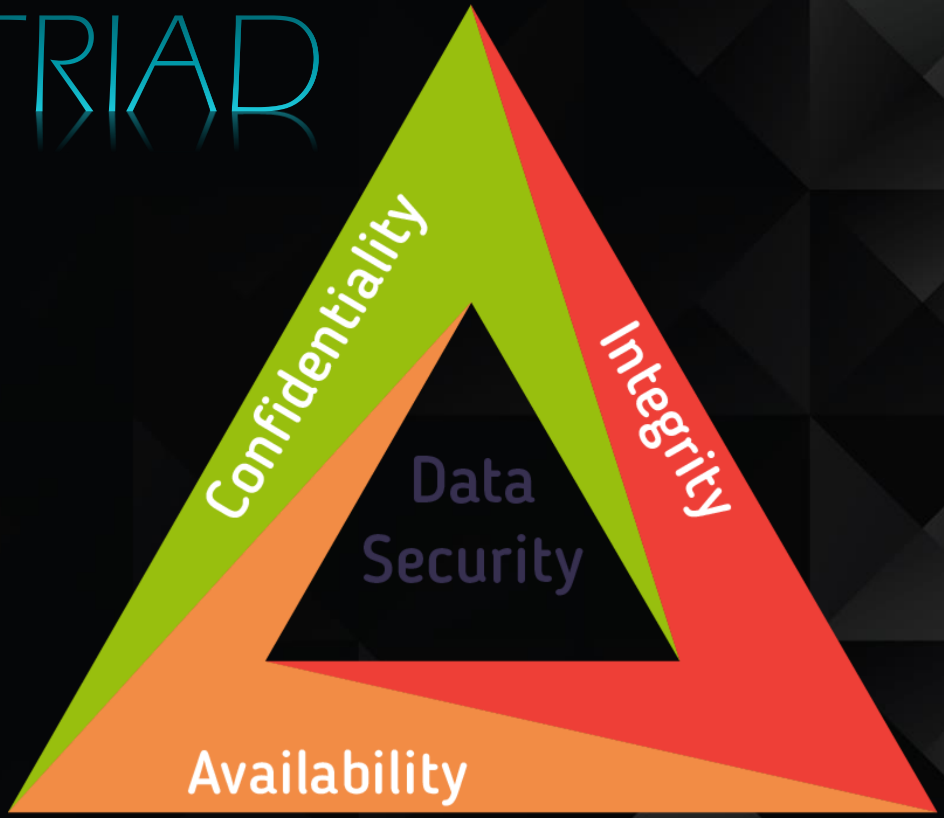
Cryptography Day 1

Today's Objectives

By the end of class, you will be able to:

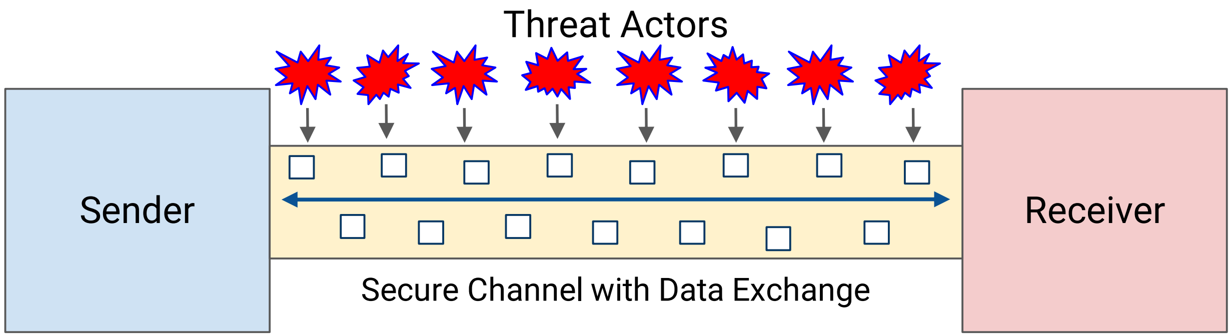
- [ ] Articulate the goals of cryptography.
- [ ] Discuss modern encryption standard, particularly AES.
- [ ] Differentiate between substitution and transposition ciphers.
- [ ] Explain how textual data can be represented in binary, octal, and hex.
- [ ] Determine the output of the XOR operation given two input bitstreams.
- [ ] Compare and contrast symmetric and asymmetric cryptography
- [ ] Use OpenSSL to encrypt data with a symmetric key.

CIA TRIAD



Introduction to Cryptography

Why Cryptography Matters



Information is often the most prized asset of individuals and organizations.

Ensuring information can be stored, transmitted, and received securely is a central responsibility of modern security professionals.

Goals of Cryptography

Introducing the P.A.I.N Framework:

01	**Privacy**
02	**Authentication**
03	**Integrity**
04	**Non-repudiation**

Goals of Cryptography

We'll look at each of these pillars in more detail later, but here is a brief overview of how each goal protects data:

01 **Privacy** ensures it is accessible by the intended recipients.

02 **Authentication** ensures it is sent from the claimed sender.

03 **Integrity** ensures it can't be intercepted and modified in flight.

04 **Non-repudiation** verifies if an individual did or did not send data.

Goals of Cryptography

Each pillar is supported by specific methods and tools.

01 **Privacy** is provided through encryption.

02 **Authentication** is provided through digital signatures.

03 **Integrity** is checked through hashes.

04 **Non-repudiation** is supported through signatures and certifications.

Student Activity:

Key Terms and Orientation

In this activity, you will familiarize themselves with some key terms in cryptography.

Activities/Stu_Orientation/README.md

Suggested Time:
15 Minutes

Key Terms and Orientation

Instructions:

1. Use the Cryptography Terminology resource (Wikipedia is an unreliable source)

- Privacy
- Authenticity
- Integrity
- Non-Repudiation

2. Define the following key terms, and provide an example of each if you can:

Plaintext Ciphertext Cryptography Code Bit Cipher Encryption

Key Encryption Decryption Symmetric-Key Cryptography Asymmetric-Key Cryptography

Hash Signature Checksum

Times Up! Let’s Review.

Key Terms and Orientation

Modern Cryptosystems and Standards

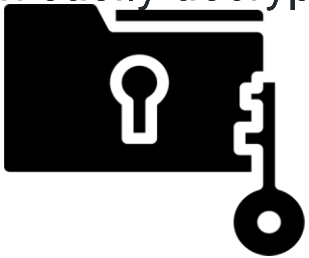
Modern Cryptosystems and Standards

Encryption, Decryption and Keys

Encryption is the process of obfuscating a plaintext by converting it into ciphertext by using a special values called a key along with a fixed set of rules.

Decryption is the process of turning a ciphertext back into plaintext.

The key must remain SECRET! Once the key is known, you can easily decrypt messages by applying the rules in reverse.



The Cryptography Trade-off

Encryption and decryption must be complex enough to provide a strong security but also simple enough to perform quickly with the key.

Ensuring Privacy and Confidentiality

Data at Rest	Data in Motion
Static data, such as that stored on a hard disk or a database. Prevents it from being intelligible to prying eyes	Data moving between machines on the network, such as your computer and YouTube, or your phone and the cell tower. More complicated than protecting data at rest

Ensuring Privacy and Confidentiality

Data at Rest

Protected by encrypting data on hard drives and in databases and storing sensitive data in multiple separate locations.

Tools:
Using AES to encrypt data

Storing AES-encrypted data in multiple database.

Data in Motion

More complicated, as at least two machines are exposed to the data.

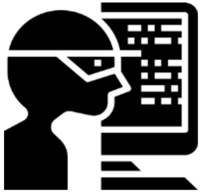
Must be able to encrypt / decrypt quickly

Tools:
Using AES in combination with RSA to protect communications on the web with TLS

Using SSH to encrypt remote shell sessions

Ensuring Authenticity

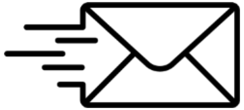
With authenticity, a user can verify the identity of a data source.



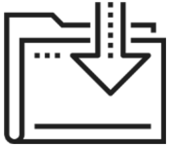
Attackers can send encrypted data claiming to be someone they're not

Encryption will provide little protection in these cases.

With the use of asymmetric cryptography, digital signatures are the main tool used to verify the authenticity of a message. Then, we can ensure:



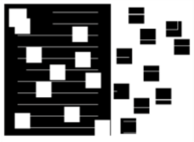
Emails came from the claimed sender.



Downloaded files come from the correct server.

Ensuring Integrity

We need to know that the data we are sharing is the correct information.



Hashing can be used to verify the integrity of data by:

Verifying that a file downloaded from a server is in fact the file on the server, as opposed to an attacker intercepting and modifying the file.

Verifying that data has made it across the wire without corruption.

Client requests data from a server → Server sends the hash digest of data to the client → Server sends the requested data → Server hashes data received and compares to digest from server → If digest matches, the data hasn't been corrupted

Ensuring Non-Repudiation

Ensuring that a message is inextricably linked to a sender.

If Bob sent a nasty memo to an entire department, non-repudiation ensures that he can’t scapegoat someone else.

Today’s de facto standard of symmetric key algorithm is AES.

Developed through community collaboration led by the National Institute of Standards and Technology.

Activity: DES Death March

In this activity, you will research the NIST competition model then answering the corresponding questions.

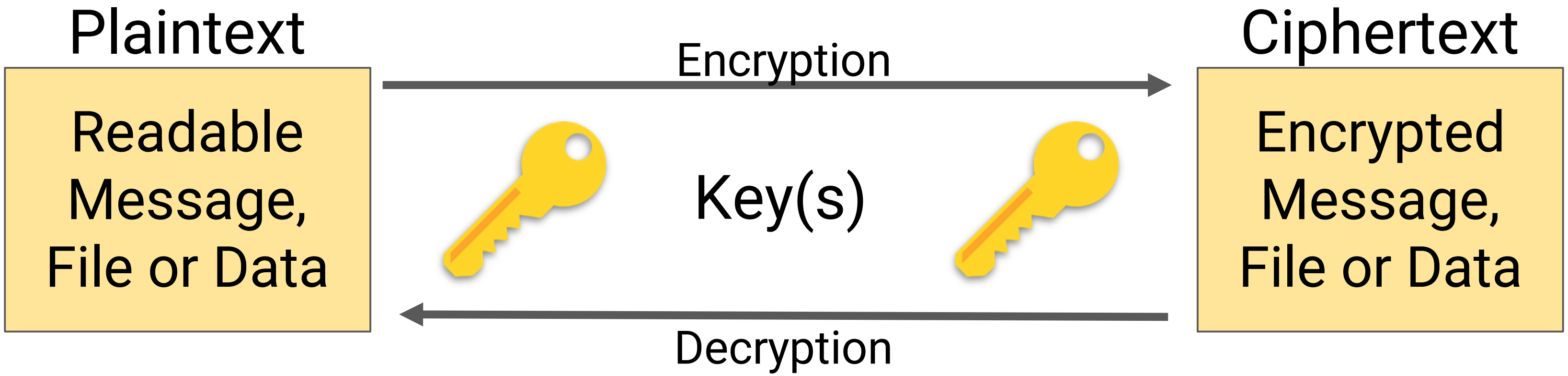
Instructions sent via Slack

Suggested Time:
15 Minutes

Times Up! Let’s Review.

DES Death March

The Key Concept of Cryptography



A key the piece of information / parameter that specifies how plaintext should be transformed into ciphertext and vice versa.

The Cryptography Trade-off

Do we want an incredibly strong cipher that’s hard to compute and difficult to decrypt?

OR

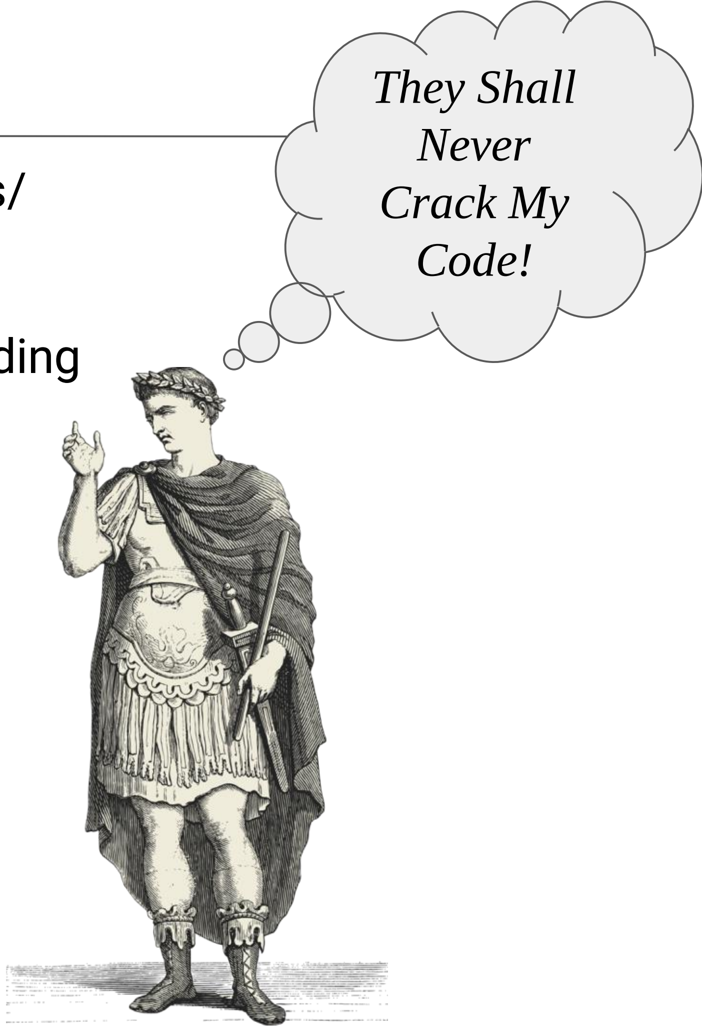
Do we prefer an *average* cipher that is faster.

Ciphers and Keys

The Ceasar Cipher

One of the best known historic encryption techniques/

A simple substitution cipher in which a message is shifted a number of letters down the alphabet according to a key.



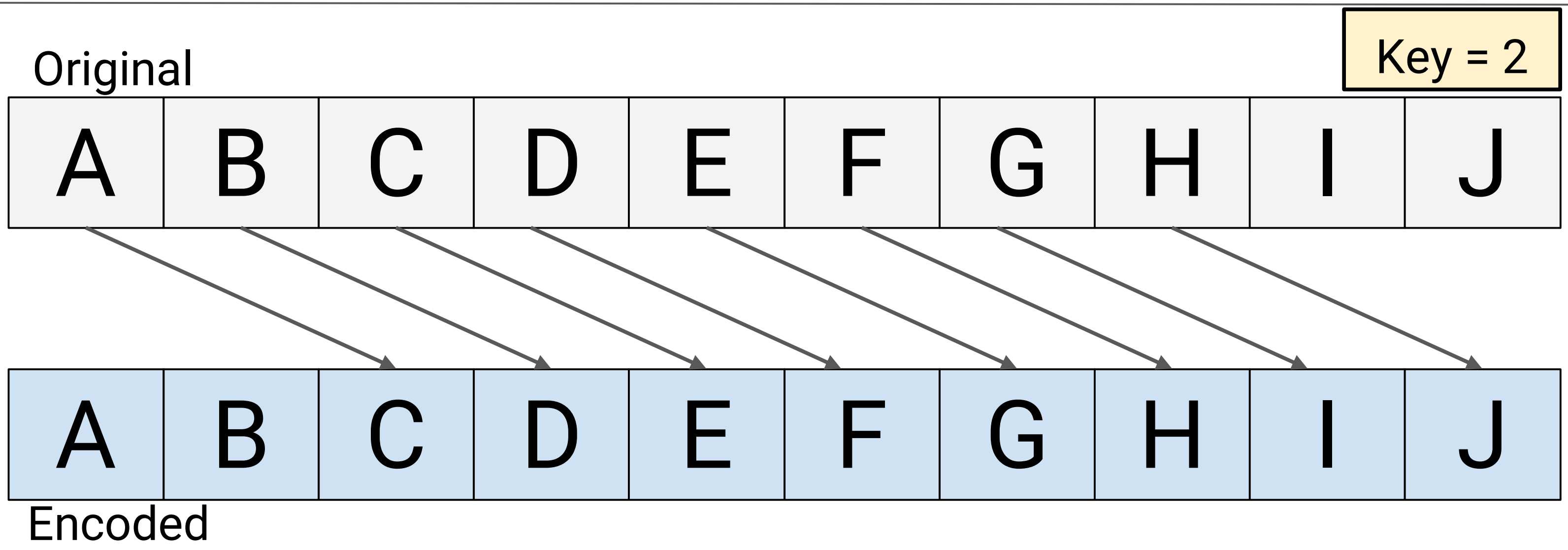
<u>Sender:</u>

1. Writes a Message
2. Selects a Key
3. Encodes Message

<u>Recipient:</u>

1. Receives Message
2. Uses *known* Key
3. Decodes Message

How the Caesar Cipher Works



The Caesar Cipher works by shifting letters a set number (key value) of indices from the original position.

<u>Example for Key = 2:</u>

“I HID A CAB” → “K JKF C ECD”

“A BAD DAD” → “C DCF FCF”

Transposition Ciphers

Transposition Ciphers are permutations that break messages into equal sized blocks and re-arrange the letters to a fixed rule.

Permutation is simply a rearrangement of a sequence of letters.

hlleo is a permutation of hello

The following rule breaks an input into three blocks and replaces the 1st, 2nd, and 3rd letters of the block with the 3rd, 1st, and 2nd respectively.

$$E = \begin{vmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{vmatrix}$$

Character Encoding
and
Binary Representation

Letter ciphers are not compatible with the numerical representation of digital data.

Character Encoding is the process of representing letters as numbers in order to encrypt data on computers.

Character Encoding

Some forms of character encoding we'll look at: ASCII

ASCII stands for the American Standard Code for Information Interchange.

- It is used to represent computer-stored characters in a *human-readable* format.
- Look down at you keyboards. Every character is part of ASCII system: Upper and lowercase letters, special characters (!@#$...), numericals (1,2,3,4…)
- We can convert strings of comprehendable sentences into purely numerical strings.

http://www.asciitable.com

Character Encoding

Some forms of character encoding we'll look at: Binary

Binary Number System is the numerical representation of computer data as 1's and 0s

We can understand Binary if we examine it relative to the common decimal system:

https://www.convertbinary.com/alphabet

Hexadecimal and Octal

Even though computers read in binary, it is not a very efficient representation of data

Hello as binary = 11010001100101110110011011001101111

- Hexadecimal system is a *more compact representation of binary data*.

  Hex uses 16 characters to represent the base value. In other words, it is a base 16 system

  The base numbers range from 0-9 and then letters A-F represent 11, 12, etc.

  (A = 11, B= 12, C =13 etc.)

- If we count our fingers using the decimal system, we’d count 1 to 10.

  If we count on our fingers using the hex system, we’d count 1 to F.

- Hexadecimal numbers are written with a \x to indicate the following number is to be read as hex: \x10

Comprehending Hex

Hex uses 16 characters to represent the base value. In other words, it is a base 16 system

The base numbers range from 0-9 and then letters A-F.

Dec.	Hex.	Dec.	Hex.
0	0	8	8
1	1	9	9
2	2	10	A
3	3	11	B
4	4	12	C
5	5	13	D
6	6	14	E
7	7	15	F

Dec.	Hex.	Dec.	Hex.
16	10	24	18
17	11	25	19
18	12	26	?
19	13	27	?
20	14	28	?
21	15	29	?
22	16	30	?
23	17	31	?

What do you think comes next?

Comprehending Hex

Hex uses 16 characters to represent the base value. In other words, it is a base 16 system

The base numbers range from 0-9 and then letters A-F.

Dec.	Hex.	Dec.	Hex.
0	0	8	8
1	1	9	9
2	2	10	A
3	3	11	B
4	4	12	C
5	5	13	D
6	6	14	E
7	7	15	F

Dec.	Hex.	Dec.	Hex.
16	10	24	18
17	11	25	19
18	12	26	1A
19	13	27	1B
20	14	28	1C
21	15	29	1D
22	16	30	1E
23	17	31	1F

And after 1F?

Comprehending Hex

Hex uses 16 characters to represent the base value. In other words, it is a base 16 system

The base numbers range from 0-9 and then letters A-F.

Dec.	Hex.	Dec.	Hex.
0	0	8	8
1	1	9	9
2	2	10	A
3	3	11	B
4	4	12	C
5	5	13	D
6	6	14	E
7	7	15	F

Dec.	Hex.	Dec.	Hex.
16	10	24	18
17	11	25	19
18	12	26	1A
19	13	27	1B
20	14	28	1C
21	15	29	1D
22	16	30	1E
23	17	31	1F

Dec.	Hex.	Dec.	Hex.
32	20	40	28
33	21	41	29
34	22	42	2A
35	23	43	2B
36	24	44	2C
37	25	45	2D
38	26	46	2E
39	27	47	2F

The mechanics of conversion are not as important as simply knowing these systems are in place to represent and encode data.

Conversion Demonstration

Instructions

Convert the following into binary code:

57 65 6c 63 6f 6d 65 20 74 6f 20 74 68 65 20 73 63 61 76 65 6e 67 65 72 20 68 75 6e 74 21 20 47 6f 20 74 6f 20 74 68 65 20 66 6f 6c 6c 6f 77 69 6e 67 20 77 65 62 73 69 74 65 3a 0a 68 74 74 70 3a 2f 2f 77 77 77 2e 70 61 67 65 6f 72 61 6d 61 2e 63 6f 6d 2f 3f 70 3d 73 65 63 72 65 74 34

Hint: Copy and paste the above a free converter Website and this should give you a list of results that will convert this code for you.

https://www.asciitohex.com/

Activity: Character-Encoding Scavenger Hunt

In this activity, you will complete a scavenger hunt by identifying the type of character encoding of the provided numbers, and then decoding it to find the next clue

Activities/Stu_scavenger_hunt/ReadMe.md

Suggested Time:
5 Minutes

Take a Break!

Checkpoint:

By the end of class, you will be able to:

- Articulate the goals of cryptography.
- Discuss modern encryption standard, particularly AES.
- Differentiate between substitution and transposition ciphers.
- Explain how textual data can be represented in binary, octal, and hex.
- Determine the output of the XOR operation given two input bitstreams.
- Compare and contrast symmetric and asymmetric cryptography
- Use OpenSSL to encrypt data with a symmetric key.

Bitwise Operators and XOR Cipher

Bits ‘n Bytes

Now we’ll dive into how encryption works in a digital setting.

Here is the binary representation for the letter “h”:

1$\boxed{1}$01000

This is a bit.
A bit is each place of a binary string.

Bits ‘n Bytes

Now we’ll dive into how encryption works in a digital setting.

Here is the binary representation for the letter “h”:

$$\boxed{1101000}$$

This is a byte. Eight bits is a byte.

Bits ‘n Bytes

Now we’ll dive into how encryption works in a digital setting.

Here is the binary representation for the letter “h”:

1101000

→ All digital data is represented in terms of bits.

→ We can encrypt bits just like letters, with substitution and permutation.

→ Another crucial tool for encryption is XOR, which will cover in a moment.

Bitwise Operators

Bitwise operators take two binary numbers, compare them, and output a binary result based on specific conditions used to compare them.

X	Y	X & Y	X | Y	X^Y (XOR)
1	1	1	1	0
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1

Input	Input	AND Output	OR Output	XOR Output

Bitwise Operator - AND

AND is the simplest bitwise operator.

X	Y	X & Y	X | Y	X^Y (XOR)
1	1	1	1	0
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1

If both inputs have a 1 in the same place, then output is 1. Otherwise, output is 0

Example: `1100 & 0110 = 0100`

Bitwise Operator - OR

OR is the opposite of AND.

X	Y	X & Y	X | Y	X^Y (XOR)
1	1	1	1	0
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1

If either input has 1 in the slot, then output is 1. Otherwise, output is 0

Example: `1100|0110 = 1110`

Bitwise Operator - ORX

XOR (Exclusive Or) is the main operator we'll focu on for encrypting:

X	Y	X & Y	X | Y	X^Y (XOR)
1	1	1	1	0
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1

If other inputs have a 1 in a slot, or if both inputs have 0 in a slot, output 0.
Otherwise, output 1.

Example: `1100^0110 = 1010`

XOR Cipher

XOR can be used for encryption because it reversible.

X	Y	X^Y
1100	0110	1010
1010	0110	1100

A ^ B = C ⟷ **C ^ B = A**

We can encrypt a number A, by using a key B, to produce the encoded data in C.

We can decrypt C by XOR-ing it with the key B, to retrieve the plaintext, A.

Bitwise Operators

Bitwise operators take two binary numbers, compare them, and output a binary result based on specific conditions used to compare them.

X	*Y*	*AND(X,Y)*	*OR(X,Y)*	*NAND(X,Y)*	*NOR(X,Y)*	*XOR(X,Y)*
0	0	0	0	1	1	0
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	1	0	0	0

XOR vs AND / OR

XOR is irreversible without the knowing the key.

When applying the bitwise operator AND to two bitstreams, there's a 25% chance you'll get a 1.

When applying the bitwise operator OR to two bitstreams, there's a 75% chance you'll get a 1.

By contrast, XOR has a 50% chance of outputting a 0 or 1.

Activity: XOR Cipher

In this activity, students will develop an intuition for the behavior of XOR and build truth tables for AND, OR, and XOR.

Activities/Stu_XOR/README.md

Suggested Time:
7 Minutes

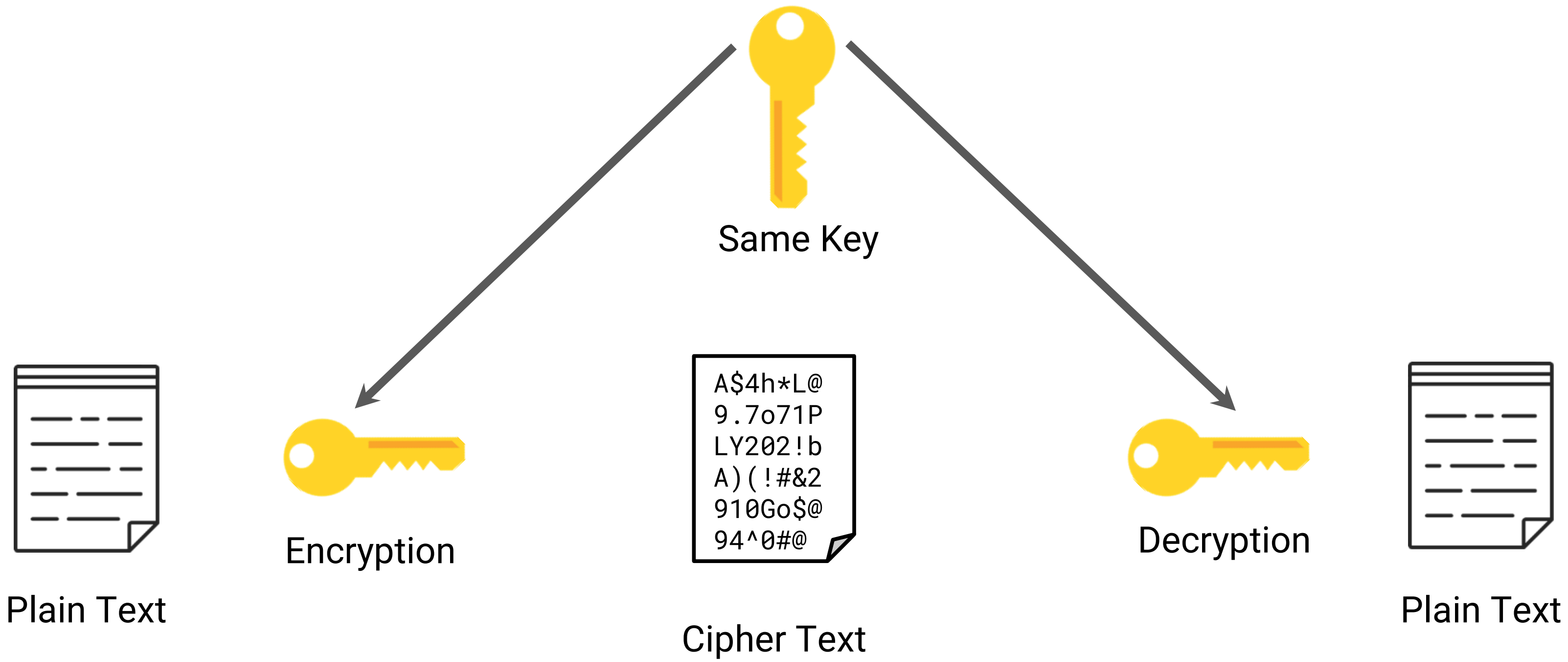
Times Up! Let's Review.

Scavenger Hunt

Symmetric vs. Asymmetric Cryptography

Symmetric-Key Algorithm

Symmetric-key algorithms are algorithms that use the same key to encrypt and decrypt.



Symmetric Encryption

The process:

01 Divide Data into Blocks (Transposition)

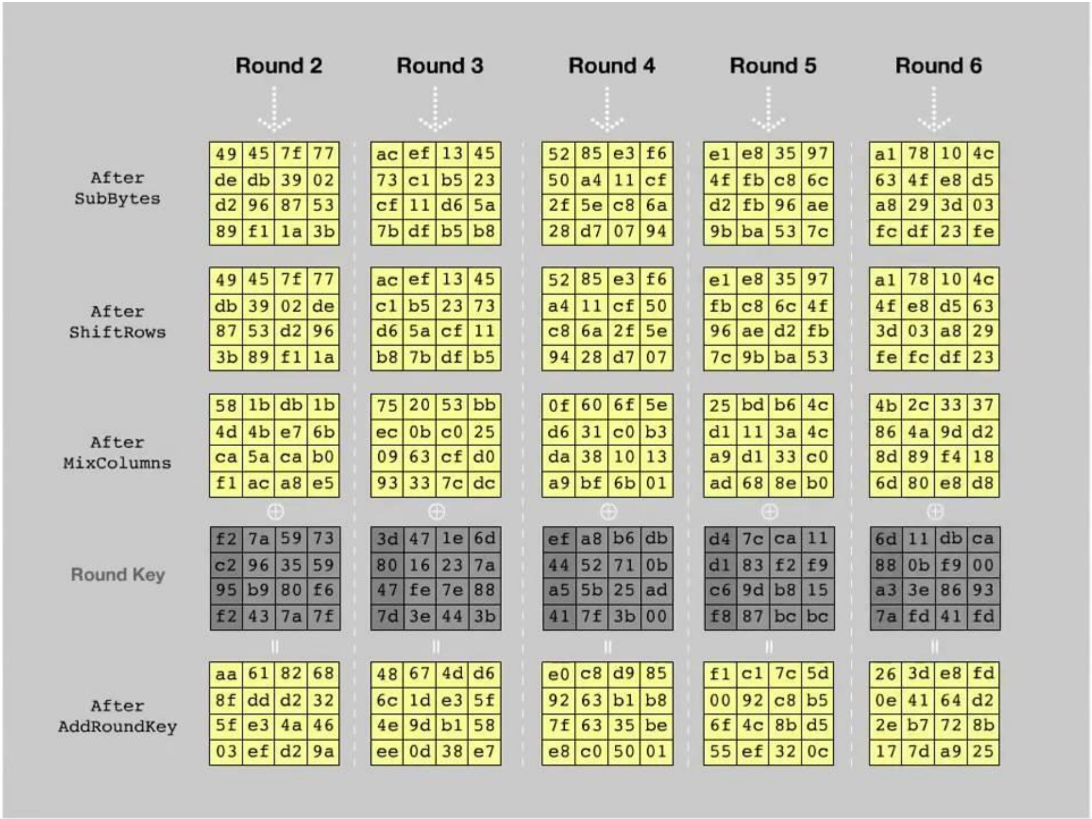
02 Perform a substitution on each block

03 Perform a permutation of the resulting substitution

04 The Key is XORed with the result permutation.

05 The process is repeated multiple times, and the blocks recombines create and encrypted whole.

Symmetric Shuffle



Key Exchange

One key is required for each pair of people who want to share encrypted messages:

Alice, Bob, Eve, and Jane

Alice ⟷ Bob

Alice ⟷ Eve

Alice ⟷ Jane

Bob ⟷ Eve

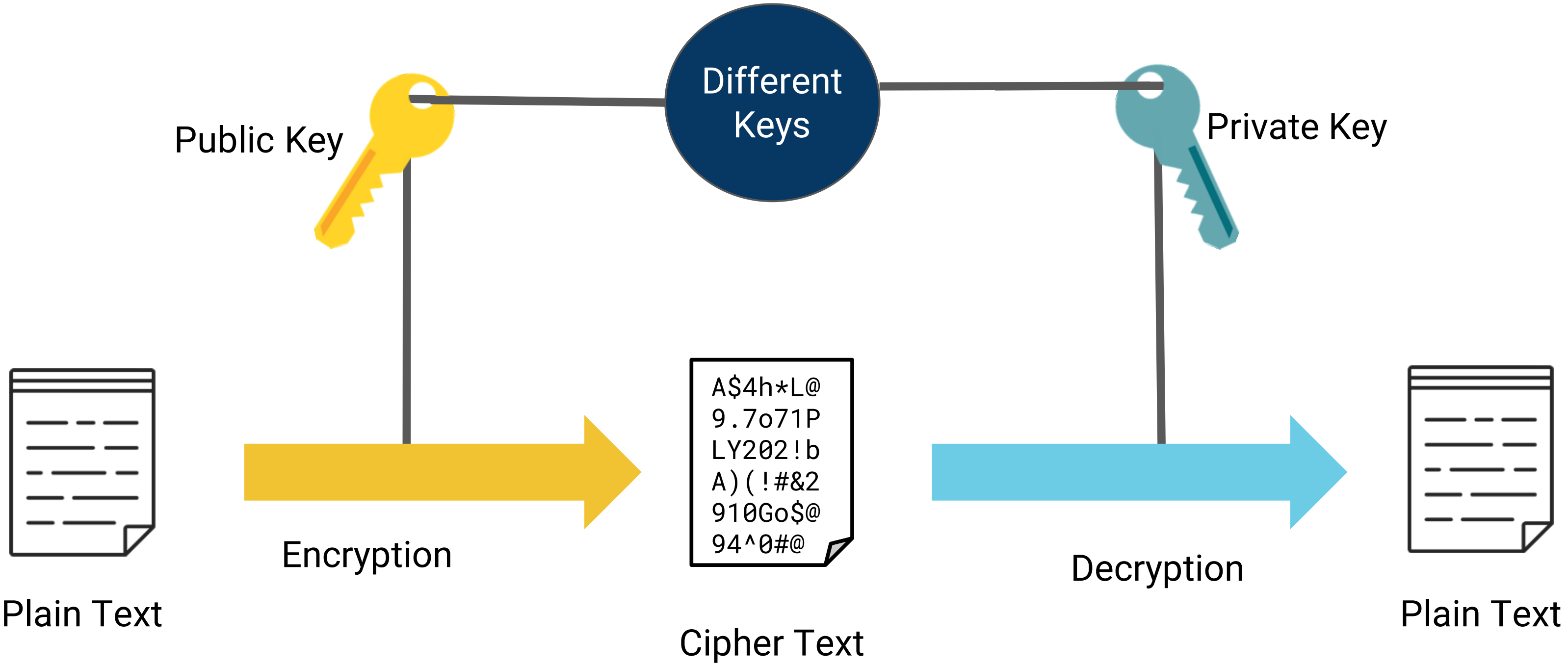
Bob ⟷ Jane

Jane ⟷ Eve

Six different keys are required to share amongst four people.
Asymmetric encryption addresses this issue...

Asymmetric-Key Algorithm

Asymmetric-key cryptography (a.k.a public-key cryptography) uses two keys to encrypt and decrypt.



Symmetric Vs. Asymmetric

Symmetric

- + Faster
- - / + Less Computationally intense
- - Key Exchange Problem

De facto standard is AES (Advanced Encryption Standard)

Asymmetric

- + / - Computationally Intense
- + Easier to use with many people
- + Used to verify identity / authenticity via digital signatures

De facto standard is RSA (Rivest-Shamir-Adleman)

Activity / Facilitated Discussion: Symmetric vs. Asymmetric

In this activity, you will look at four scenarios and determine whether symmetric or asymmetric is more appropriate.

Activities/Stu_Symmetric_vs_Asymmetric/README.md

Suggested Time:
7 Minutes

Checkpoint: Almost There!

By the end of class, you will be able to:

- [x] Articulate the goals of cryptography.
- [x] Discuss modern encryption standard, particularly AES.
- [x] Differentiate between substitution and transposition ciphers.
- [x] Explain how textual data can be represented in binary, octal, and hex.
- [x] Determine the output of the XOR operation given two input bitstreams.
- [x] Compare and contrast symmetric and asymmetric cryptography
- [ ] Use OpenSSL to encrypt data with a symmetric key.

Symmetric Encryption with Open SSL

Activity : Encrypting Data with OpenSSL

In this activity, you will use OpenSSL to encrypt data using AES.

Activities/Stu_OpenSSL

Suggested Time:
5 Minutes

Crypto Class Checkpoint

By the end of class, you will be able to:

- Articulate the goals of cryptography.
- Discuss modern encryption standard, particularly AES.
- Differentiate between substitution and transposition ciphers.
- Explain how textual data can be represented in binary, octal, and hex.
- Determine the output of the XOR operation given two input bitstreams.
- Compare and contrast symmetric and asymmetric cryptography
- Use OpenSSL to encrypt data with a symmetric key.