Cryptography I

ISSS Beginner Series

Objective-

Eavesdroppers should not gain any additional information even if they intercept messages being exchanged.

Topics:

- ➔ Caesar Cipher
- ➔ Confusion/Diffusion
- ➔ Stream/Block Ciphers
- ➔ Key Distribution Problem
- ➔ One Time Pad
- ➔ Symmetric/Asymmetric Ciphers

Some Vocabulary

- **Plain Text:** original form of a message
- **Cipher Text:** encrypted form of a message
- **Key:** specifies a set of rules to transform the plaintext into the cipher text
- **Encryption:** process of concealing the message of a text, through a key
- **Decryption:** process of recovering the message from a cipher text, using a key

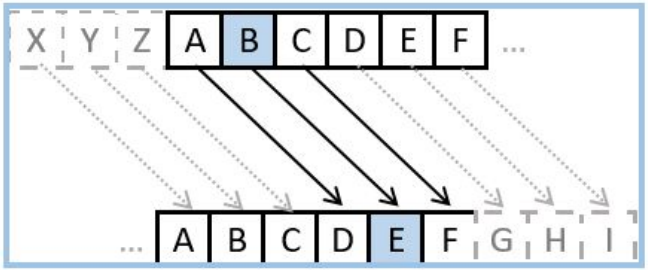
Qualities of Good Encryption Algorithms

- Almost every strong encryption algorithm is theoretically breakable
- For a good encryption algorithm there will be no better alternative to decrypting than brute forcing the different key options

But when are there alternatives to brute-forcing encryption algorithms?

Caesar Cipher

- Caesar Cipher is a very simple encryption technique that chooses an arbitrary number that it uses to shift the alphabet over and then substitutes the values



SHIFT +3

This Caesar cipher has a shift of 3 characters.

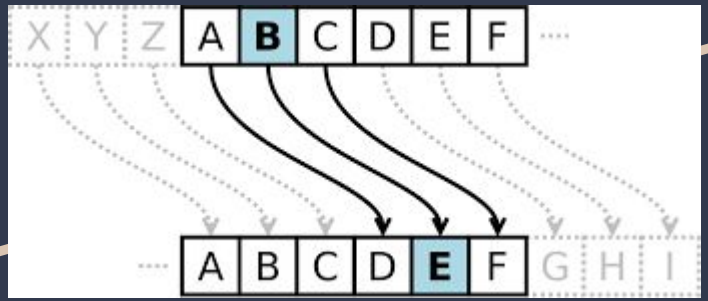
The letter ‘A’ becomes a ‘D’. The letter ‘B’ becomes ‘E’.

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

Plaintext

Ciphertext

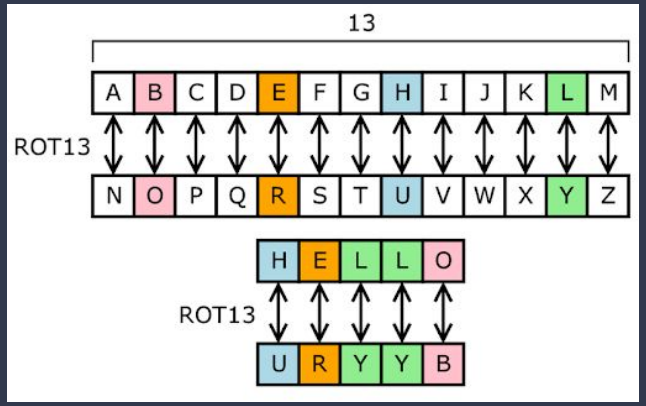
Why is Caesar Cipher considered a Weak Encryption?



- EX. If you know that E is the most common letter in the English alphabet
  - Knowing this you could find what the most common letter in the encryption test is, and assume it is actually E, and figure out what the shift is.
  - Thus, we have reduced the key space by making educated guesses

Confusion

- Exchange symbols
- Often achieved through substitution
- Ex. Stream Ciphers



Diffusion

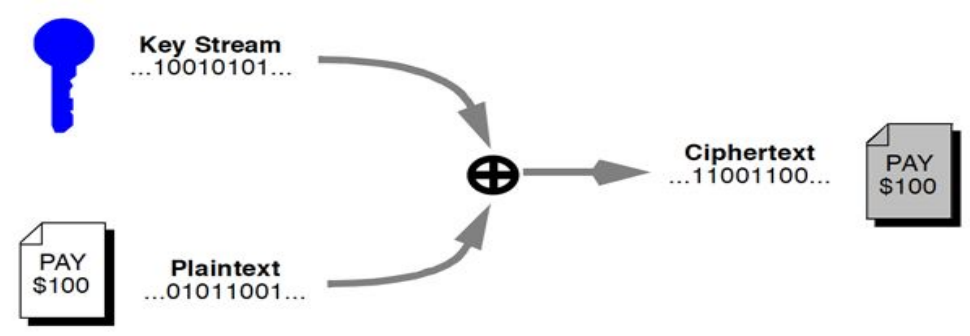
- Change the placement of symbols
- Often achieved with transposition
- Ex. Block Ciphers

Message: JAMESBONDNEEDSBACKUP
Code: JEONDAUASNESCPMBDEBK

J	E	O	N	D	A	U
A	S	N	E	S	C	P
M	B	D	E	B	K	

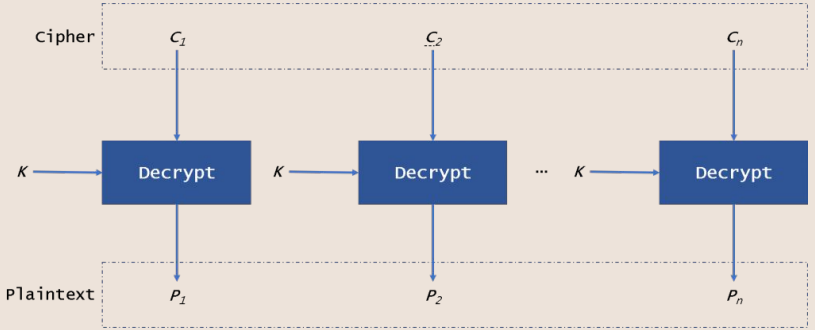
Stream Ciphers

- **Convert symbols in the text directly (Ex. simple substitution)**



Pros	Cons
High Speed of Encryption (typically linear)	Low Diffusion
Simple, less error prone	Susceptible to insertions by attackers

Block Ciphers



- Encrypt a group of text as one block
- Modern Block sizes are about 128 bits
- EX.RSA

Message: JAMESBONDNEEDSBACKUP
Code: JEONDAUASNESCPMBDEBK

J	E	O	N	D	A	U
A	S	N	E	S	C	P
M	B	D	E	B	K	

Pros	Cons
High Diffusion	Slow Encryption/Decryption
Harder to Tamper with	Error Prone

Ok so... we now know about encryption requires a key... and the key has to be kept private...

So how do we get the key from person A to person B without it getting leaked??

Hence....

The Key Distribution Problem.

Perfect Ciphers

One for which there is no reduction of the search space gained from knowing the following:

- ➔ Encryption Algorithm
- ➔ Cipher Text

The perfect cipher requires the key to be at least as long as the plaintext

One Time Pad

- Theoretically perfect cipher
- Stream Cipher
- XOR a random key (that is the same length as the plaintext) with the plaintext

EX.

Key:	**0110100010101**
Plaintext:	**1110101011010**
Ciphertext:	**1000001001111**

So why don't we always use the one time pad?

Why is this so strong?

When an attacker tries to decrypt cipher text, there are 2^n possible plaintexts that could be the pre-image of the cipher text under a plausible key

Fatal Flaw in One Time Pad

- Problem with one time pad is that it only works for one message exchange
- If you create a new key for every iteration
  - you would need a secure channel to exchange that key
  - The key is the same length as the message
  - at that point you could just exchange the message itself ->*key exchange problem*

Why can we only use it once?

1st Round of Encryption: $m_1 \oplus k$

2nd Round of Encryption: $m_2 \oplus k$

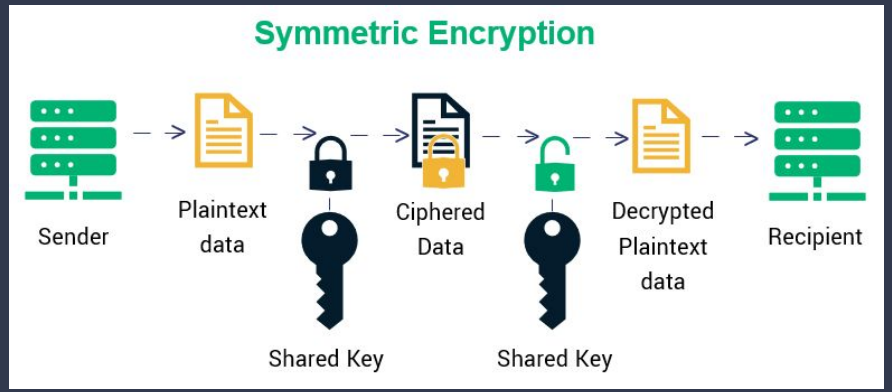
$$(m_1 \oplus k) \oplus (m_2 \oplus k) = m_1 \oplus m_2$$

Hence, there is memory leakage.

There is a approximate one time pad that solves this issue. *(Vernam Cipher)*

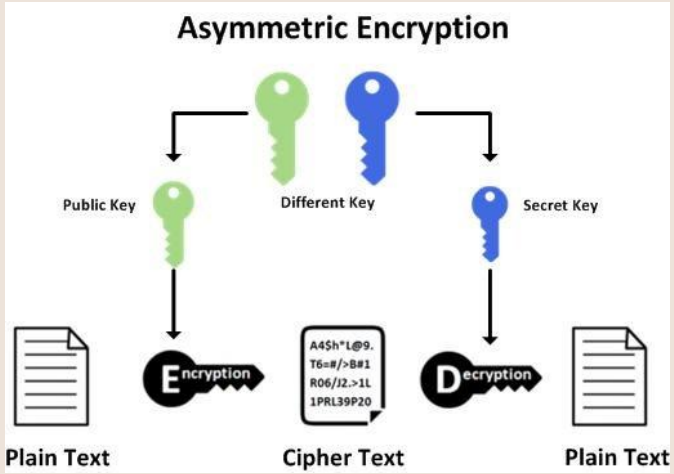
Symmetric

- Same key is used to encrypt and decrypt messages
- Generally involves a randomly generated k-bit string of characters for the key
- EX. AES, DES



Asymmetric

- Same key is not used to encrypt and decrypt a message
- Involves a private and public key
- EX. RSA, Diffie Hellman



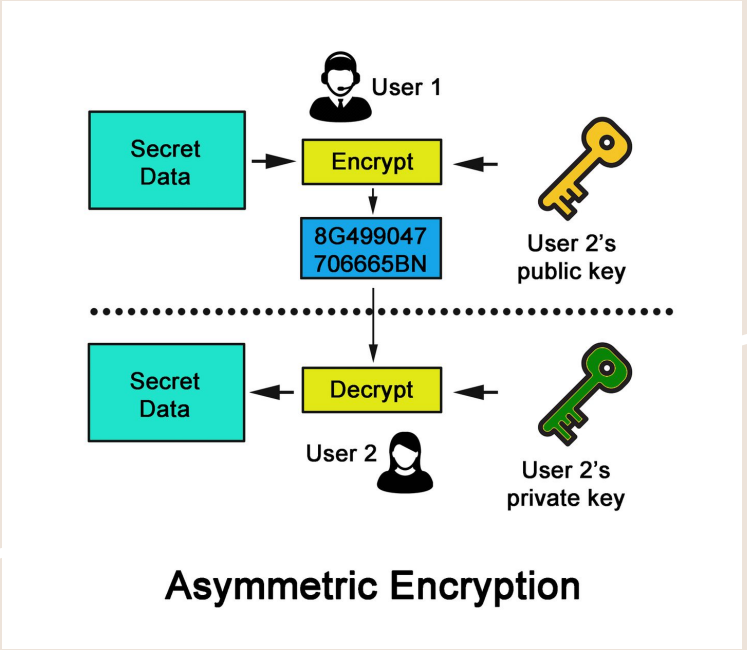
Symmetric Ciphers

- Same key is used to encrypt and decrypt it

Pros	Cons
Simple Calculations, very fast	Key Distribution Problem

Asymmetric Ciphers

- Each subject S has a publicly disclosed key (***public key***) that anyone can use to encrypt a plaintext and a ***private key*** that only they can use to decrypt the ciphertext



Asymmetric Encryption

Pros	Cons
Solves the key distribution problem	**Expensive** to generate public keys, (keys usually involve large prime numbers because is no polynomial time algorithm to factor prime numbers)

Next Time:

- Diffie-Hellman
- RSA

Questions?