CSCI 466: Networks

Network Security (Message Confidentiality)

Reese Pearsall
Fall 2024

Announcements

- Wireshark Lab 3 due **tonight**
- PA3 posted. Due Sunday November 3rd
- You can see the questions you got wrong on quizzes
- Class registration begins this week

Programming Assignment 4

Wireshark Lab 4

Goal:

Two parties (Bob) and (Alice) want to communicate **securely**

Principles of **secure communication**

1. **Confidentiality** – only the sender and receiver should be able to understand the contents of the transmitted message

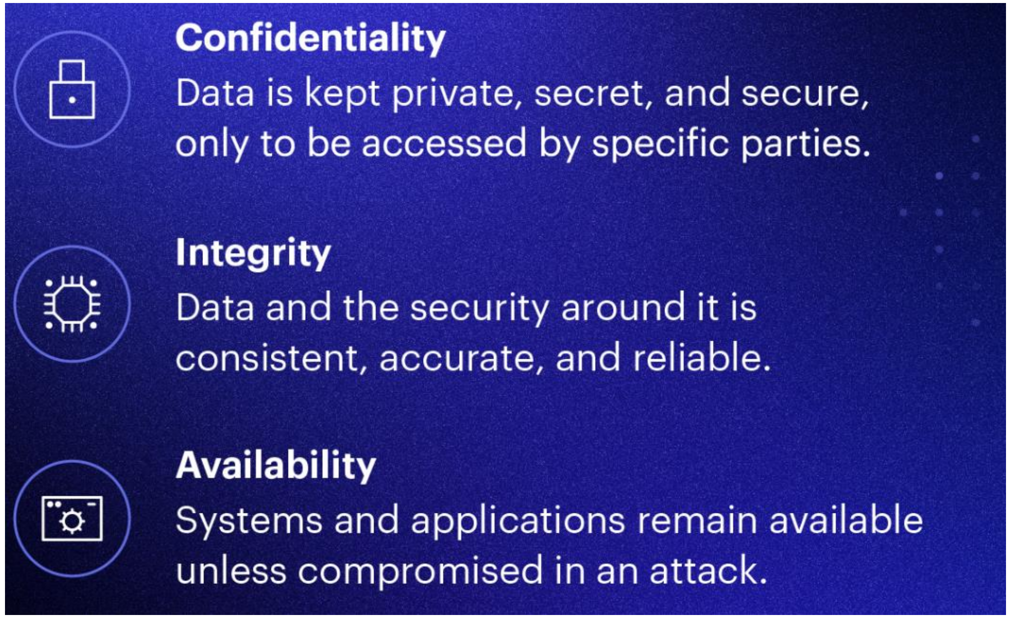
2. **Message Integrity** – the contents of the message have not been altered (maliciously or by accident)

3. **Authentication** – Both the sender and receiver should be able to confirm the identity of the other party involved in communication

4. **Authorization-** Should a user be allowed to do such action?

Operational Security- Infrastructure to help prevent hosts/networks from getting compromised (Firewalls, VPNs, etc)


CONFIDENTIALITY
INTEGRITY
AVAILABILITY
Data protected by CIA Triad

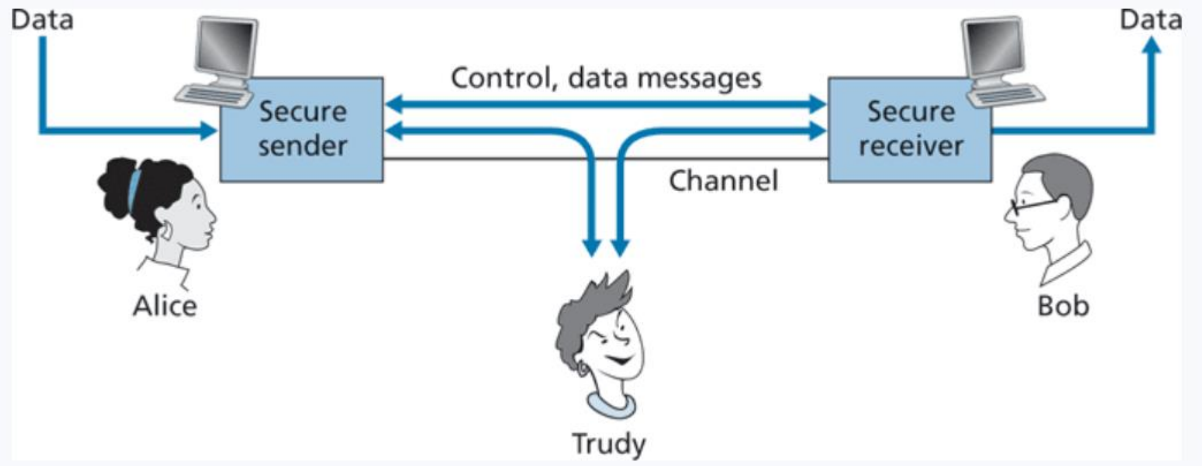

Confidentiality
Data is kept private, secret, and secure, only to be accessed by specific parties.
Integrity
Data and the security around it is consistent, accurate, and reliable.
Availability
Systems and applications remain available unless compromised in an attack.

Principles of Cryptography

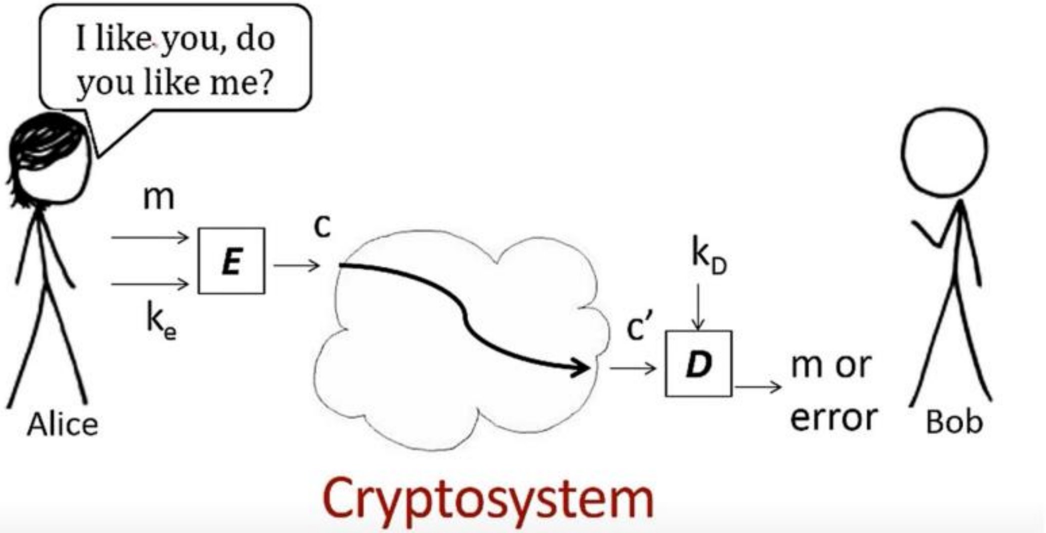
Presentation Layer

Session Layer

Goal: Only the sender and intended receiver should be able to understand the contents of a transmitted message (confidentiality), so sender must find a way to **encrypt** his message



(Trudy could steal information, modify information, or **spoof** her own message)



m: Plaintext ***k_e***: Encryption Key ***k_d***: Decryption Key

c: Ciphertext ***E***: Encryption Program ***D***: Decryption Program

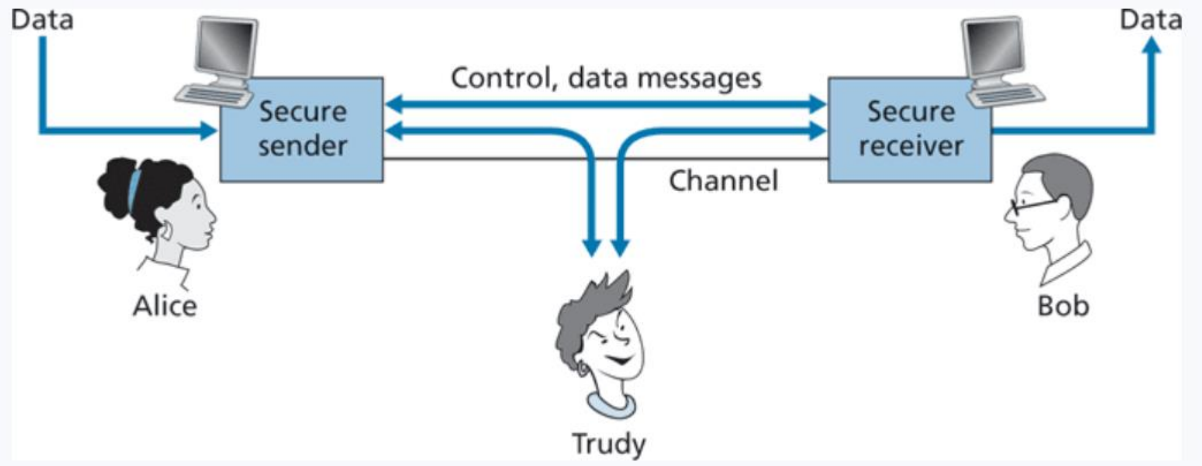
Deterministic programs*

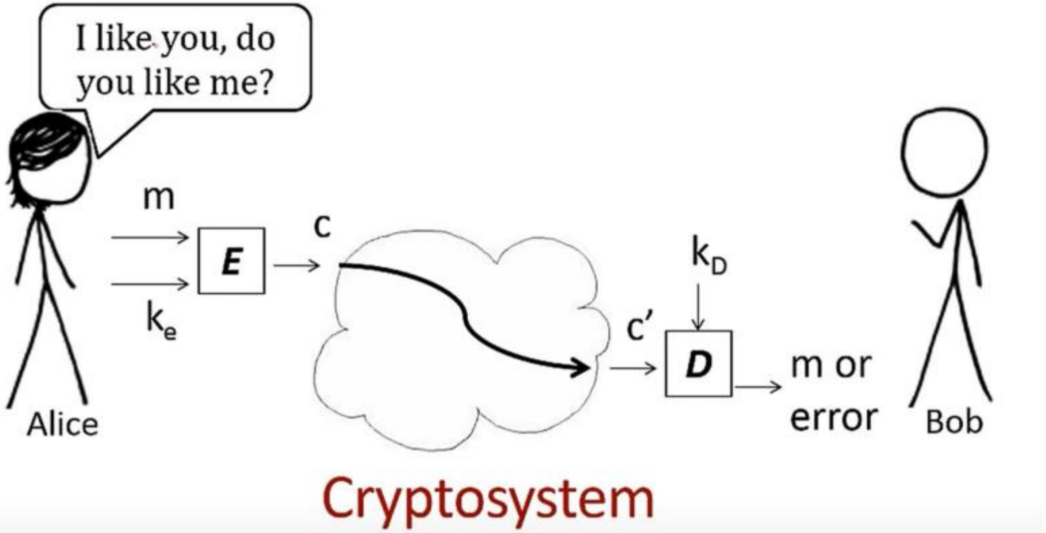
Principles of Cryptography

Presentation Layer

Session Layer

Goal: Only the sender and intended receiver should be able to understand the contents of a transmitted message (confidentiality), so sender must find a way to **encrypt** his message





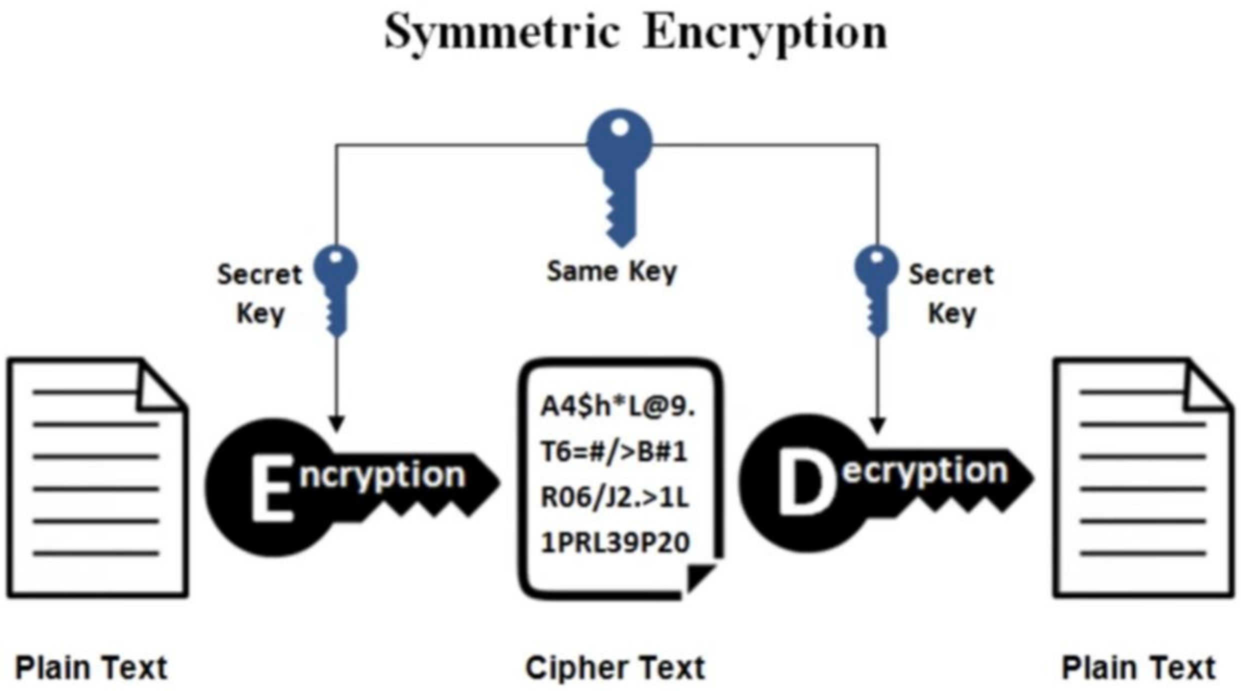
m: Plaintext
c: Ciphertext
k_e: Encryption Key
E: Encryption Program
k_d: Decryption Key
D: Decryption Program

Deterministic programs*

*We also need to make sure that the message is not tampered with before arrival (**message integrity**) and that both parties can identify each other (**authentication**)

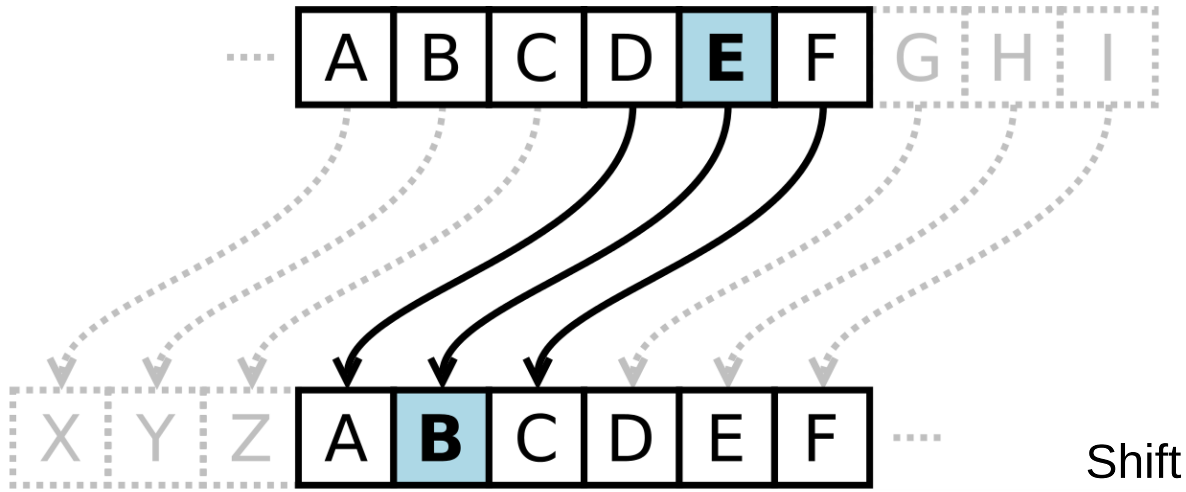
Symmetric Key Cryptography

Symmetric Key Cryptography is a type of encryption where only one key (a secret key) is used to both encrypt and decrypt electronic information

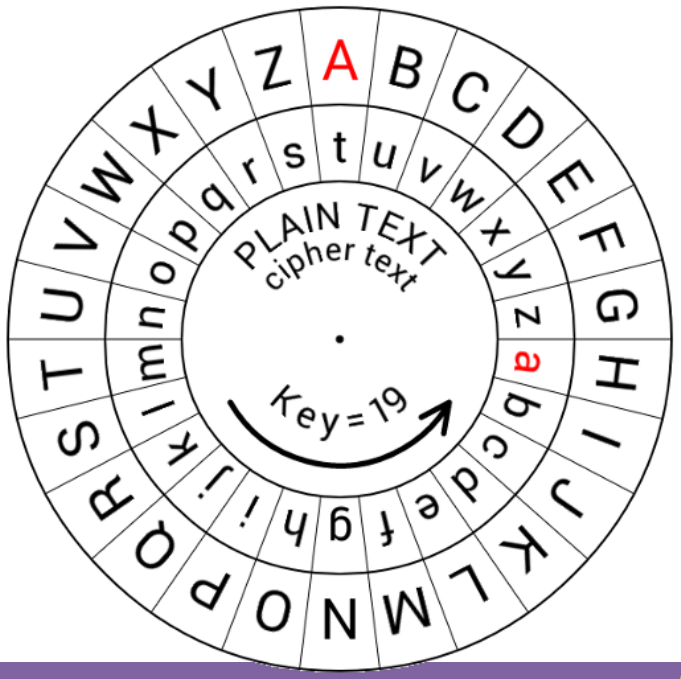


Early Symmetric Key Cryptography

Caesar Cipher- Each letter in plaintext is replaced by a letter some *fixed number* of positions down the alphabet



Shift of 3

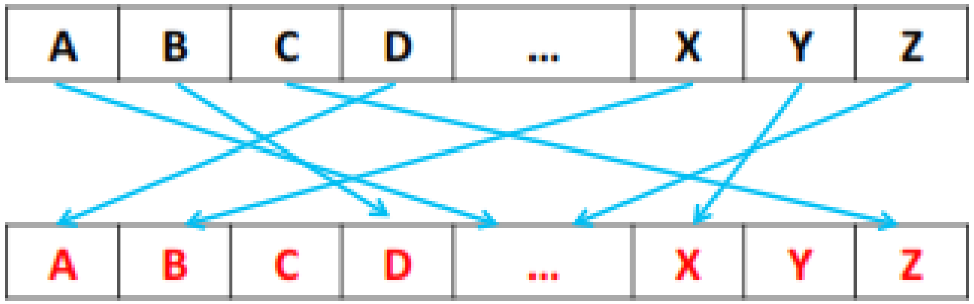


Brown **L**a**z**y **F**o**x** → **E**urzq **O**d**c**b **I**r**a**

If you did not know the key, how difficult would it be to crack a Caesar cipher?

Early Symmetric Key Cryptography

Monolithic Substitution Cipher- each letter of the plain text is replaced with another letter of the alphabet (no fixed length position)



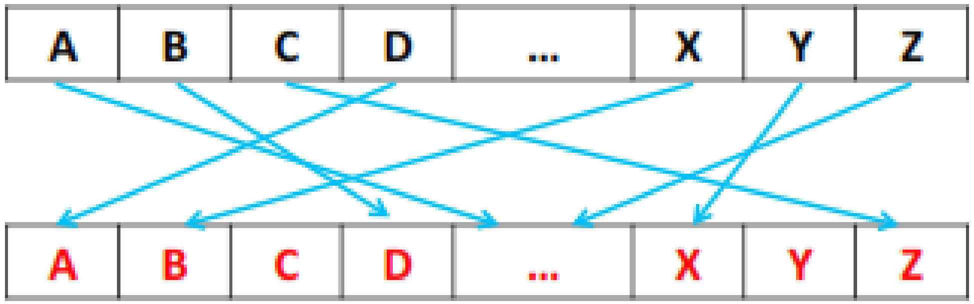
What does a key look like?

26-Characters
"EABZTIVGSKXFJPYCDWONMHQLRU"

If we don't know the key, how difficult would it be to **brute force** this?

Early Symmetric Key Cryptography

Monolithic Substitution Cipher- each letter of the plain text is replaced with another letter of the alphabet (no fixed length position)



What does a key look like?

26-Characters
“EABZTIVGSKXFJPYCDWONMHQLRU”

If we don’t know the key, how difficult would it be to **brute force** this?

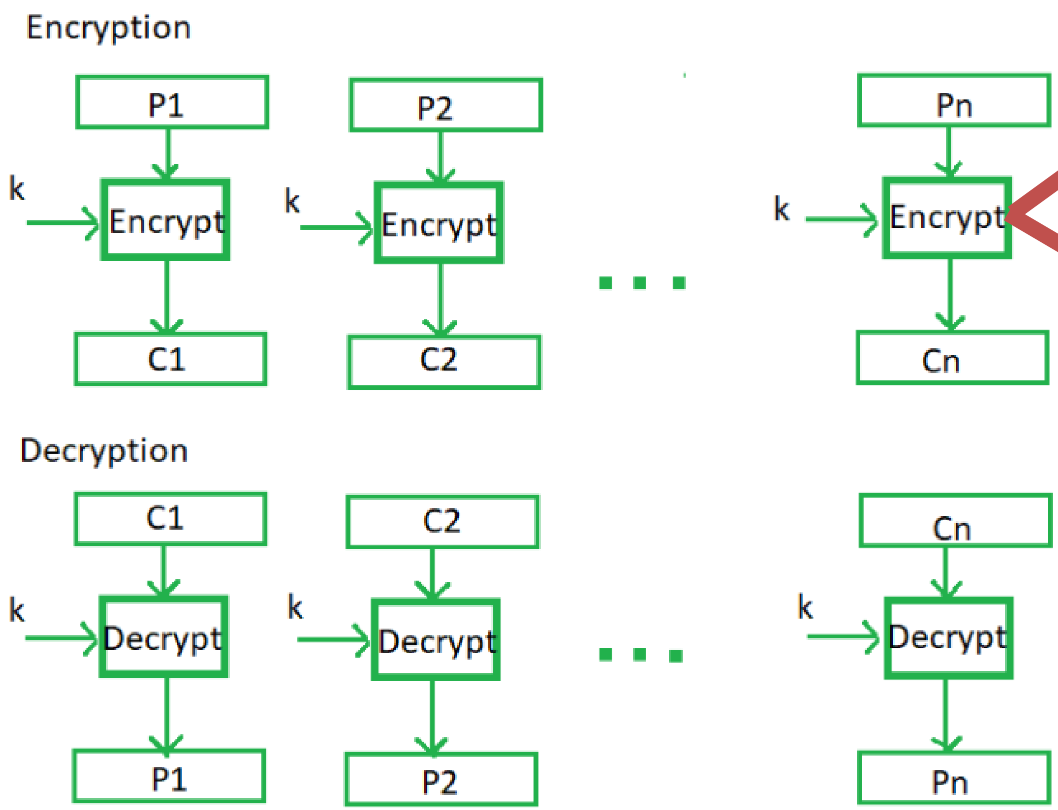
26! Possible permutations

However, we can leverage the fact that certain characters appear more commonly in the English language (a, e, i, t, r) to make guessing *much* easier

(**frequency analysis**)

Block Cipher

Plaintext is divided into n-sized blocks and each block is encrypted independently



3-bit Block Cipher Table

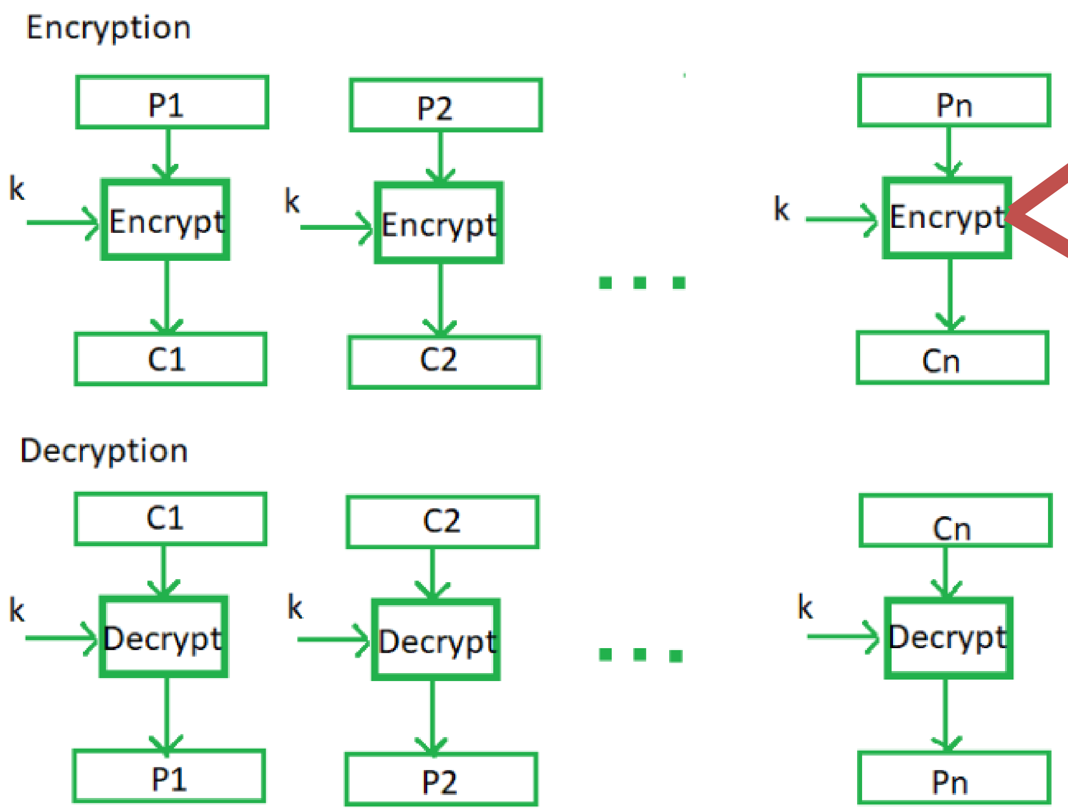
Input	output
000	110
001	111
010	101
011	100
100	011
101	010
110	000
111	001

Key!

010000111→

Block Cipher

Plaintext is divided into n-sized blocks and each block is encrypted independently



3-bit Block Cipher Table

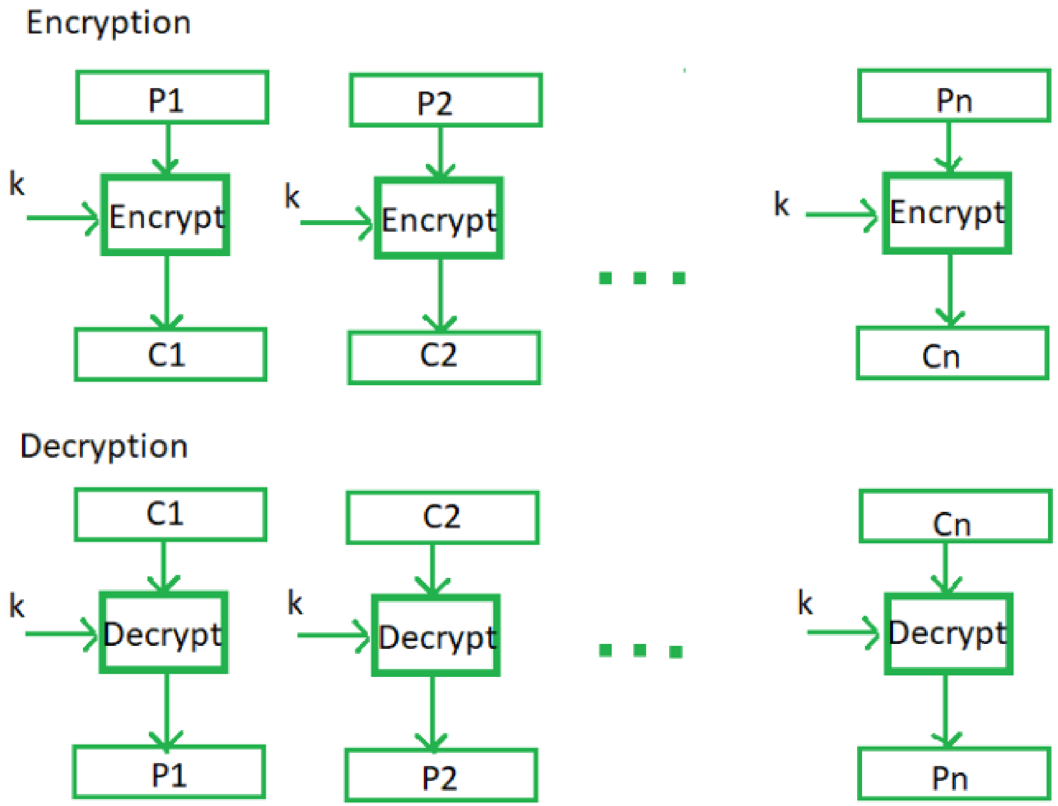
Input	output
000	110
001	111
010	101
011	100
100	011
101	010
110	000
111	001

Key!

010000111→101110001

Block Cipher

Plaintext is divided into n-sized blocks and each block is encrypted independently



010000111→101110001

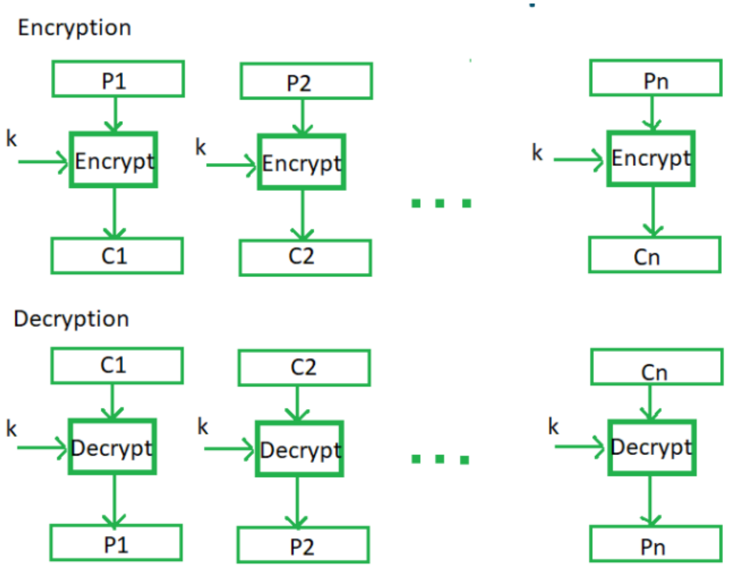
Typically, the block sizes are going to be 64 bits or even larger

# of mappings
general formula: **2^k!**

Important Properties

- Even small differences in plaintext result in different ciphertexts
- Blocks in plaintext that are the same will also have matching ciphertexts

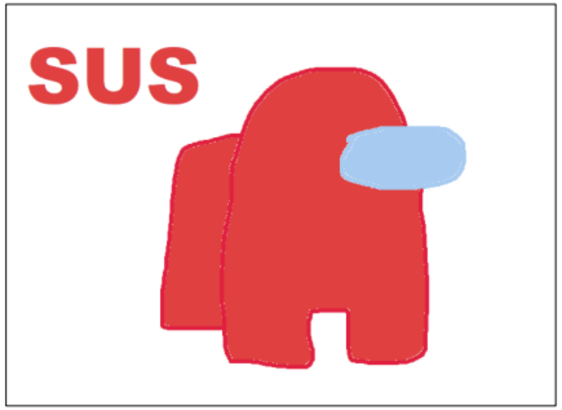
Block Cipher

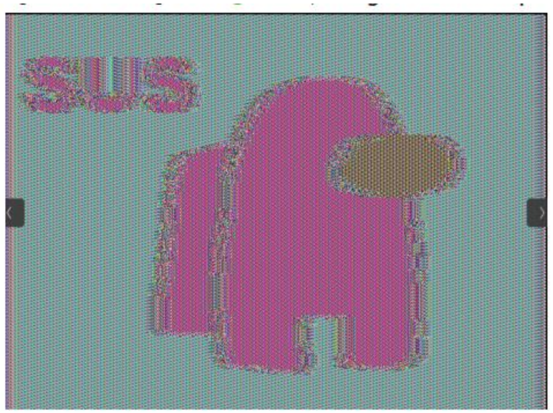


Important Properties

- Even small differences in plaintext result in different ciphertexts
- Blocks in plaintext that are the same will also have matching ciphertexts

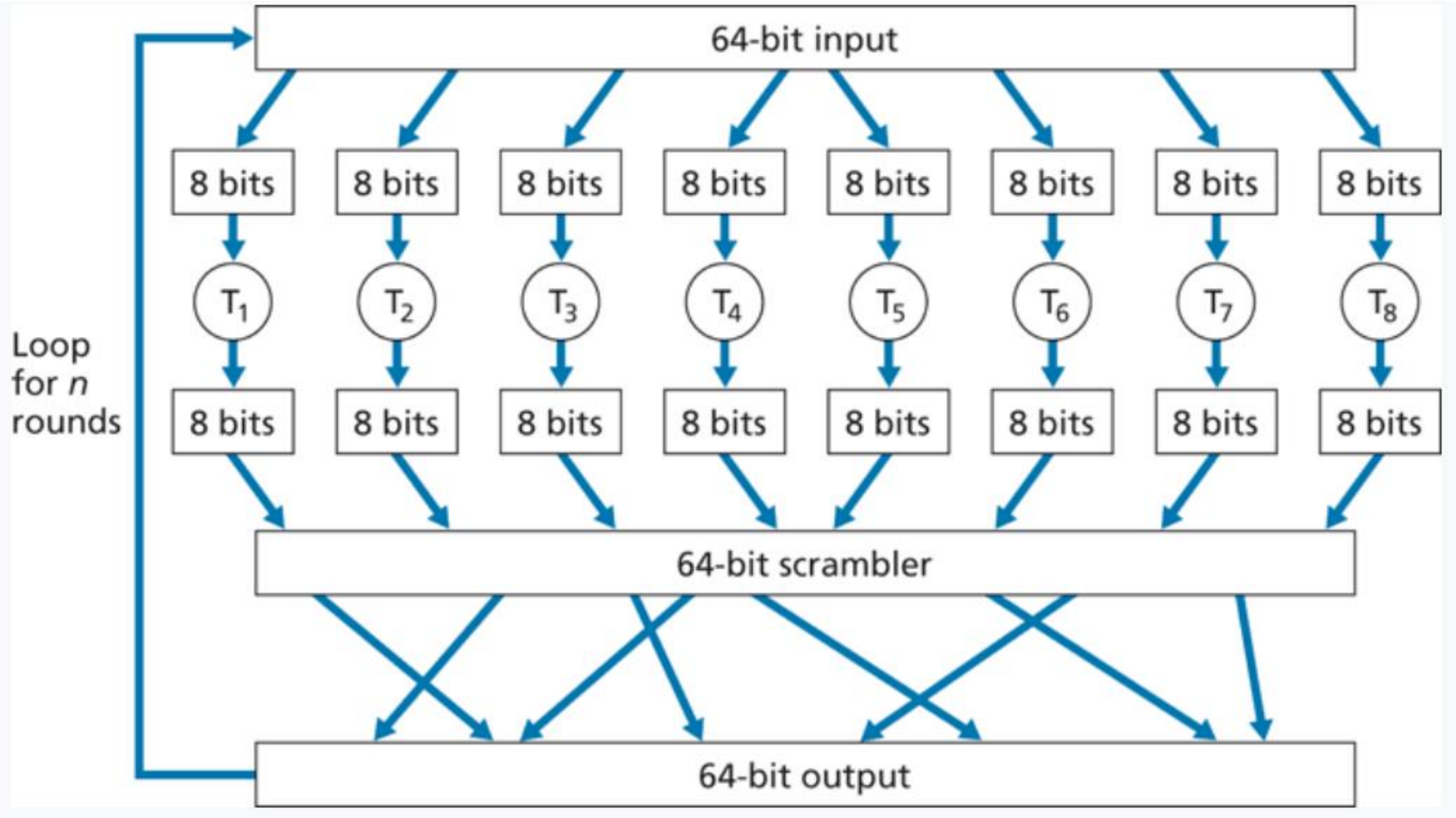
If identical keys are used:





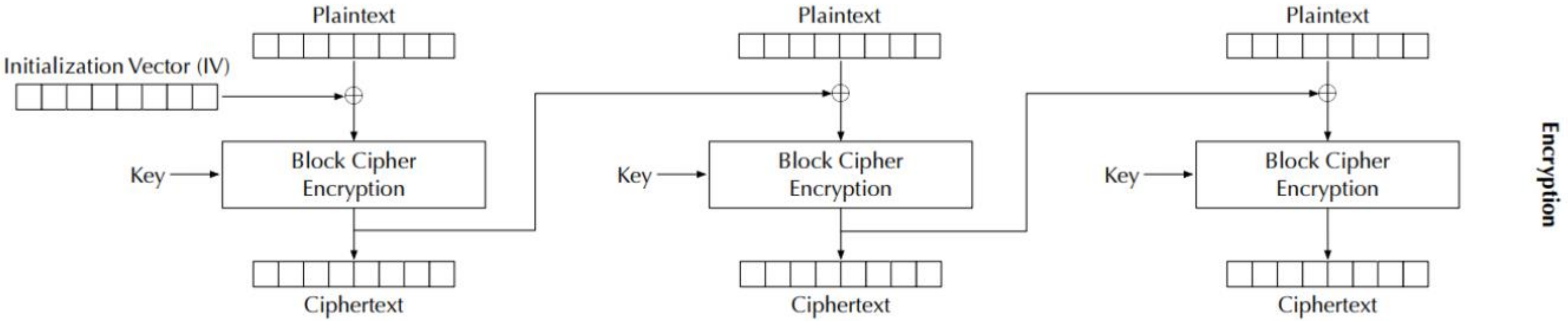
Think about storing table information for 64 block size ☹

Block Cipher



Block Cipher

Cipher Block Chaining (CBC) Mode



Rather than using predetermined tables, block ciphers usually use some type of **function** that simulate randomly permutated tables

Introduces **block dependency** $\boxed{C_i = E_K(P_i \oplus C_{i-1})}$

Introduces an **initialization vector (IV)** to ensure that even if two plaintexts are identical, their ciphertexts are still different because different IVs will be used

Block Cipher

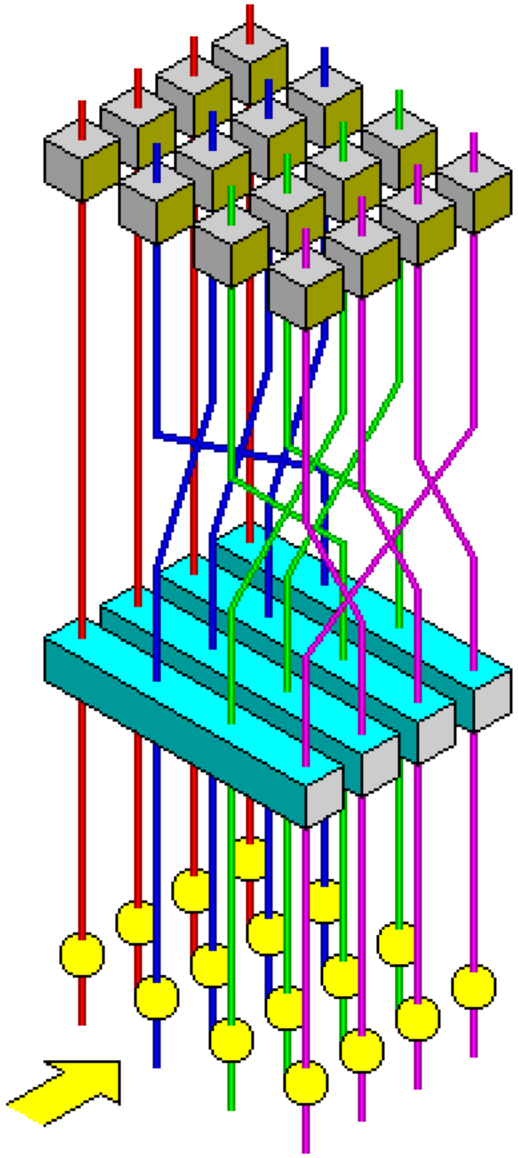
The two most common block ciphers are Advanced Encryption Standard (**AES**) and Data Encryption Standard (**DES**)

AES (2001)	DES (1977)
Keys can be of length 128, 192, and 256	Key length can only be 56
Number of rounds: 10, 12, or 14	Number of rounds: 16
Very secure (no known attacks)	Broken (can be cracked easily)
Can encrypt 128 bits of plaintext	Can encrypt 64 bits of plaintext
Faster that DES	Slower than AES

(AES is the most widely used symmetric block cipher algorithm nowadays, DES should never be used)

AES

AES is a rather complicated algorithm (for good reason), you can read more about it on your own

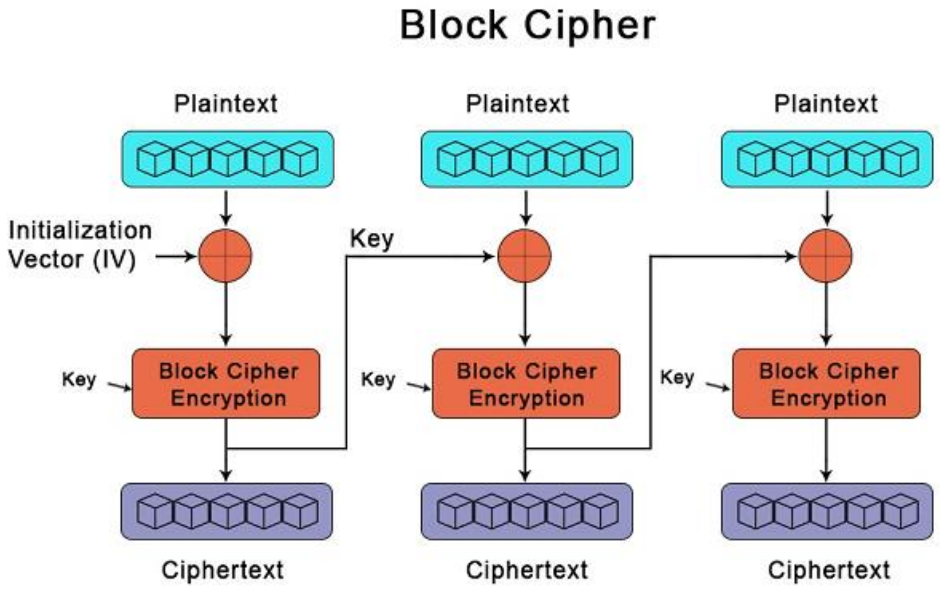


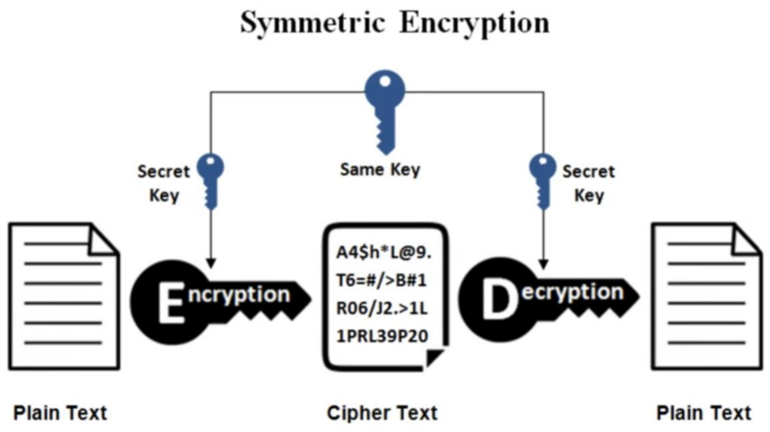
Several rounds of encrypting (“shifting”), using something like CBC

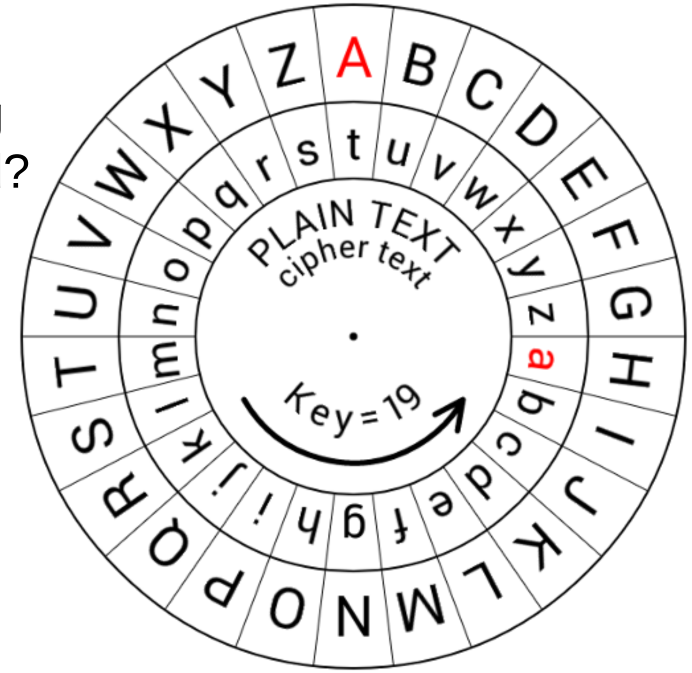
Symmetric key encryption uses the same, **shared**, key for encrypting and decrypting

What is the one major hurdle we have not discussed yet?

How do the keys get sent without being intercepted? Do the keys get encrypted?







Asymmetric Cryptography

AKA **Public key Cryptography**

The keys used for encrypting and decrypting data are *different*

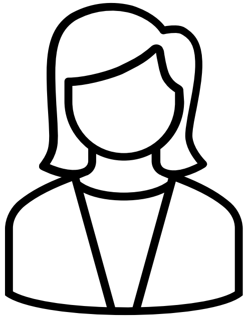
Additionally, each user now gets two-keys. A **public key**, and a **private key**

This involves some complicated math, and I won't go super deep into it. YouTube videos can explain it much better than I can

RSA (Rivest–Shamir–Adleman) is the most popular public key cryptosystem. We rely on it whenever we do communicate securely on the internet

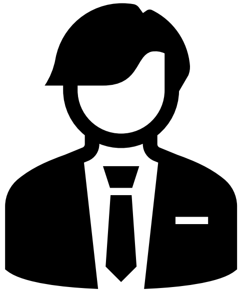
Asymmetric Cryptography (RSA)

Alice



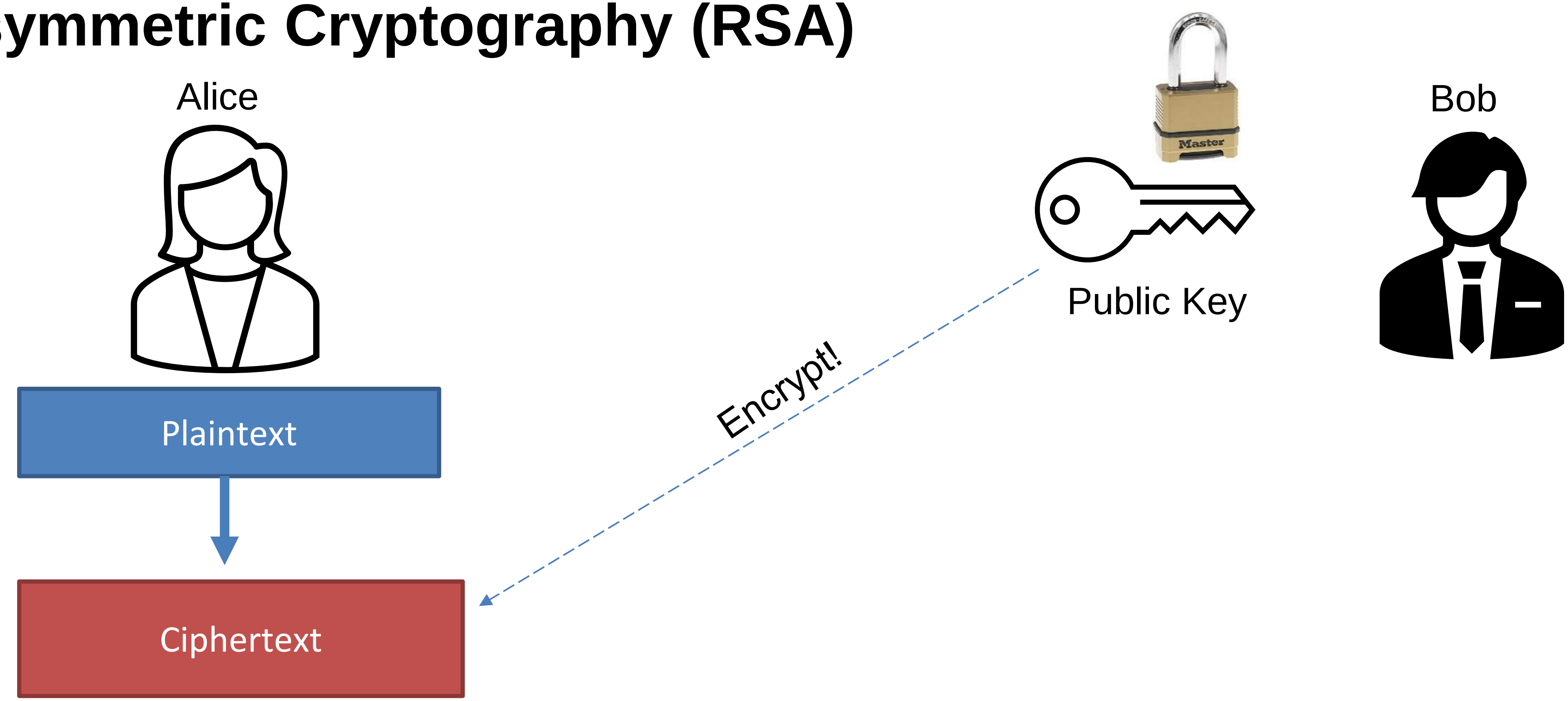
Plaintext

Bob



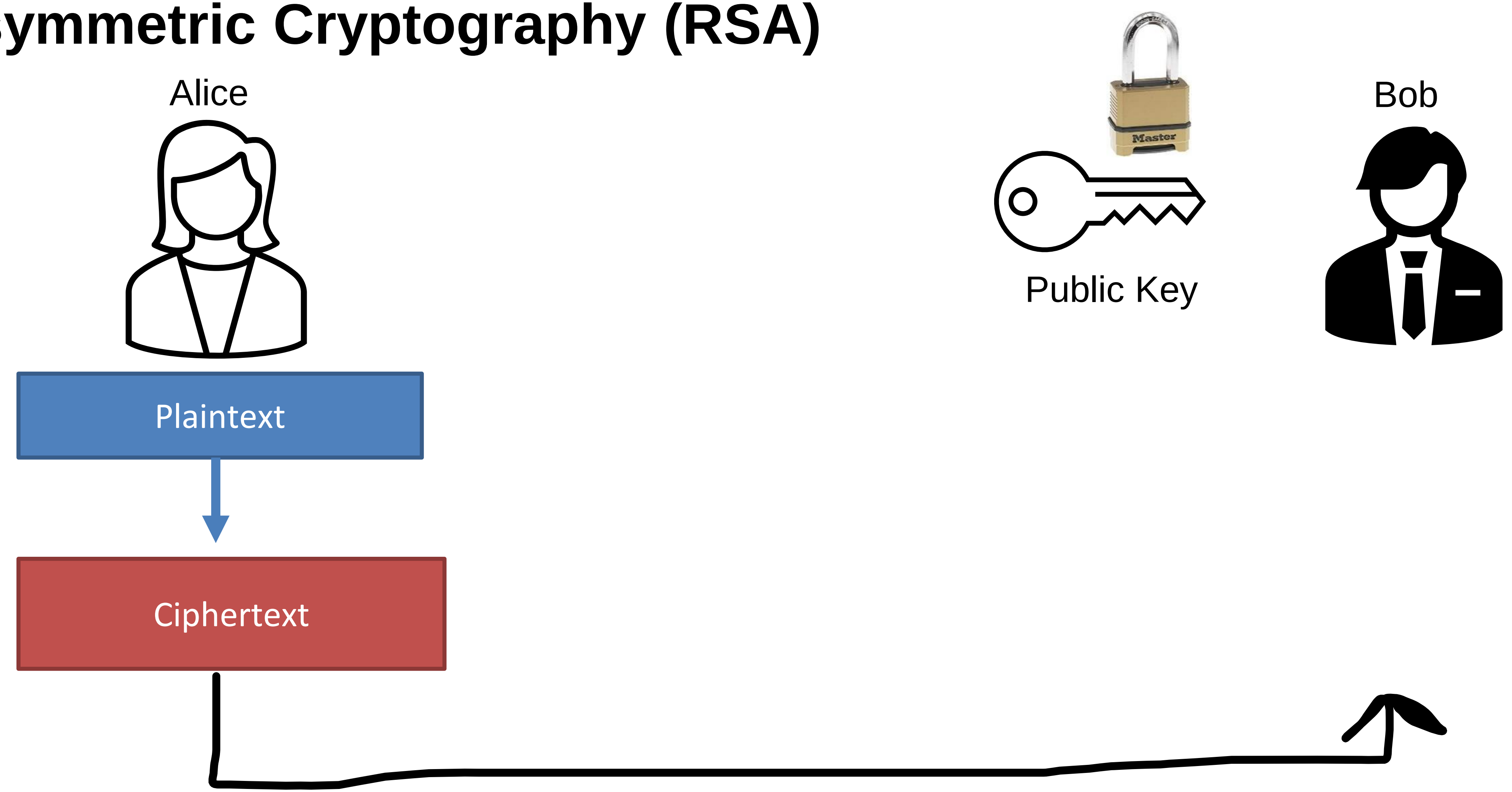
Alice has a plaintext that she wants to send to bob

Asymmetric Cryptography (RSA)

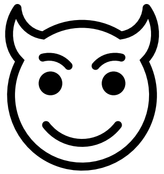


She uses Bob’s **public key** to encrypt her message

Asymmetric Cryptography (RSA)

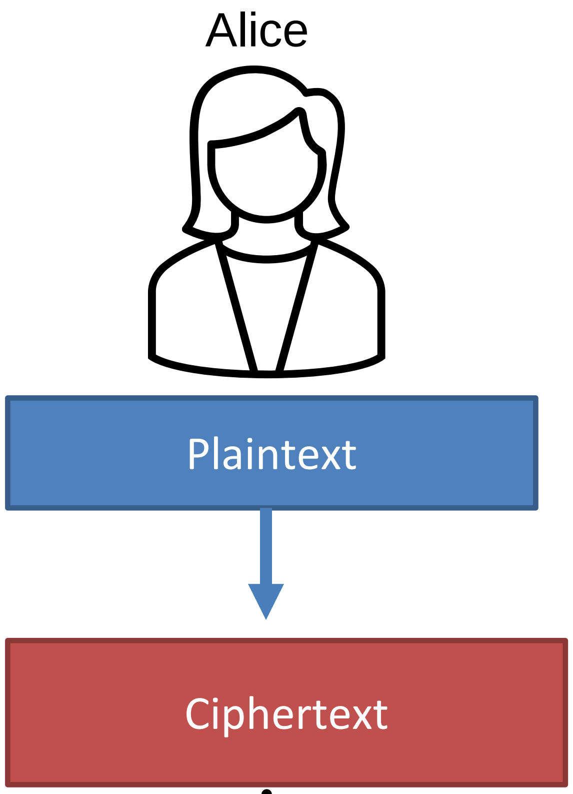


Ciphertext is sent over some medium

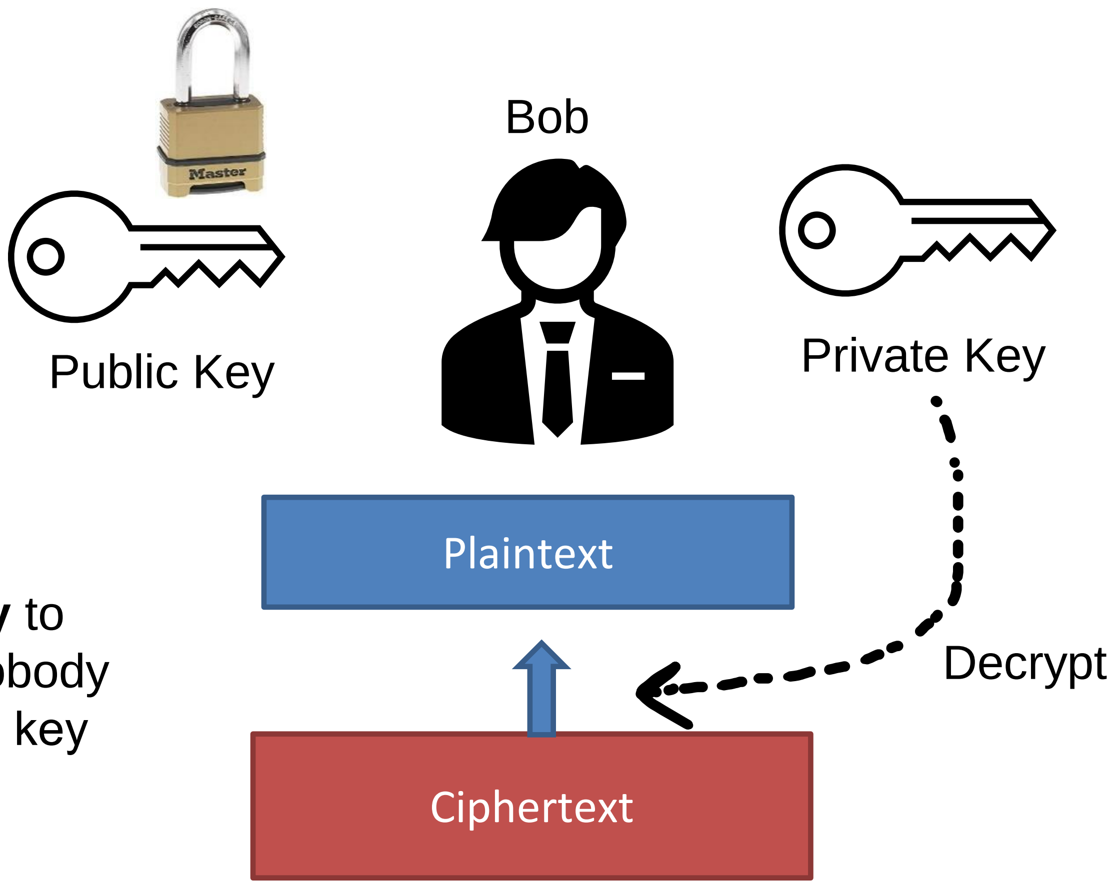


Eve can intercept this message,
But **can't** decrypt it (public key is not used for decrypting!)

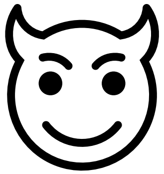
Asymmetric Cryptography (RSA)



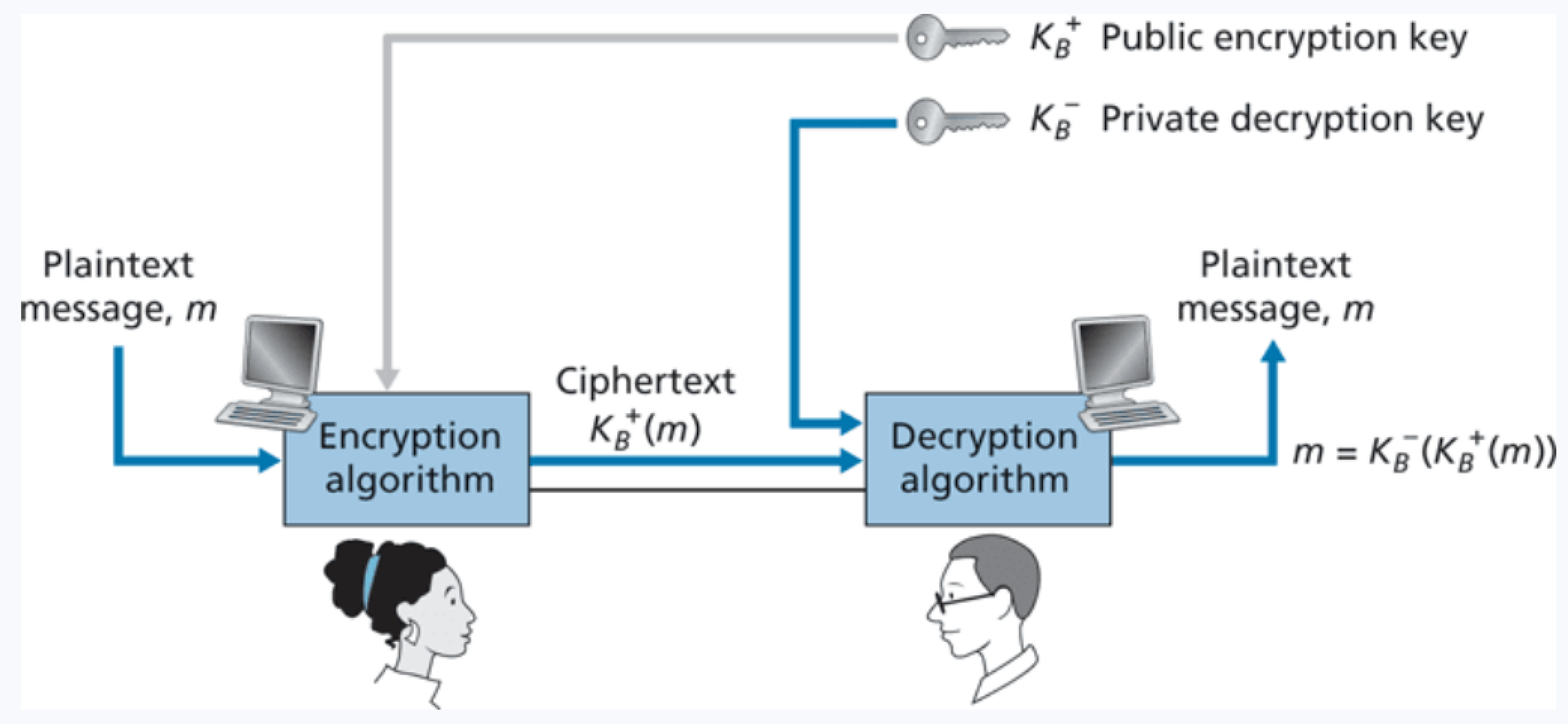
Bob uses his **private key** to decrypt the message. Nobody else knows Bob’s private key



Ciphertext is sent over some medium



Eve can intercept this message,
But **can’t** decrypt it (public key is not used for decrypting!)


K_B^+ Public encryption key
K_B^- Private decryption key
Plaintext message, m
Encryption algorithm
Ciphertext $K_B^+(m)$
Decryption algorithm
Plaintext message, m
$m = K_B^-(K_B^+(m))$

Asymmetric Cryptography (RSA)

If you multiply two prime numbers (**p** and **q**) together, the product can only be divisible by those two number

5183

??? * ??? = 5183

This is very difficult to figure out for the people that don't know p or q

In fact, there is not an *efficient* program that can calculate the factors of integers

Remember what these are called?

Asymmetric Cryptography (RSA)

If you multiply two prime numbers (**p** and **q**) together, the product can only be divisible by those two number

5183

??? * ??? = 5183

This is very difficult to figure out for the people that don't know p or q

In fact, there is not an *efficient* program that can calculate the factors of integers

This problem is in NP

Asymmetric Cryptography (RSA)

If you multiply two prime numbers (**p** and **q**) together, the product can only be divisible by those two number

RSA is based on large numbers that are difficult to factorize
The public and private keys are derived from these prime numbers

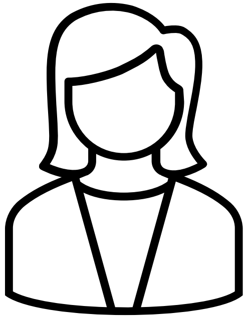
How long should RSA keys be? 1024 or 2048 bits long!

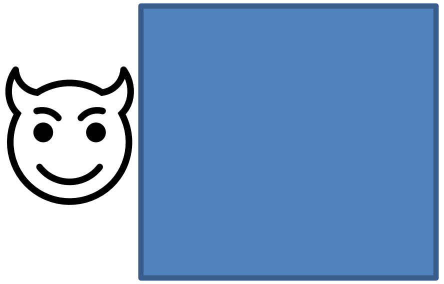
The longer the key = the more difficult to crack (exponentially)

Asymmetric Cryptography (RSA)

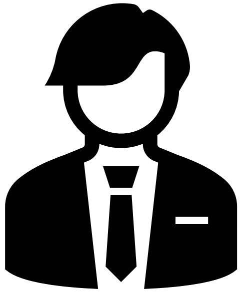
Eve's stolen goods

Alice





Bob



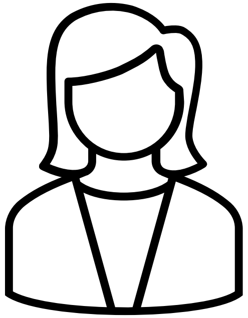
p = 53
q = 59

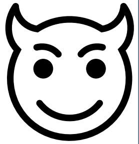
Step 1: Choose two large primer numbers, p and q

Asymmetric Cryptography (RSA)

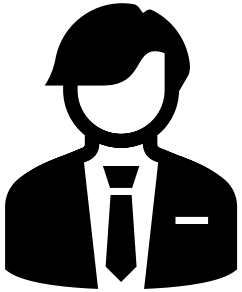
Eve’s stolen goods

Alice





Bob



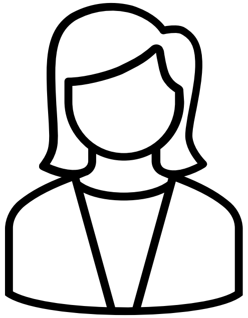
p = 53
q = 59
n = 3127

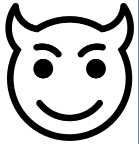
Step 1: Choose two large primer numbers, p and q
Step 2: Calculate the product n

Asymmetric Cryptography (RSA)

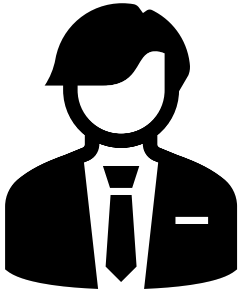
Eve's stolen goods

Alice

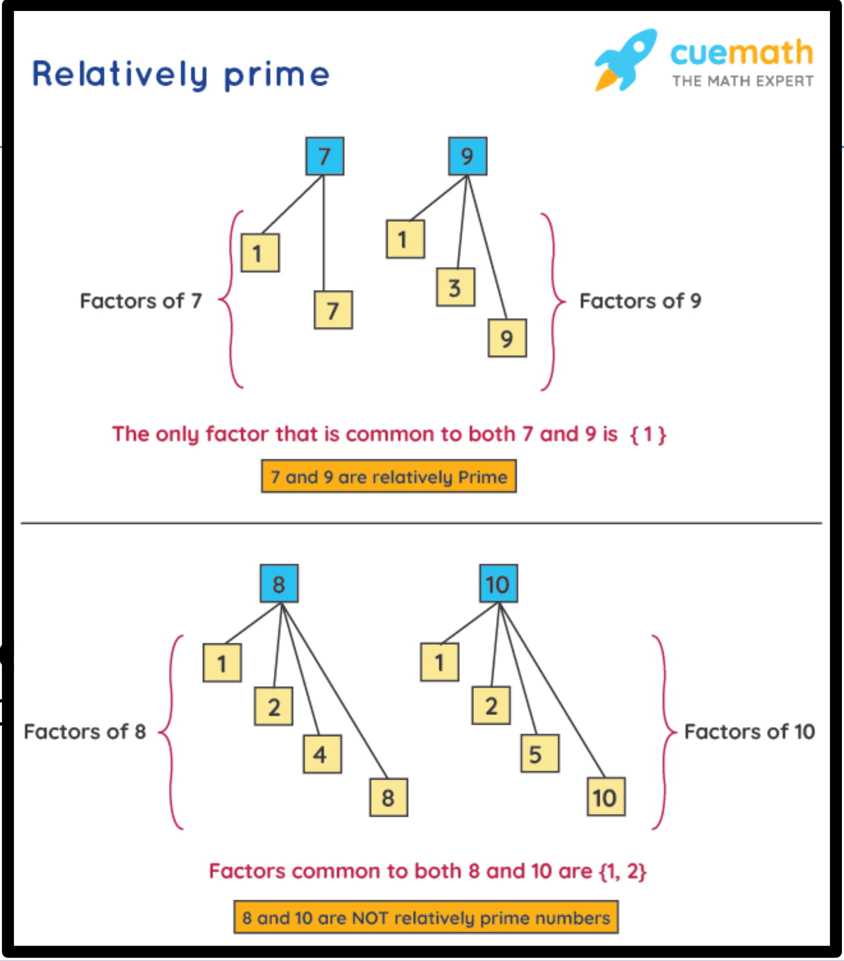




Bob



p = 53
q = 59
n = 3127



Φ(n) = number of values less than n which are *relatively prime* to n

Step 1: Choose two larg
Step 2: Calculate the pro
Step 3: Calculate **Φ(n)**

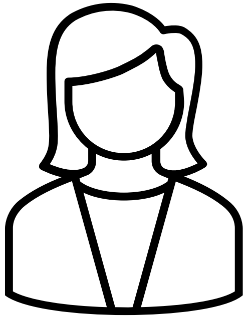
1
2
3
…
3125
3126

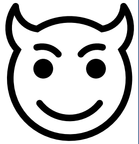
How many of these numbers are relatively prime w/ 3127?

Asymmetric Cryptography (RSA)

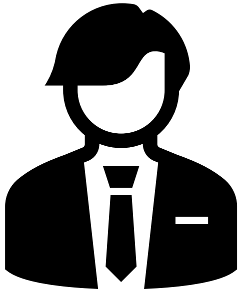
Eve's stolen goods

Alice

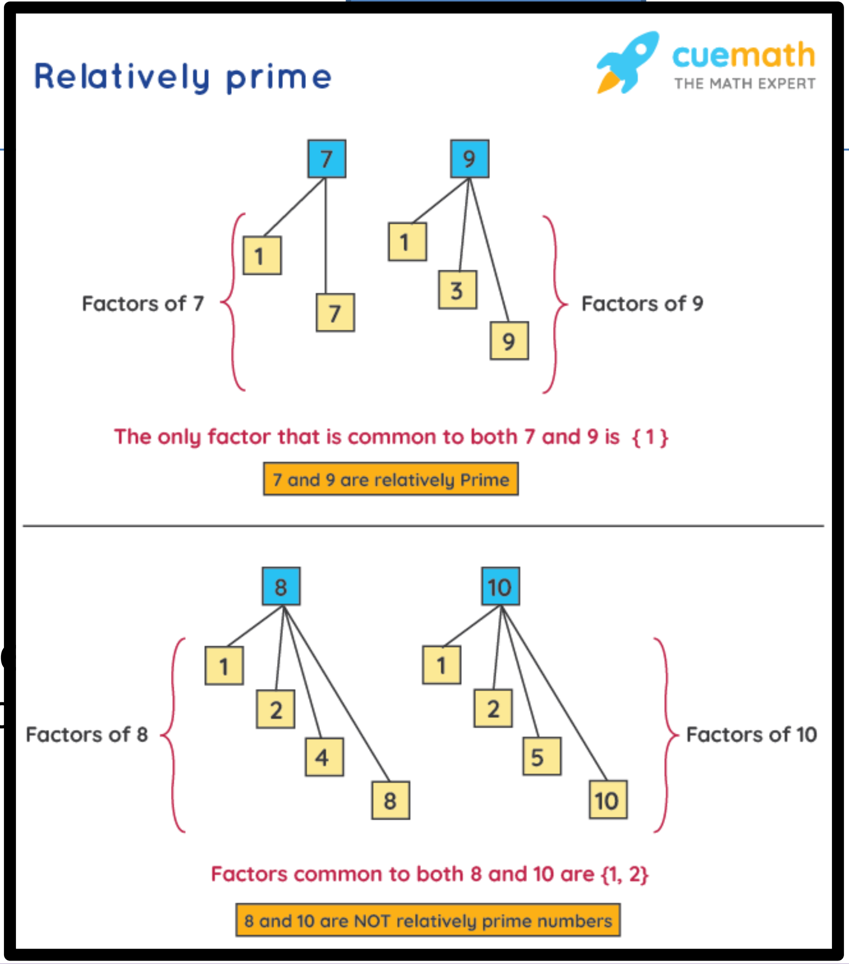




Bob



$p = 53$
$q = 59$
$n = 3127$



$\Phi(n)$ = number of values less than n which are *relatively prime* to n

Step 1: Choose two larg
Step 2: Calculate the pr
Step 3: Calculate **Φ(n)**

1
2
3
…
3125
3126

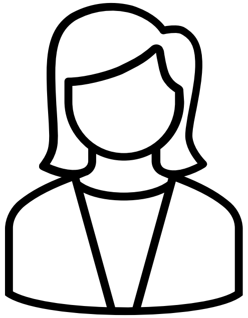
How many of these numbers are relatively prime w/ 3127?

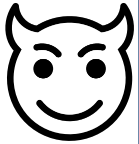
Difficult.. But very easy for the product of two prime #s!

Asymmetric Cryptography (RSA)

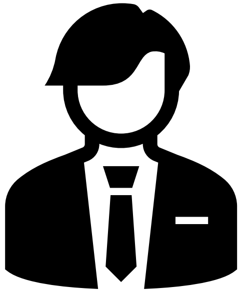
Eve's stolen goods

Alice





Bob



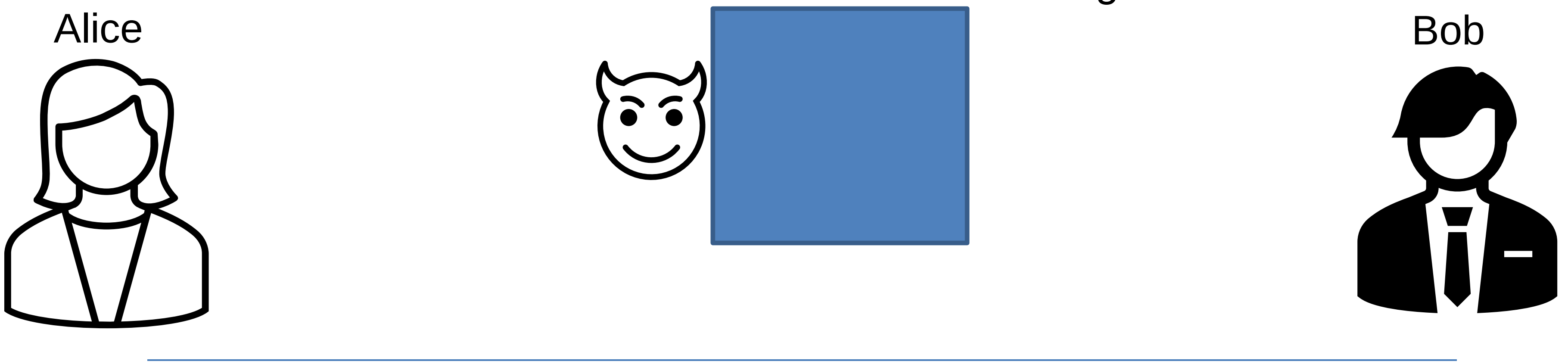
p = 53
q = 59
n = 3127

Φ(n) = number of values less than n which are *relatively prime* to n

The Φ(n) of a product of two prime numbers will always be (p-1)(q-1)

Step 1: Choose two large primer numbers, p and q
Step 2: Calculate the product n
Step 3: Calculate **Φ(n)**

Asymmetric Cryptography (RSA)



p = 53
q = 59
n = 3127
Φ(n) = 52*28 = 3016

Φ(n) = number of values less than n which are *relatively prime* to n

The Φ(n) of a product of two prime numbers will always be (p-1)(q-1)

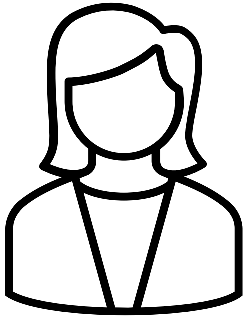
Step 1: Choose two large primer numbers, p and q
Step 2: Calculate the product n
Step 3: Calculate **Φ(n)**

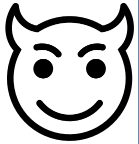
Asymmetric Cryptography (RSA)

Φ(n) = number of values less than n which are *relatively prime* to n

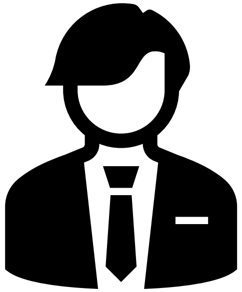
Eve's stolen goods

Alice





Bob



p = 53
q = 59
n = 3127
Φ(n) = 3016

e = 1 < e < **Φ(n)**
Not be a factor of n, but an integer

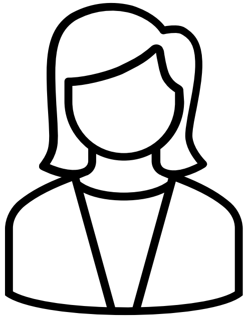
Step 1: Choose two large primer numbers, p and q
Step 2: Calculate the product n
Step 3: Calculate **Φ(n)**
Step 4: Choose public exponent **e**

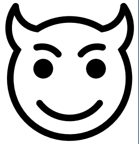
Asymmetric Cryptography (RSA)

$\Phi(n)$ = number of values less than n which are *relatively prime* to n

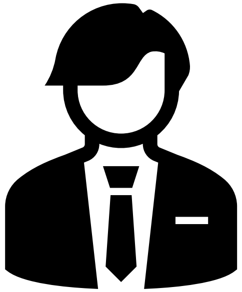
Eve's stolen goods

Alice





Bob



p = 53
q = 59
n = 3127
$\Phi(n)$ = 3016
e = 3

e = 1 < e < **$\Phi(n)$**
Not be a factor of n, but an integer

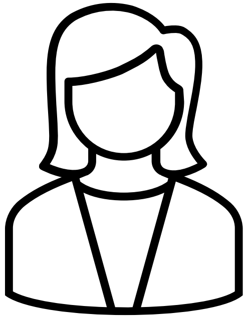
Step 1: Choose two large primer numbers, p and q
Step 2: Calculate the product n
Step 3: Calculate **$\Phi(n)$**
Step 4: Choose public exponent **e**

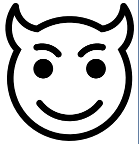
Asymmetric Cryptography (RSA)

$\Phi(n)$ = number of values less than n which are *relatively prime* to n

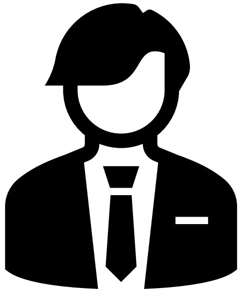
Eve's stolen goods

Alice





Bob



p = 53
q = 59
n = 3127
$\Phi(n)$ = 3016
e = 3

$$d = \frac{K * \Phi(n) + 1}{e}$$

Step 1: Choose two large primer numbers, p and q
Step 2: Calculate the product n
Step 3: Calculate **Φ(n)**
Step 4: Choose public exponent **e**
Step 5: Select private exponent d

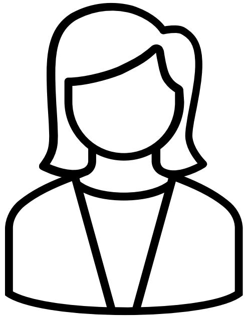
K = some integer that will make the quotient an integer

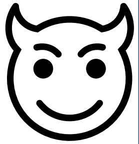
Asymmetric Cryptography (RSA)

Φ(n) = number of values less than n which are *relatively prime* to n

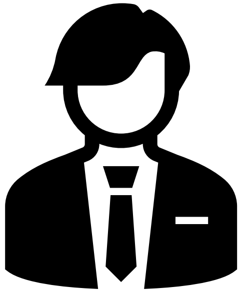
Eve's stolen goods

Alice





Bob



p = 53
q = 59
n = 3127
Φ(n) = 3016
e = 3

$$d = \frac{2 * 3016 + 1}{3}$$

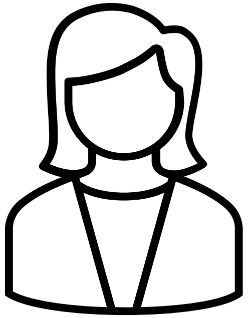
Step 1: Choose two large primer numbers, p and q
Step 2: Calculate the product n
Step 3: Calculate **Φ(n)**
Step 4: Choose public exponent **e**
Step 5: Select private exponent d

K = some integer that will make the quotient an integer

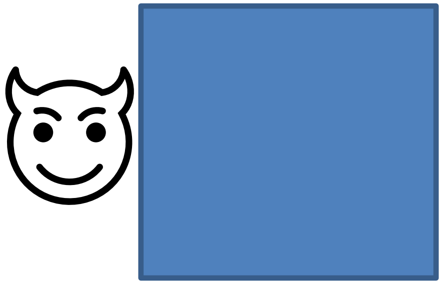
Asymmetric Cryptography (RSA)

$\Phi(n)$ = number of values less than n which are *relatively prime* to n

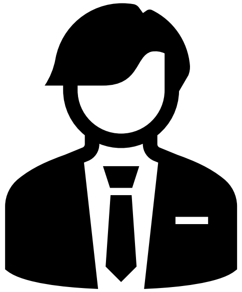
Alice



Eve's stolen goods



Bob



p = 53
q = 59
n = 3127
$\Phi(n)$ = 3016
e = 3
d = 2011

$$d = \frac{2 * 3016 + 1}{3}$$

Step 1: Choose two large primer numbers, p and q
Step 2: Calculate the product n
Step 3: Calculate **Φ(n)**
Step 4: Choose public exponent **e**
Step 5: Select private exponent d

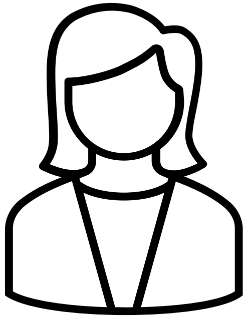
K = some integer that will make the quotient an integer

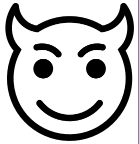
Asymmetric Cryptography (RSA)

Φ(n) = number of values less than n which are *relatively prime* to n

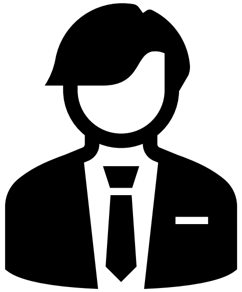
Eve's stolen goods

Alice





Bob



Alice's Public Key

n = 3127
e = 3

Secret Information

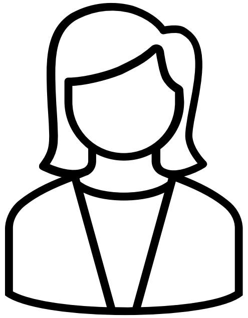
p = 53
q = 59
Φ(n) = 3016
d = 2011

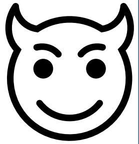
Asymmetric Cryptography (RSA)

$\Phi(n)$ = number of values less than n which are *relatively prime* to n

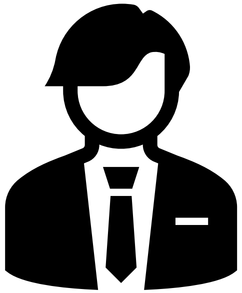
Eve's stolen goods

Alice





Bob



Alice's Public Key

n = 3127
e = 3

Bob has a message to send to Alice

HI → 89

Message must be converted into a number

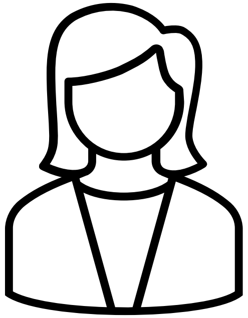
Secret Information

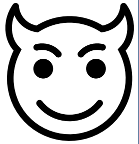
p = 53
q = 59
$\Phi(n)$ = 3016
d = 2011

Asymmetric Cryptography (RSA)

$\Phi(n)$ = number of values less than n which are *relatively prime* to n

Alice

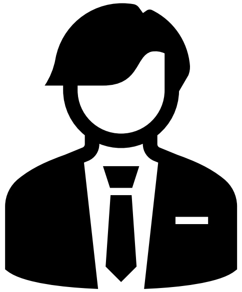




Eve's stolen goods

n = 3127
e = 3

Bob



Alice's Public Key

n = 3127
e = 3

Bob has a message to send to Alice

89

Use Alice's Public Key to encrypt

$m^e \bmod 3127$

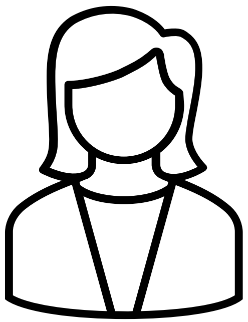
Secret Information

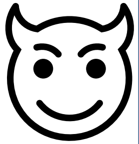
p = 53
q = 59
$\Phi(n)$ = 3016
d = 2011

Asymmetric Cryptography (RSA)

$\Phi(n)$ = number of values less than n which are *relatively prime* to n

Alice

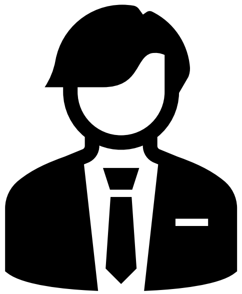




Eve's stolen goods

n = 3127
e = 3

Bob



Alice's Public Key

n = 3127
e = 3

Bob has a message to send to Alice

89

Use Alice's Public Key to encrypt

$89^3 \bmod 3127$

$C = 1394$

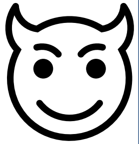
Secret Information

p = 53
q = 59
$\Phi(n)$ = 3016
d = 2011

Asymmetric Cryptography (RSA)

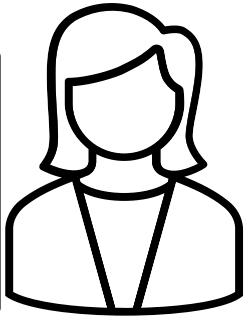
Φ(n) = number of values less than n which are *relatively prime* to n

Eve's stolen goods



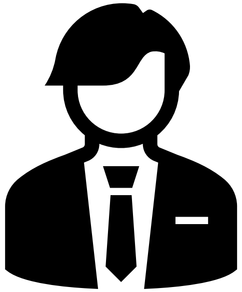
n = 3127
e = 3
c=1394

Alice



p = 53
q = 59
Φ(n) = 3016
d = 2011

Bob



Alice's Public Key

n = 3127
e = 3

Bob has a message to send to Alice

89

Use Alice's Public Key to encrypt

$89^3 \bmod 3127$

$C = 1394$

$1394^{2011} \bmod 3127$

Alice decrypts message using her private key

Asymmetric Cryptography (RSA)

$\Phi(n)$ = number of values less than n which are *relatively prime* to n

Alice

p = 53
q = 59
$\Phi(n) = 3016$
d = 2011

Eve's stolen goods

n = 3127
e = 3
c=1394

Bob

Alice's Public Key

n = 3127
e = 3

Bob has a message to send to Alice

89

Use Alice's Public Key to encrypt

$89^3 \bmod 3127$

$C = 1394$

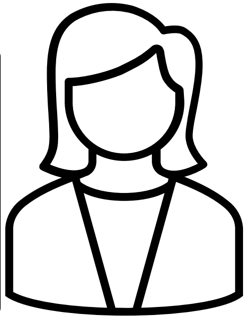
$1394^{2011} \bmod 3127 = \mathbf{89}$ ✓

Alice decrypts message using her private key

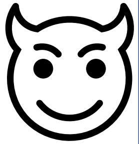
Asymmetric Cryptography (RSA)

$\Phi(n)$ = number of values less than n which are *relatively prime* to n

Alice



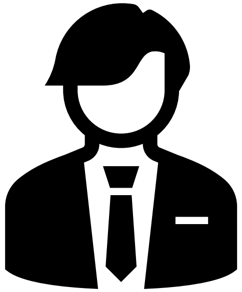
p = 53
q = 59
$\Phi(n)$ = 3016
d = 2011



Eve's stolen goods

n = 3127
e = 3
c=1394

Bob



Alice's Public Key

n = 3127
e = 3

Bob has a message to send to Alice

89

Alice's Private Key

n = 3127
d = 2011

What does eve know??

$???^3 \bmod 3127 = 1394$

These is very difficult to figure out, since she does not know the factorization of n

Use Alice's Public Key to encrypt

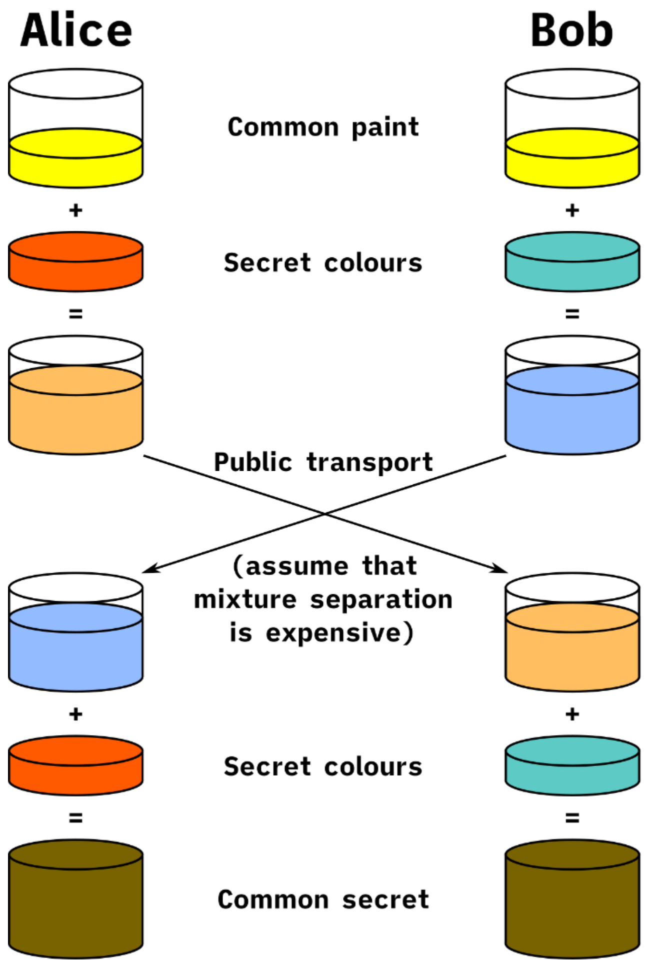
$89^3 \bmod 3127$

$C = 1394$

Asymmetric Cryptography (RSA)

We now have a method for sending secure messages over a possibly unsecure channel!

Limitation of RSA: Can only encrypted data that is smaller or equal to key length (< 2048 bits)

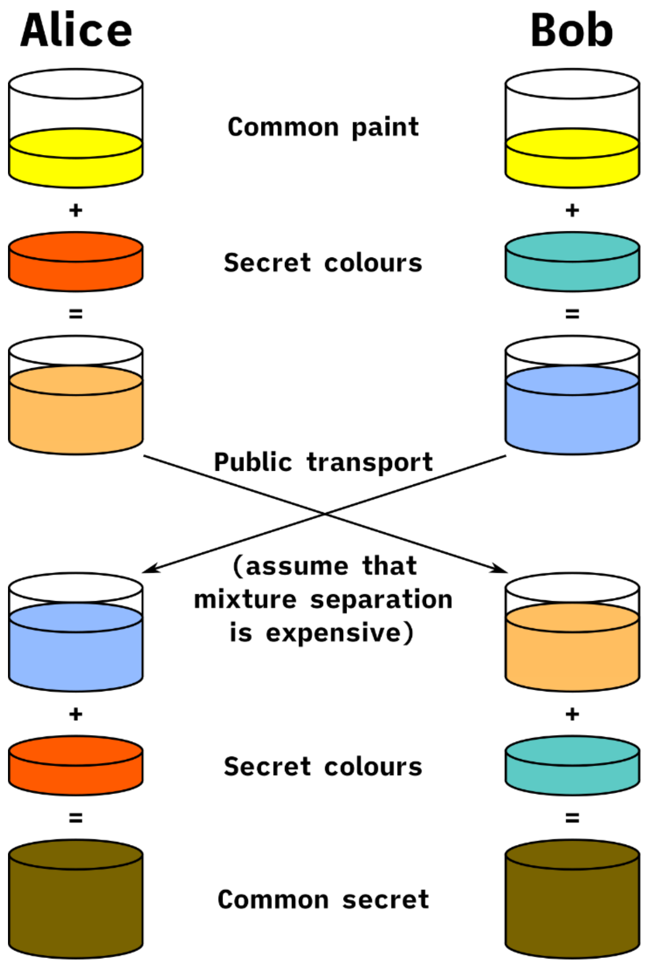


Asymmetric Cryptography (RSA)

We now have a method for sending secure messages over a possibly unsecure channel!

Limitation of RSA: Can only encrypted data that is smaller or equal to key length (< 2048 bits)

What could we encrypt instead??



Asymmetric Cryptography (RSA)

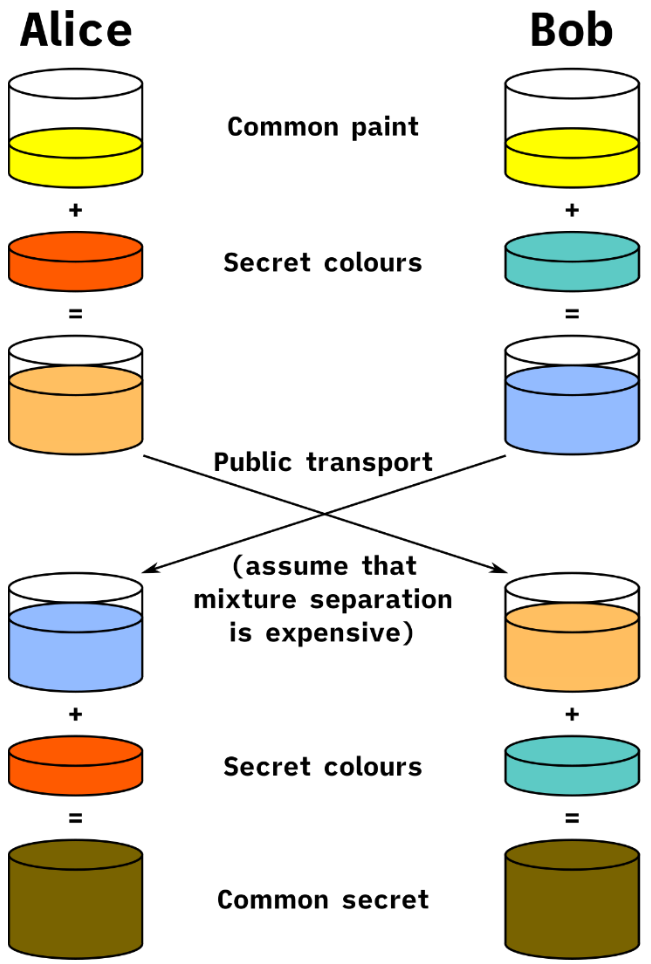
We now have a method for sending secure messages over a possibly unsecure channel!

Limitation of RSA: Can only encrypted data that is smaller or equal to key length (< 2048 bits)

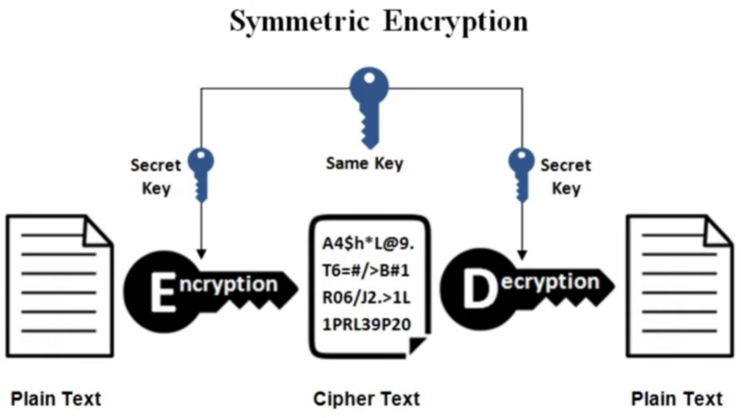
What could we encrypt instead??

The key for a symmetric cryptography algorithm! (< 2048 bits)

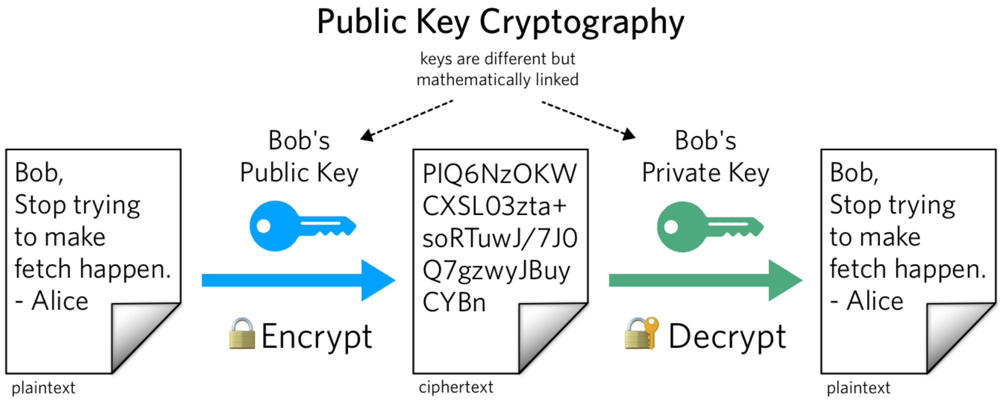
RSA can be used to generate **session key,** which can then be used to encrypt many chunks of data (also much faster then asymmetric crypto)



Review



- Same key used for encrypting and decrypting
- Using block ciphers (AES), we can encrypt an arbitrary size of data
- Issue: How to securely share secret keys with each other?



- Two keys: Public Key (a lock), and a price key (the key)
- Public key is used to encrypt. Private key used to decrypt message
- Using math, we can securely send messages over an unsecure channel without sharing any sensitive information
- Issue: We can not encrypt stuff bigger than our key (2048 bits)

- symmetric and asymmetric cryptography are used **together**

(use RSA to send the key for symmetric crypto!)