

# COMP435: *SECURITY CONCEPTS!*

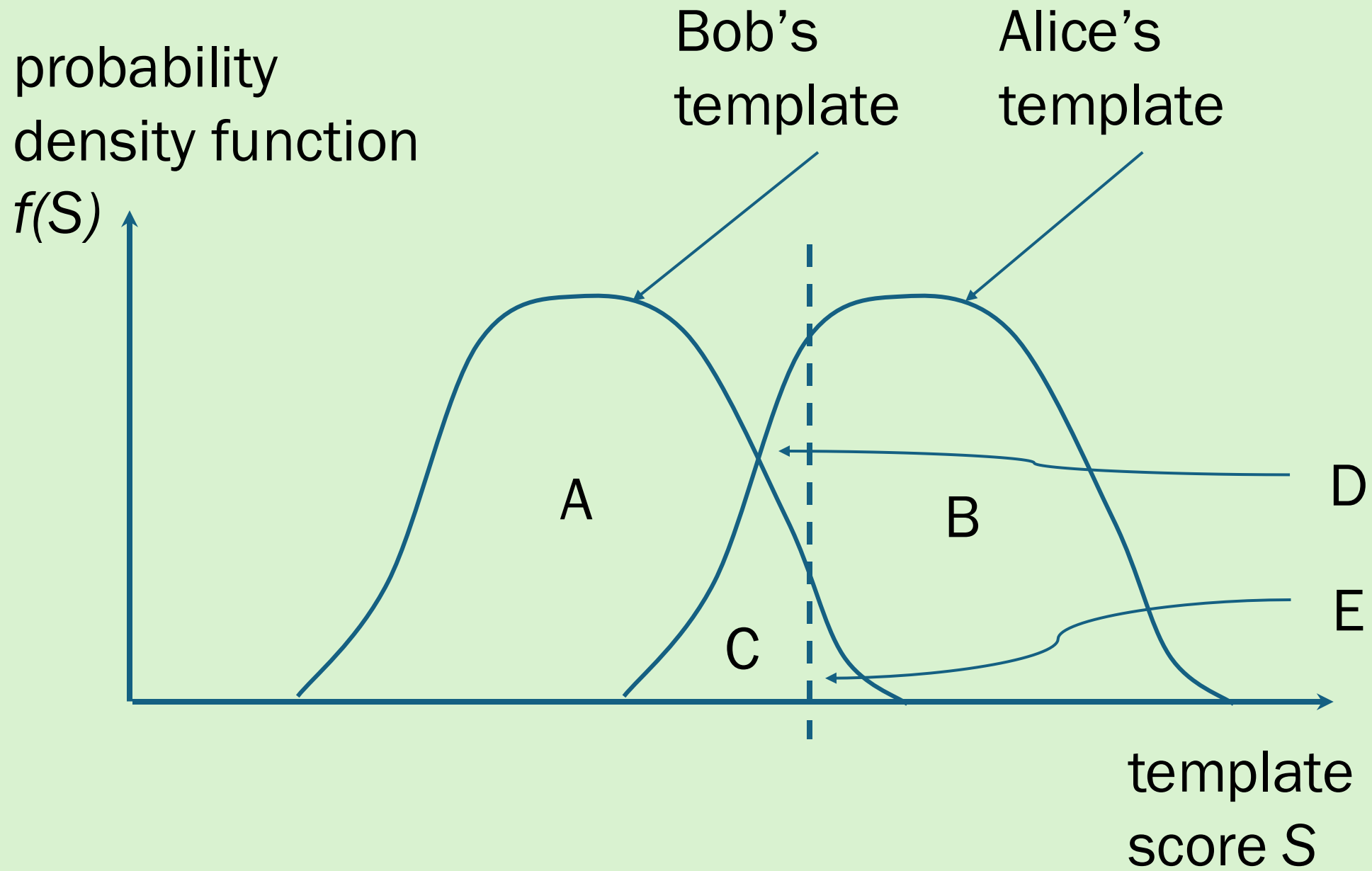
Lecture 6: Finish Authentication, Start  
Crypto!

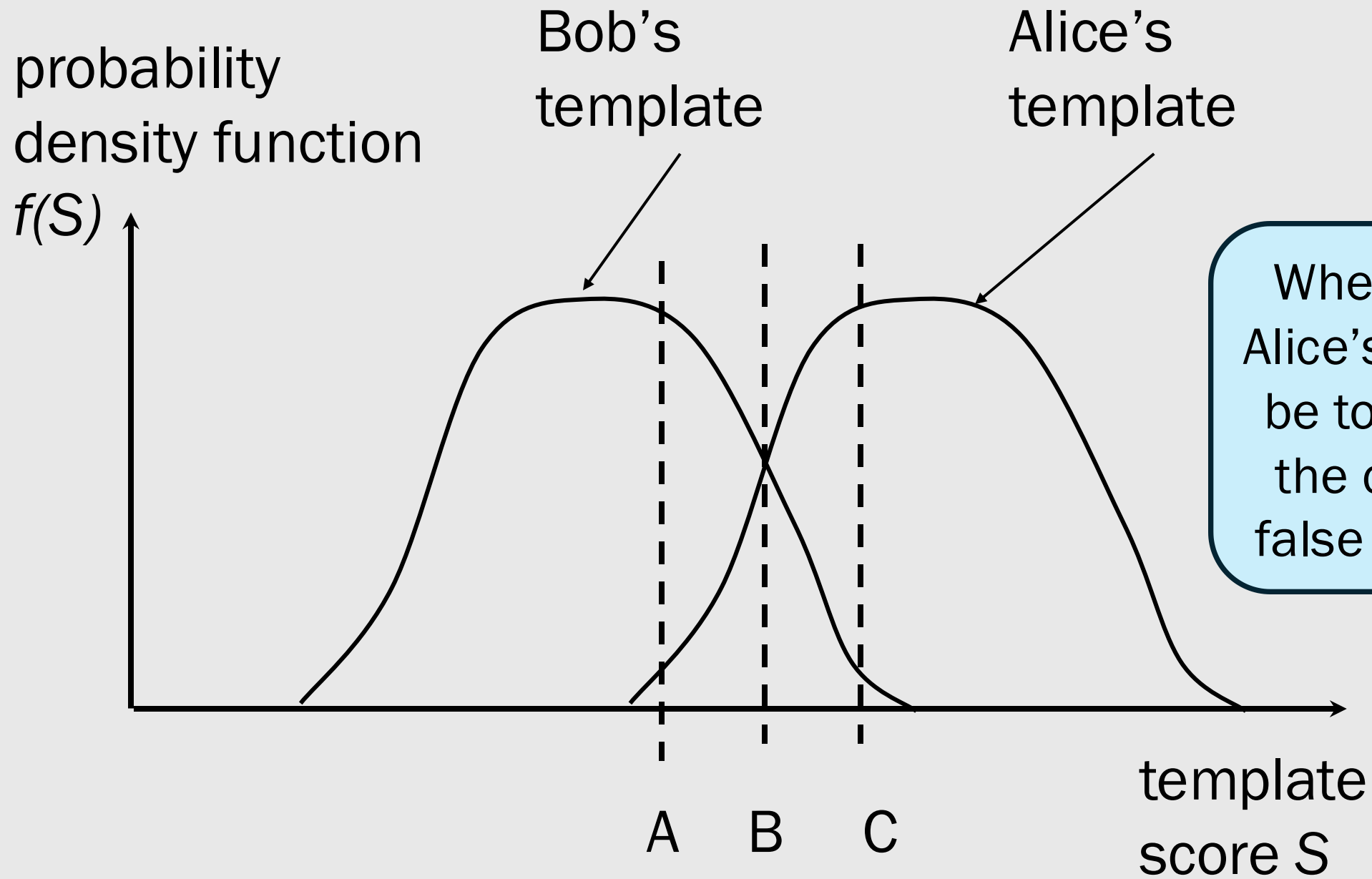
[tinyurl.com/comp435-fa25](https://tinyurl.com/comp435-fa25)



# BIOMETRICS







Where should Alice's threshold be to minimize the chance of false positives?



WHAT YOU HAVE, TWO-  
FACTOR AUTHENTICATION



# Means of Authentication

What you  
know

What you do  
or what you  
are

What you  
have

Where you  
are

tokens

# Authentication by what you have

Def'n: Possession of an item is sufficient for authentication

E.g.,

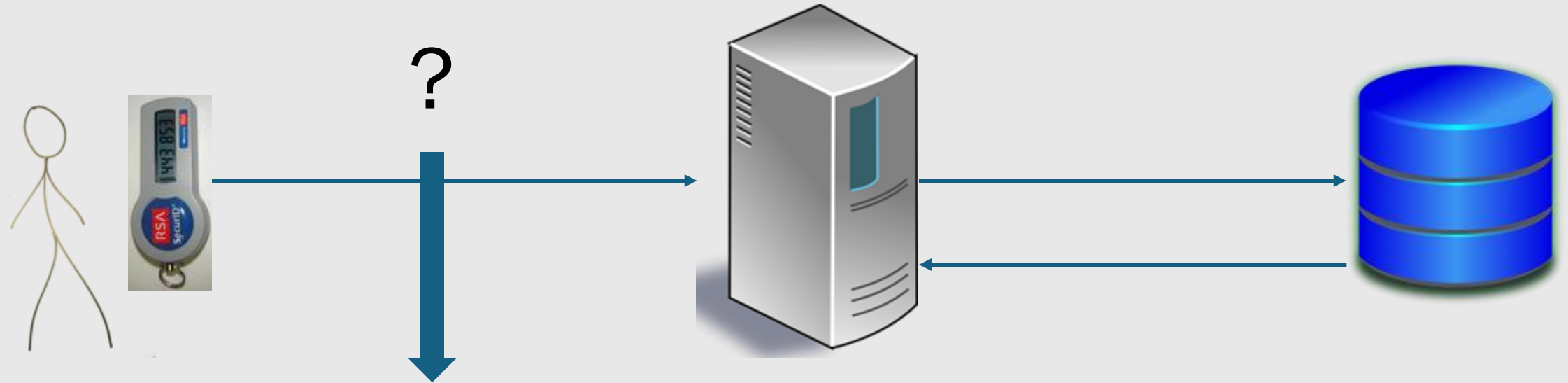
- keys
- credit card
- hardware token

# Hardware Tokens





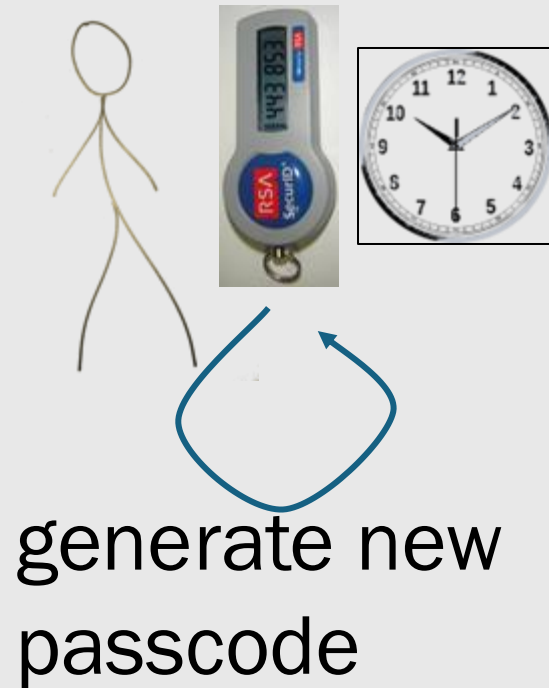
# Demonstrating Possession Remotely



- Challenge-Response
- Synchronized Passcodes

The Challenge-Response piece involves some crypto... We will revisit!

# Synchronized Passcodes



passcode



# Two-factor Authentication

Def'n: require two independent means of authentication

E.g., Driver's license:

- license: a physical token

- picture: biometric authentication

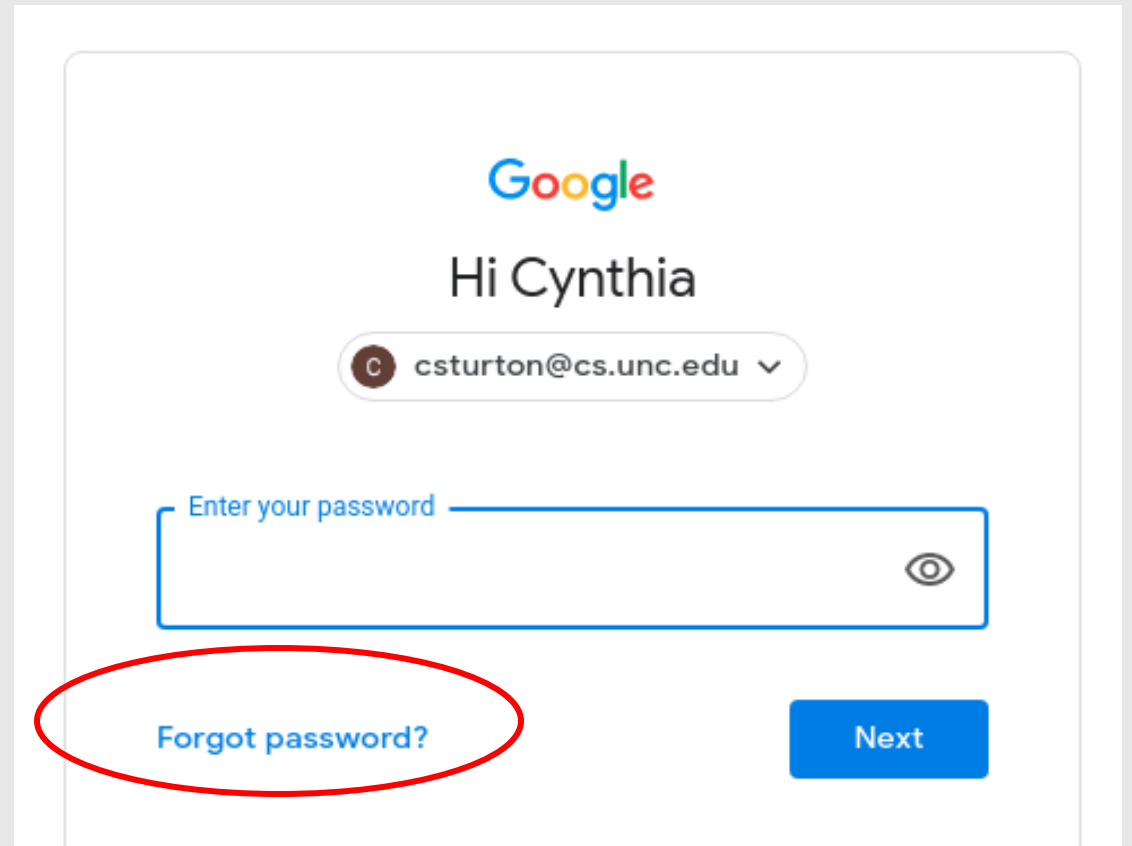


# REVOCATION & RESETTING

# Revocation

Def'n: cancelling a means of authentication

- Passwords
- Biometrics
- Tokens



The image shows a Google login interface. At the top is the Google logo. Below it, the text "Hi Cynthia" is displayed. Underneath is a rounded rectangle containing a small circular profile icon with the letter 'C' and the email address "csturton@cs.unc.edu" followed by a downward arrow. Below this is a password input field with the placeholder text "Enter your password" and a toggle icon (an eye) on the right. At the bottom left, the text "Forgot password?" is circled in red. At the bottom right is a blue button labeled "Next".

# Resetting a Password

1. Authenticate as user
2. Reset password

# Q&A

Question 1 of 5

What is your favorite pizza topping? ▼

Question 2 of 5

Select question\* ▼

Question 3 of 5

Select question\* ▼

Question 4 of 5

Select question\* ▼


Question 5 of 5

Select your answer\* ▼

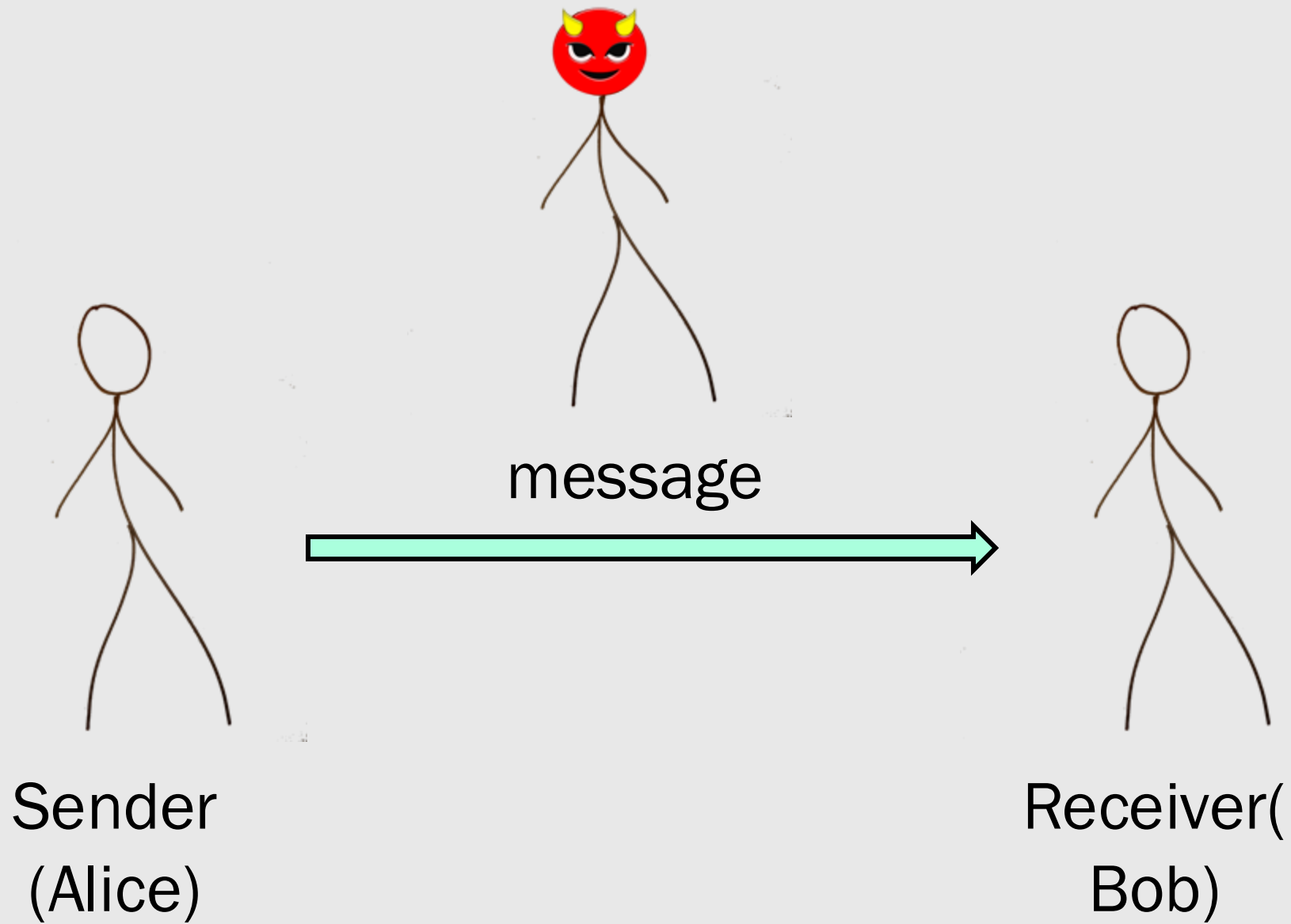
- Artichoke
- Bacon
- Barbecue chicken
- Basil
- Black olive
- Cheese
- Chicken
- Chili peppers
- Chorizo
- Garlic
- Giardiniera
- Green pepper
- Ham
- Jalapeno
- Mashed potato
- Mushroom



# INTRODUCING THE CRYPTOGRAPHIC TOOLBOX



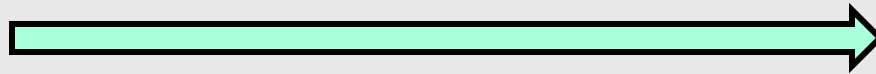






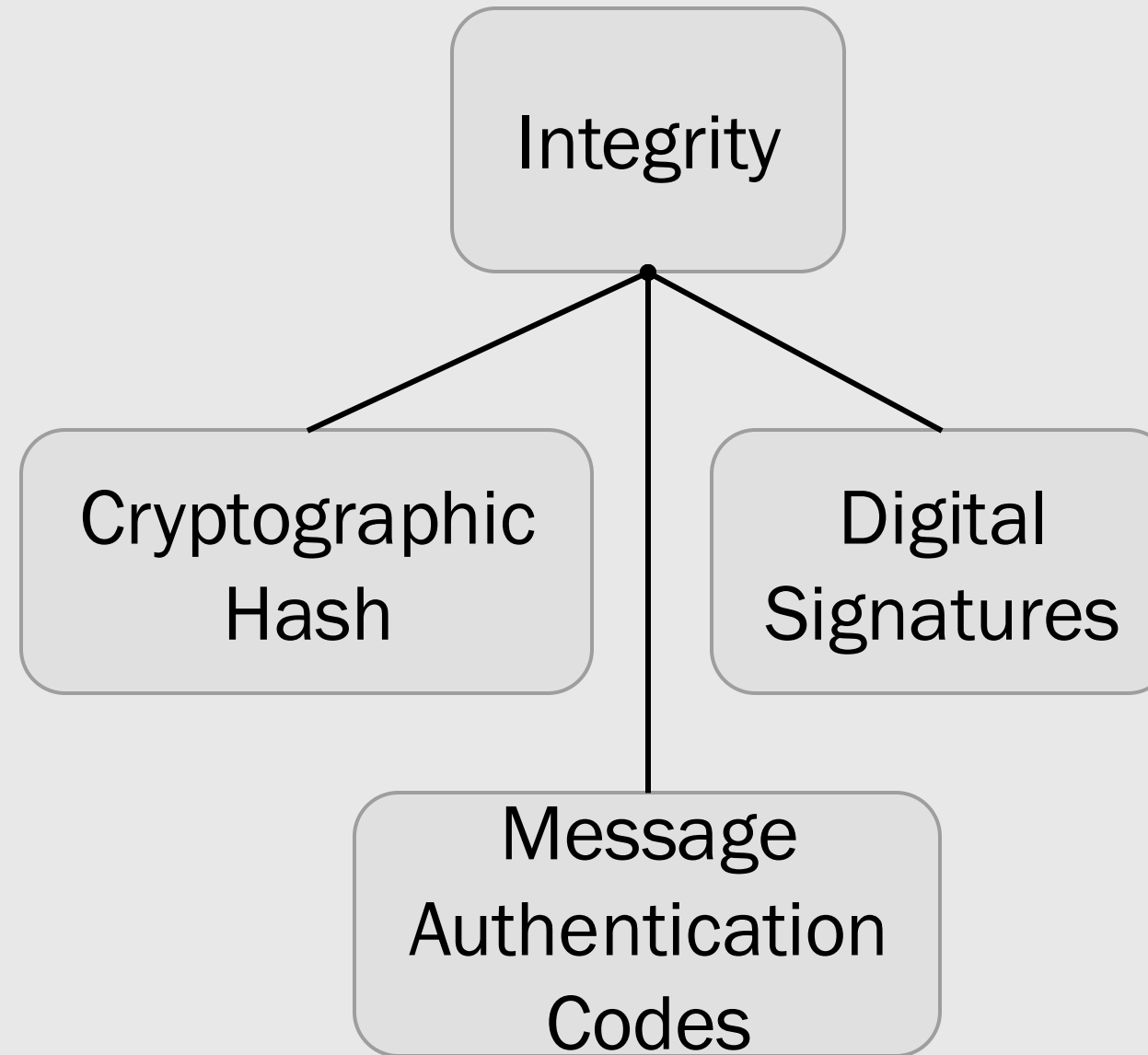
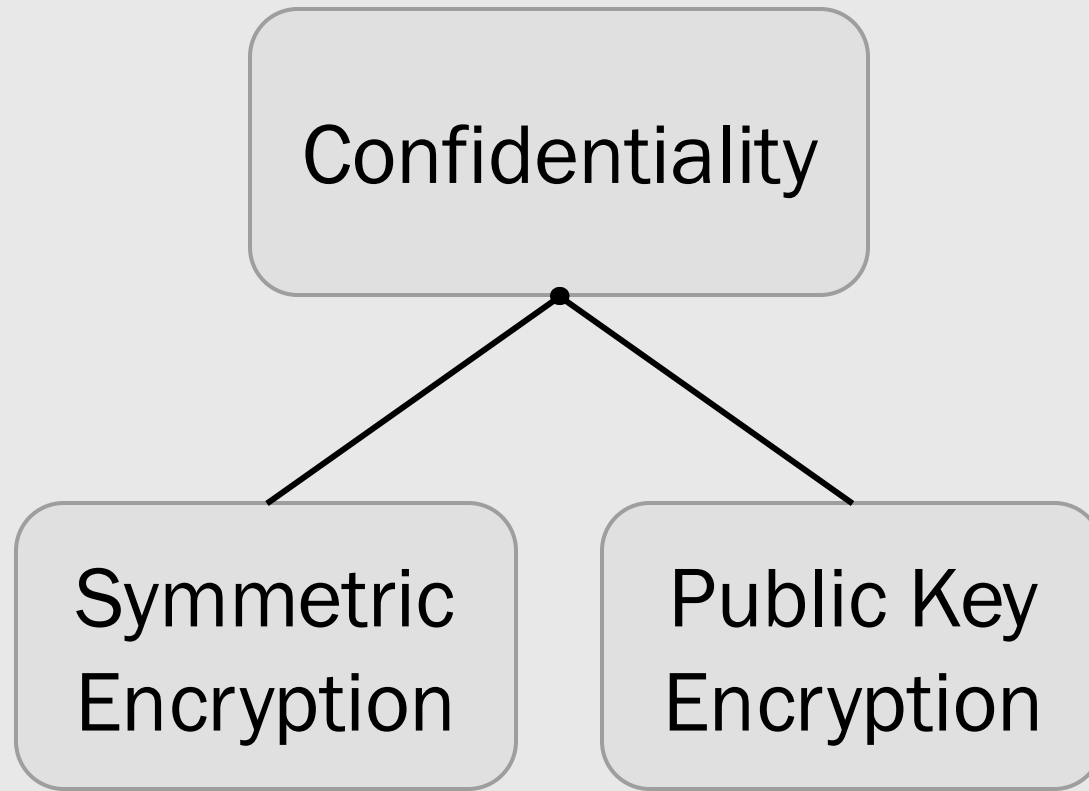
Sender  
(Alice)

message

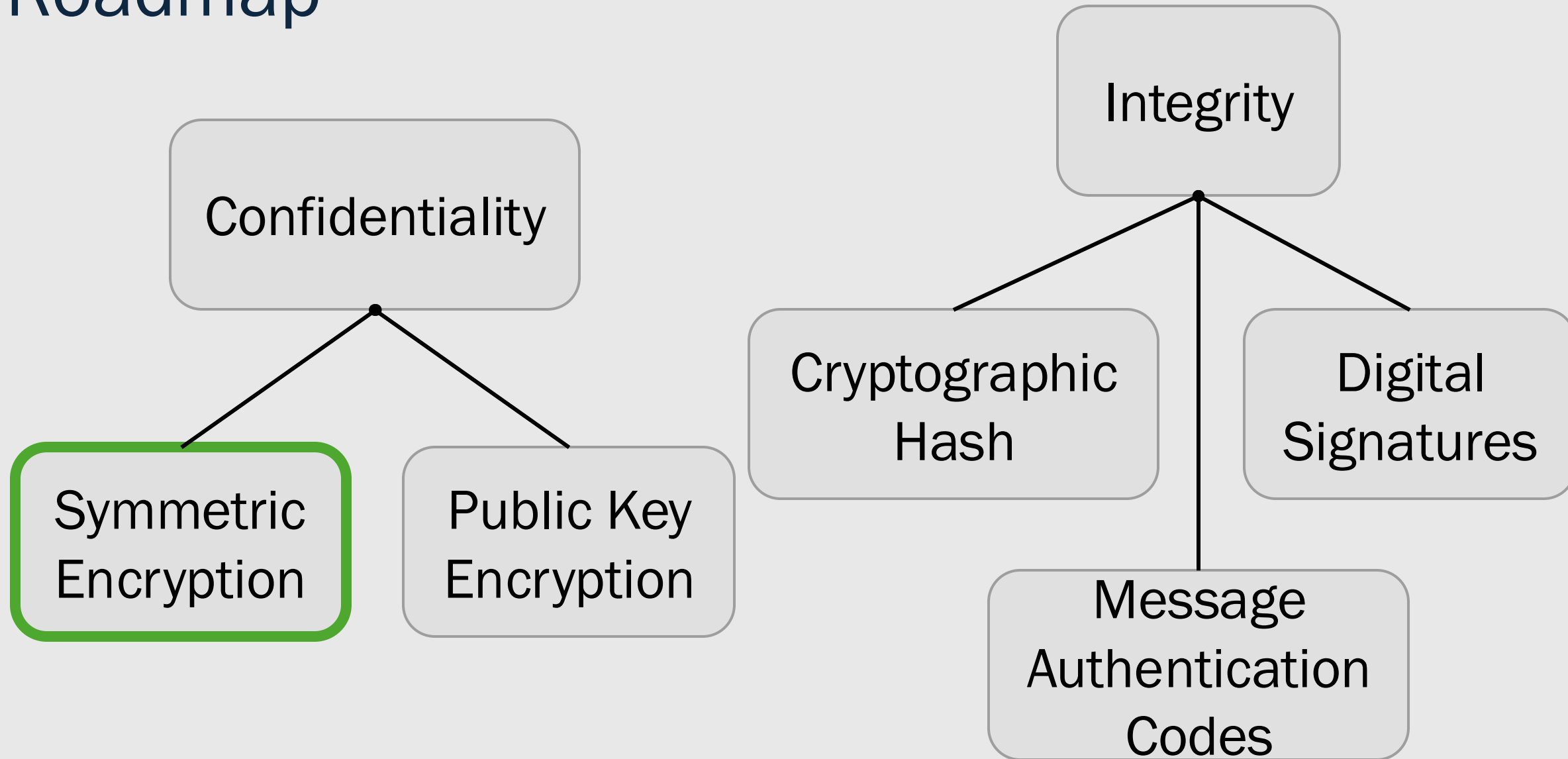


Receiver(  
Bob)

# Roadmap




# Roadmap




# Encryption Terminology

- Encrypt
- Decrypt
- Plaintext
- Ciphertext
- Cipher

# Encryption Terminology

- ***Encrypt***  Encode a message to obscure its meaning. (Also, encipher)
- Decrypt
- Plaintext
- Ciphertext
- Cipher

# Encryption Terminology

- Encrypt
- ***Decrypt***  Decode an encrypted message to reveal its original meaning. (Also, decipher)
- Plaintext
- Ciphertext
- Cipher

# Encryption Terminology

- Encrypt
- Decrypt
- ***Plaintext***  Original message
- Ciphertext
- Cipher



# Encryption Terminology

- Encrypt
- Decrypt
- Plaintext
- ***Ciphertext*** ← Encrypted message
- Cipher

# Encryption Terminology

- Encrypt
- Decrypt
- Plaintext
- Ciphertext
- ***Cipher***



Algorithm for encrypting or  
decrypting

# Encryption and Decryption

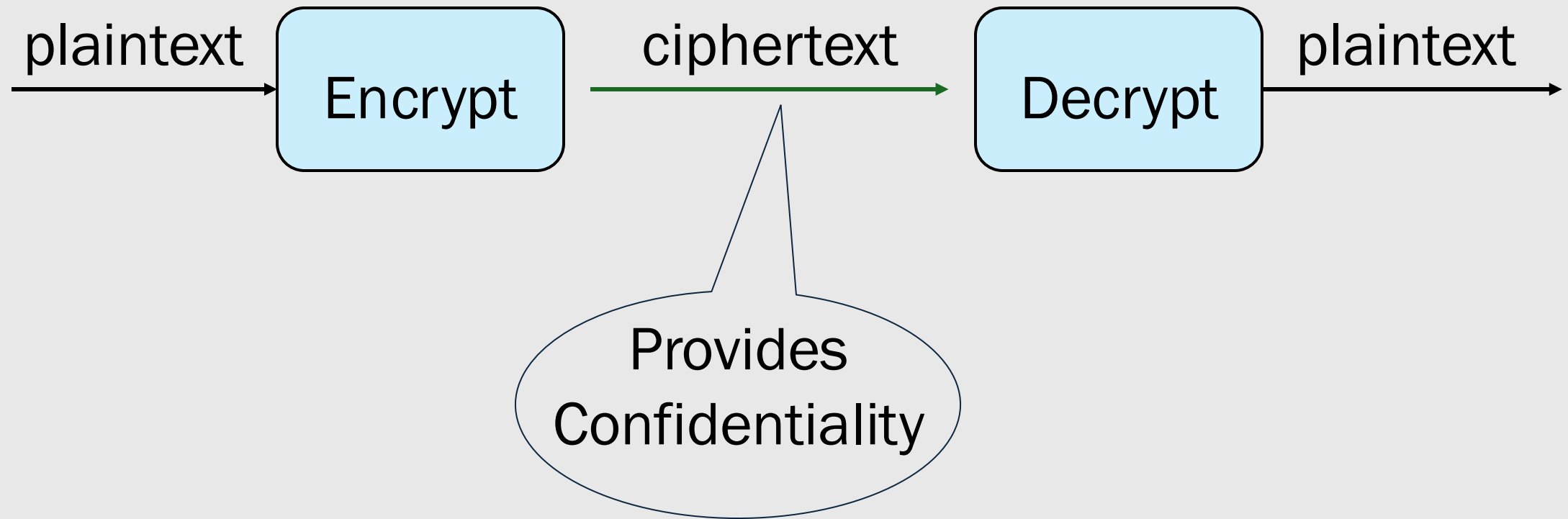


# Encryption and Decryption

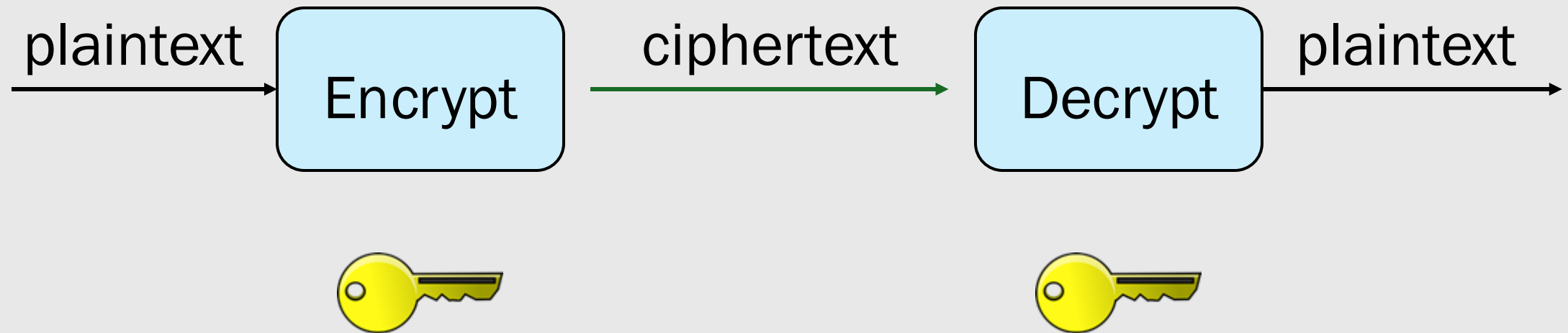


$$\text{msg} = \text{Dec}(\text{Enc}(\text{msg}))$$

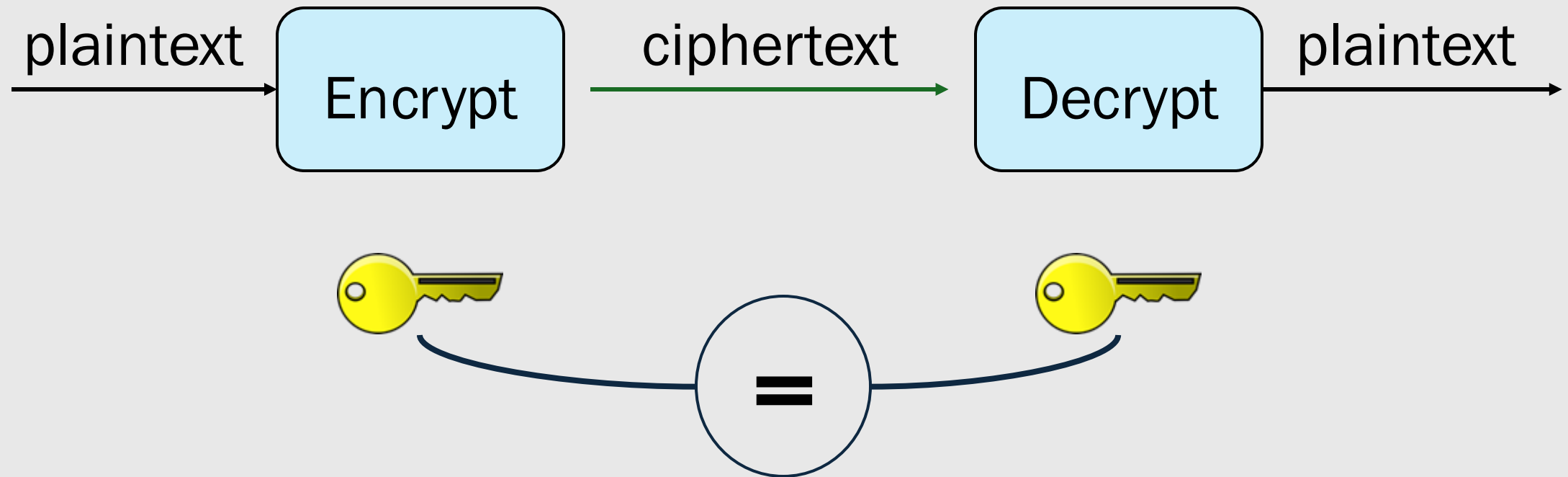
# Encryption and Decryption



# Encryption and Decryption

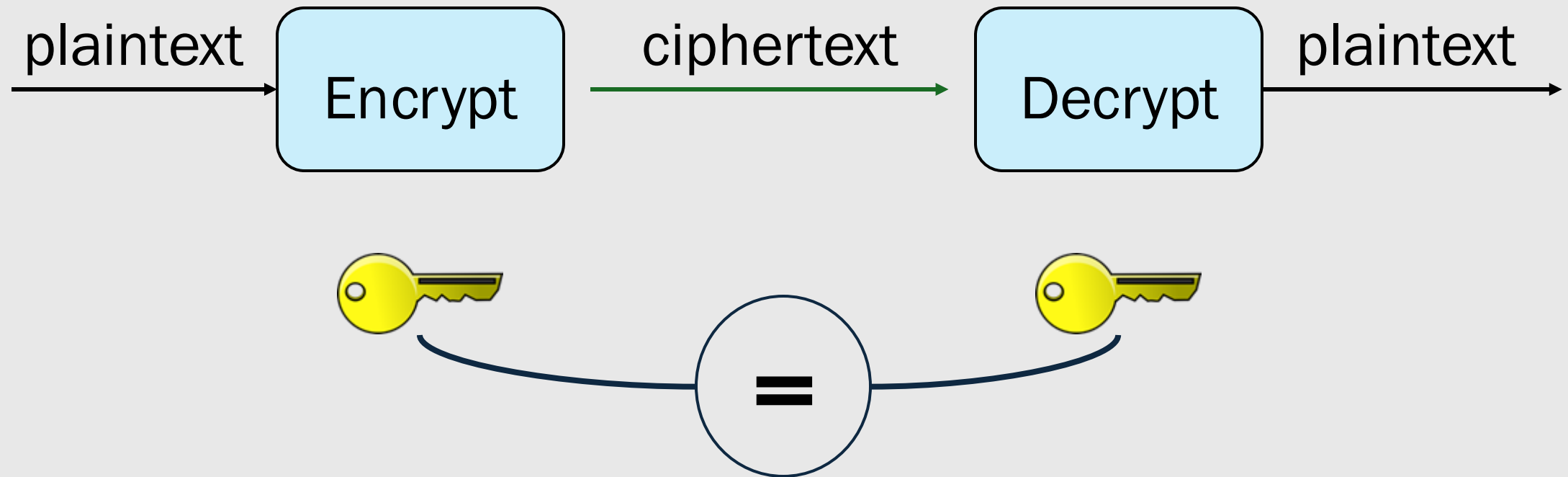


# Encryption and Decryption



# Encryption and Decryption

$$c = \text{Enc}_k(\text{msg})$$
$$\text{msg} = \text{Dec}_k(c)$$





# Cryptosystem

Def'n: A system for encryption and decryption

- Encryption algorithm
- Decryption algorithm
- Key generation
- Key management

The security of a cryptosystem should depend only on the secrecy of the keys

*Kerckhoffs' Principle*

# Caesar Cipher

a b c d e f g ... w x y z



d e f g h i j ... z a b c

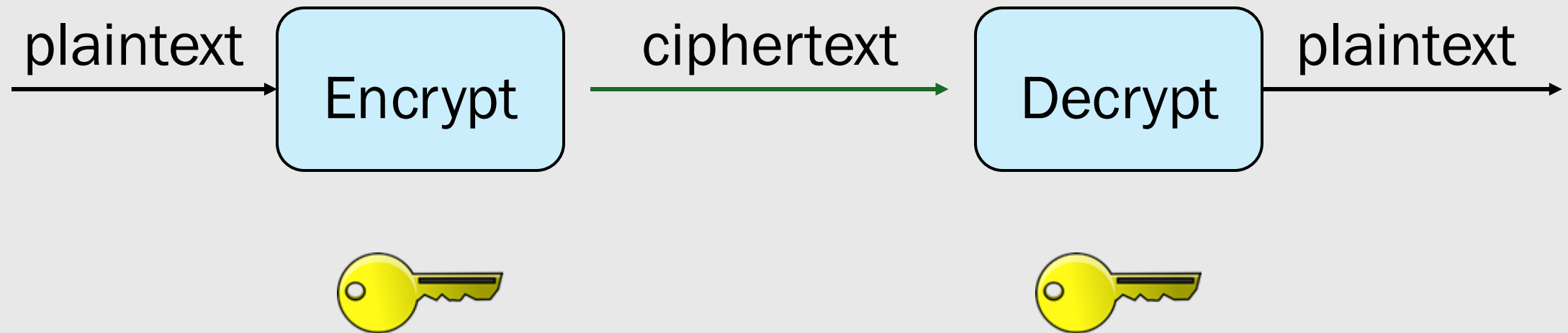
key = 3



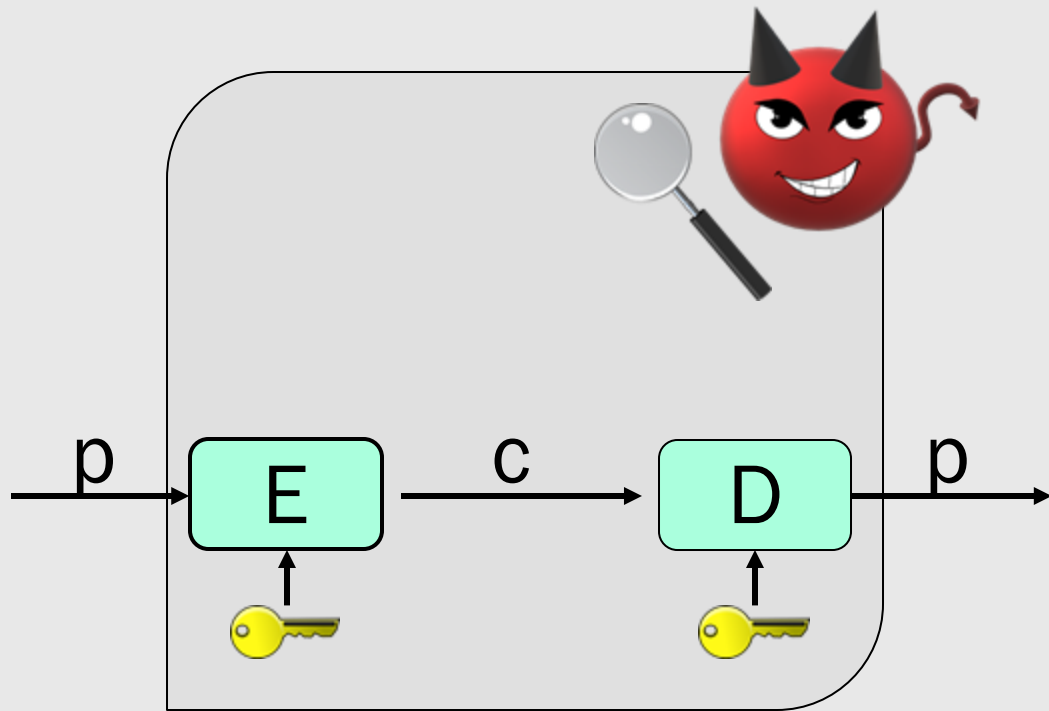
BREAKING ENCRYPTION

# Encryption and Decryption

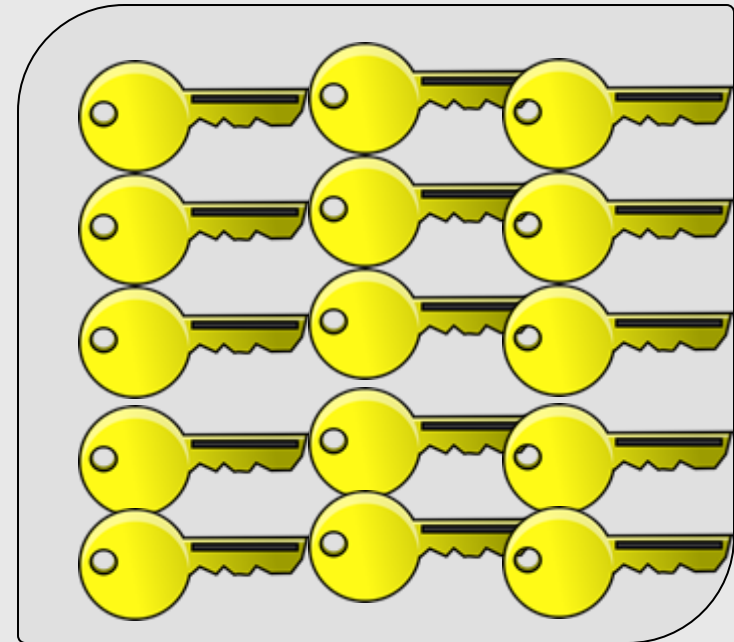
$$c = \text{Enc}_k(\text{msg})$$
$$\text{msg} = \text{Dec}_k(c)$$



# Breaking Encryption

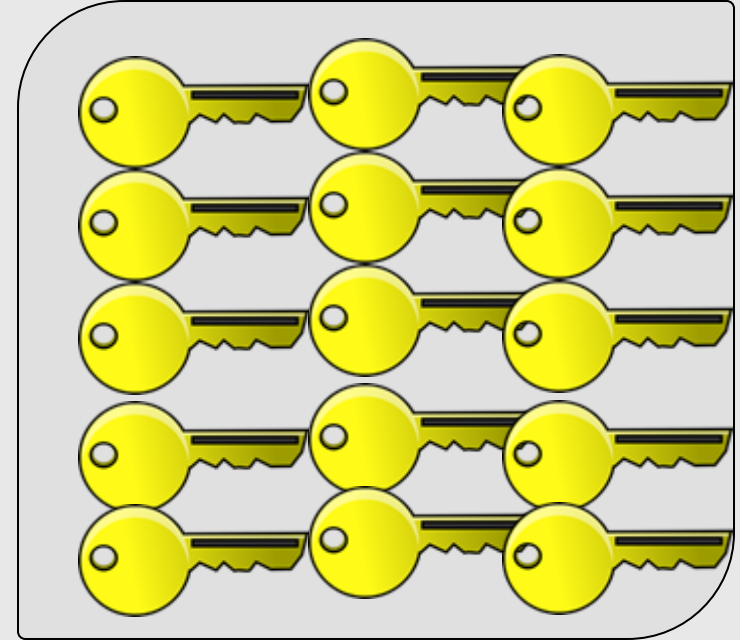


Cryptanalysis



Exhaustive  
Search

# Exhaustive Search



# Exhaustive Search

Def'n: Try every possible key until the correct one is found.



# Shift Cipher (Caesar Cipher with key)

a b c d e f g ... w x y z



d e f g h i j ... z a b c

$$\text{Enc}_k(m) \triangleq (m + k) \bmod 26$$

# Shift Cipher (Caesar Cipher with key)

Number of keys to try: 25

E.g.,

Klhy AHZ,

Aol huzdly rlf pz pu Zpaalyzvu Ohss.

Zpujlylsf,

Wyvm. Yfu

# Exhaustive Search

E.g., DES

- 56-bit key
- $10^{10}$  –  $10^{15}$  encryptions per second

# Exhaustive Search

E.g., DES


- 56-bit key
- $10^{10}$  –  $10^{15}$  encryptions per second
- Search time: [75 days, ~1 minute]



much too short for  
today's computers!

# Exhaustive Search

E.g., DES

- 56-bit key
  - $10^{10}$  –  $10^{15}$  encryptions per second
  - Search time: [75 days, ~1 minute]
- much too short for today's computers!
- 

On average, the attacker can expect to be successful in the time it takes to search only  $2^{55}$  keys – half the key space!

# Computational Security

Cost to break encryption  $\gg$  value of asset

Time to break encryption  $\gg$  lifetime of asset

# Computational Security

Def'n: Secure against an attacker with fixed computational resources

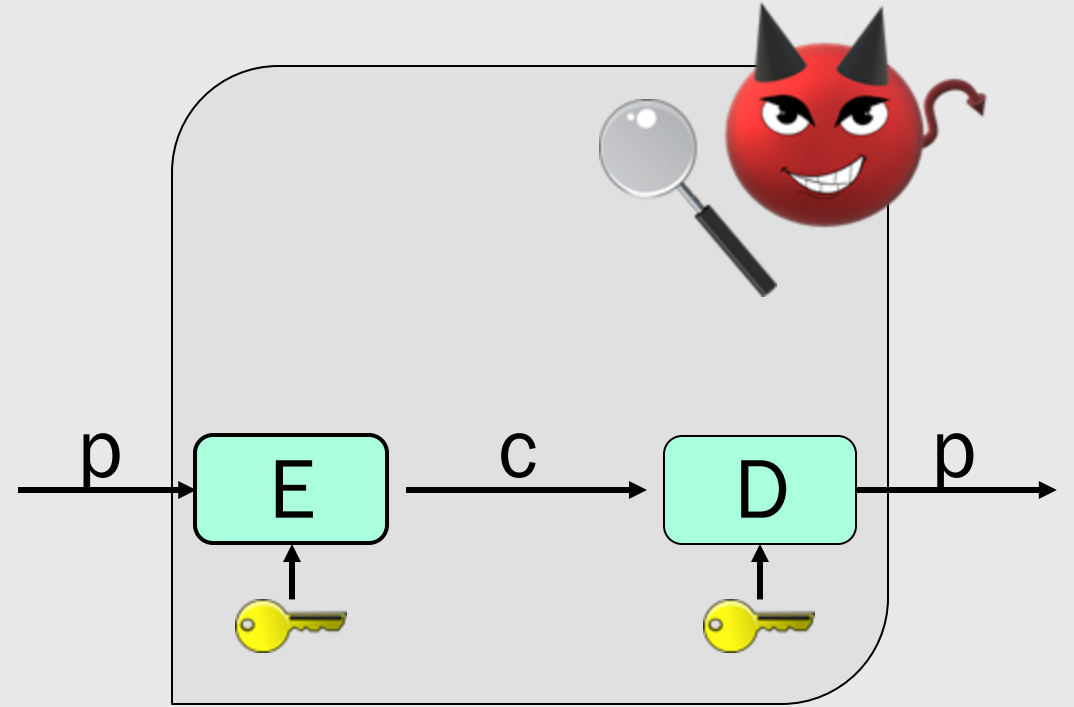
E.g., An encryption algorithm using 128-bit keys is computationally secure against exhaustive search.

Any secure encryption scheme must have a key space that is not vulnerable to exhaustive search

--Sufficient Key Space Principle



# Cryptanalysis



# Cryptanalysis



Def'n: recover key material or plaintext by exploiting flaws in the cryptosystem

# Mono-Alphabetic Substitution

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



x e u a d n b k v m r o c q f s y h w g l z i j p t

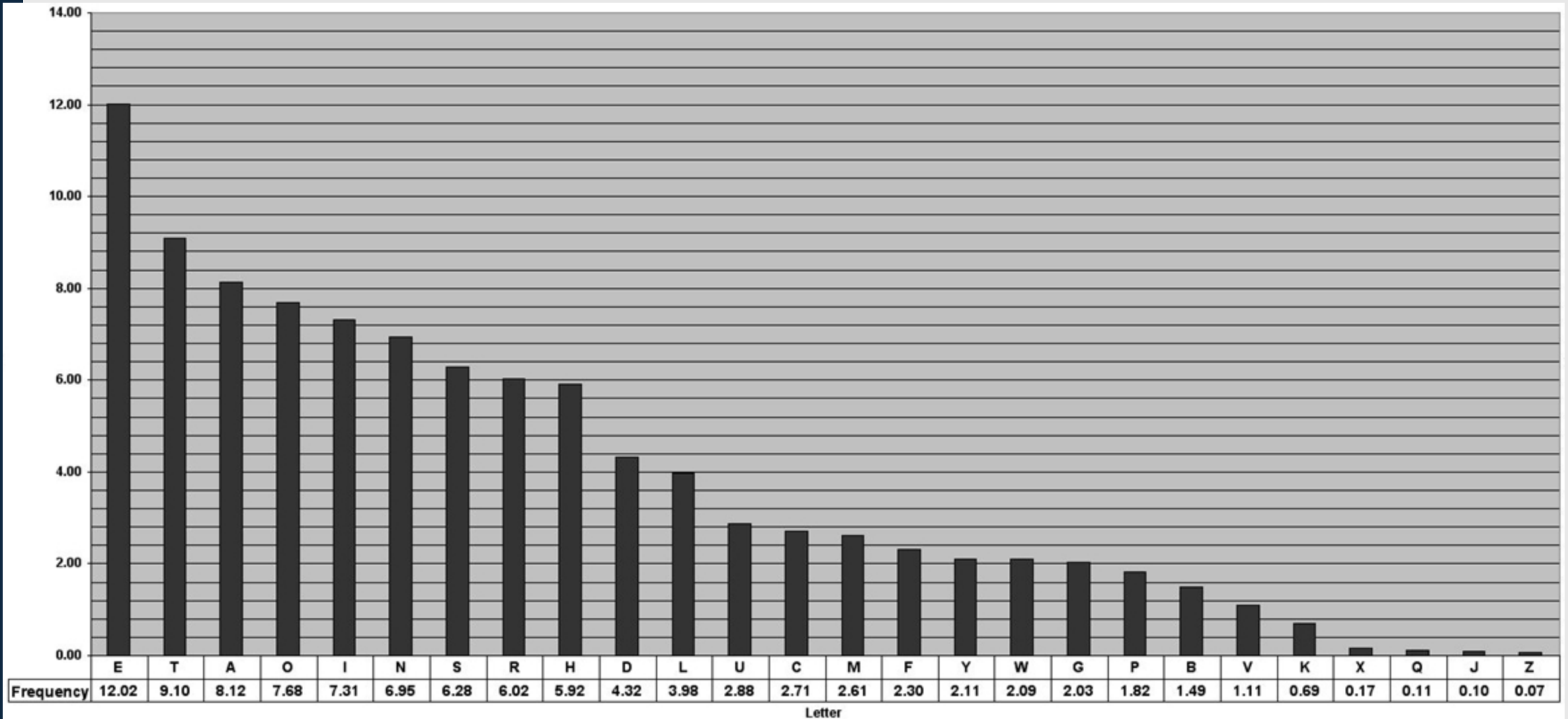
# Mono-Alphabetic Substitution

Number of keys to try:  $26! \approx 2^{88}$

E.g.,

GDOOGKDCXEFLGCD

# Letter Frequencies in English



# Poly-Alphabetic Shift (Vigenère Cipher)

Plaintext:    tell them about me

Key:            cafe cafe cafec af

Ciphertext:    veqp vhjq cbtyv mj

$$\text{Enc}_k(m_i) \triangleq (m_i + k_i) \bmod 26$$

# Cryptanalysis



- Cipher-text only attack
- Known-plaintext attack
- Chosen-plaintext attack (CPA)
- Chosen-ciphertext attack (CCA)

# Cryptanalysis



- ***Cipher-text only attack*** ←
- Known-plaintext attack
- Chosen-plaintext attack (CPA)
- Chosen-ciphertext attack (CCA)

adversary observes  
ciphertexts



# Cipher-text Only Attack



Given:

Enc, Dec,  $M$ , and a ciphertext (ct)

Goal:

Recover a msg such that  
 $m = \text{Dec}_{\text{key}}(\text{ct})$

Set-up:

Enc, Dec,  $M = \{0, 1\}^n$

$K = \{0, 1\}^p$

$\text{msg} \in M$ ,  $\text{key} \in K$

$\text{ct} = \text{Enc}_{\text{key}}(\text{msg})$

# Cipher-text Only Attack



Given:

Enc, Dec, M, and a ciphertext (ct)

Goal:

Recover a msg such that  
 $m = \text{Dec}_{\text{key}}(\text{ct})$

Set-up:

Enc, Dec,  $M = \{0, 1\}^n$

$K = \{0, 1\}^p$

$\text{msg} \in M$ ,  $\text{key} \in K$

$\text{ct} = \text{Enc}_{\text{key}}(\text{msg})$

Enc = Mono-alphabetic substitution

$M = \{\text{"banana"}, \text{"orange"}\}$

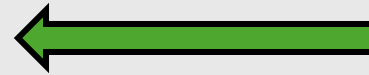
$\text{ct} = \text{xmqmqm}$



Attacker can figure out  $m = \text{"banana"}$

# Cryptanalysis

- Cipher-text only attack
- ***Known-plaintext attack***
- Chosen-plaintext attack (CPA)
- Chosen-ciphertext attack (CCA)



adversary learns  
plaintext-ciphertext  
pairs



# Known-Plaintext Attack



Given:

Enc, Dec,  $M$ ,

$(m_1, c_1), (m_2, c_2) \dots (m_j, c_j)$

and a ciphertext  $(ct_i)$

Goal:

Recover the msg such that

$$m_i = \text{Dec}_{\text{key}}(ct_i)$$

Set-up:

Enc, Dec,  $M = \{0, 1\}^n$

$K = \{0, 1\}^p$

$m_1, m_2 \dots m_j \in M, \text{key} \in K$

$ct_1 = \text{Enc}_{\text{key}}(m_1)$

$ct_2 = \text{Enc}_{\text{key}}(m_2) \dots$

$ct_j = \text{Enc}_{\text{key}}(m_j)$

# Known-Plaintext Attack



Given:

Enc, Dec,  $M$ ,  
 $(m_1, c_1), (m_2, c_2) \dots (m_j, c_j)$   
and a ciphertext  $(ct_i)$

Goal:

Recover the msg such that  
 $m_i = \text{Dec}_{\text{key}}(ct_i)$

Set-up:

Enc, Dec,  $M = \{0, 1\}^n$

$K = \{0, 1\}^p$

$m_1, m_2 \dots m_j \in M, \text{key} \in K$

$ct_1 = \text{Enc}_{\text{key}}(m_1)$

$ct_2 = \text{Enc}_{\text{key}}(m_2) \dots$

$ct_j = \text{Enc}_{\text{key}}(m_j)$

Enc = Shift Cipher

Given  $(m_1, c_1) = ("abc", "cfg")$



Attacker can recover the key and  
decrypt any future message!

# Cryptanalysis



- Cipher-text only attack
- Known-plaintext attack
- ***Chosen-plaintext attack (CPA)***
- Chosen-ciphertext attack (CCA)



Adversary can  
obtain ciphertext  
for plaintext of its  
choosing

# Chosen-Plaintext Attack



Given:

Enc, Dec,  $M$ , **ct** and

$$\mathbf{ct}_1 = \mathbf{Enc}_{\mathbf{key}}(m_1)$$

$$\mathbf{ct}_2 = \mathbf{Enc}_{\mathbf{key}}(m_2) \dots$$

$$\mathbf{ct}_j = \mathbf{Enc}_{\mathbf{key}}(m_j)$$

Goal:

Recover the msg such that

$$\mathbf{msg} = \mathbf{Dec}_{\mathbf{key}}(\mathbf{ct})$$

Set-up:

Enc, Dec,  $M = \{0, 1\}^n$

$K = \{0, 1\}^p$

$m \in M$ ,  $\mathbf{key} \in K$

$$\mathbf{ct} = \mathbf{Enc}_{\mathbf{key}}(m)$$

# Chosen-Plaintext Attack



Given:

Enc, Dec,  $M$ , **ct** and

**ct**<sub>1</sub> = Enc<sub>key</sub>( $m_1$ )

**ct**<sub>2</sub> = Enc<sub>key</sub>( $m_2$ ) ...

**ct**<sub>j</sub> = Enc<sub>key</sub>( $m_j$ )

Goal:

Recover the msg such that

**msg** = Dec<sub>key</sub>(**ct**)

Set-up:

Enc, Dec,  $M = \{0, 1\}^n$

$K = \{0, 1\}^p$

$m \in M$ ,  $\text{key} \in K$

**ct** = Enc<sub>key</sub>( $m$ )

$K = \{a-z\}^3$

key: "key"

$M = \{ "a", "b", "c" \}$



CPA adversary can request Enc("a"),  
Enc("b"), Enc("c"), Enc("a") again,  
and learn the key



# Cryptanalysis



- Cipher-text only attack
- Known-plaintext attack
- Chosen-plaintext attack (CPA)
- ***Chosen-ciphertext attack (CCA)***



Adversary can  
obtain decryption  
for ciphertext of its  
choosing

# Chosen-Ciphertext Attack



Given:

Enc, Dec,  $M$ , **ct** and

**$m_1 = \text{Dec}_{\text{key}}(\text{ct}_1)$**

**$m_2 = \text{Enc}_{\text{key}}(\text{ct}_2) \dots$**

**$m_j = \text{Enc}_{\text{key}}(\text{ct}_j)$**

Goal:

Recover the msg such that

**$\text{msg} = \text{Dec}_{\text{key}}(\text{ct})$**

Set-up:

Enc, Dec,  $M = \{0, 1\}^n$

$K = \{0, 1\}^p$

$m \in M$ ,  $\text{key} \in K$

**$\text{ct} = \text{Enc}_{\text{key}}(m)$**

$K = \{0, 1\}^8$

Want to decrypt some given ct

→

Chooses:  $c_1 = \text{ct XOR } 0000\_0001$

Can query:  $m_1 = \text{Dec}_k(c_1)$  to try to  
recover  $m$  such that  $m = \text{Dec}_k(\text{ct})$

# Chosen-Ciphertext Attack



Given:

Enc, Dec,  $M$ , **ct** and

$m_1 = \text{Dec}_{\text{key}}(\text{ct}_1)$

$m_2 = \text{Enc}_{\text{key}}(\text{ct}_2) \dots$

$m_j = \text{Enc}_{\text{key}}(\text{ct}_j)$

Goal:

Recover the msg such that

**msg** =  $\text{Dec}_{\text{key}}(\text{ct})$

Set-up:

Enc, Dec,  $M = \{0, 1\}^n$

$K = \{0, 1\}^p$

$m \in M$ ,  $\text{key} \in K$

**ct** =  $\text{Enc}_{\text{key}}(m)$

$K = \{0, 1\}^8$

Want to decrypt some given ct

→

Chooses:  $c_1 = \text{ct XOR } 0000\_0001$

Can query:  $m_1 = \text{Dec}_k(c_1)$  to try to  
recover  $m$  such that  $m = \text{Dec}_k(\text{ct})$

# Cryptanalysis



- Cipher-text only attack
- Known-plaintext attack
- Chosen-plaintext attack (CPA)
- ***Chosen-ciphertext attack (CCA)***

} passive attacks

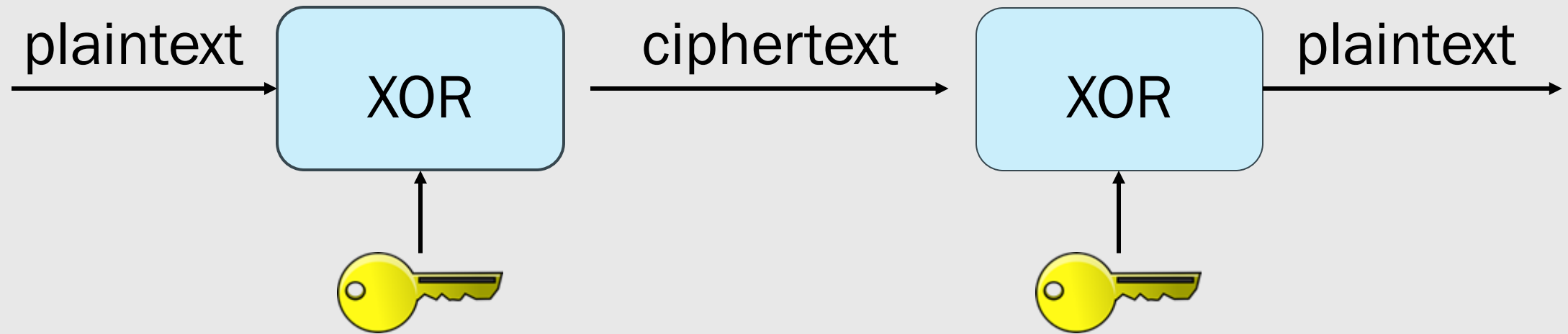
} active attacks

Klhy AHz,  
Aol huzdly rlf pz pu  
Zpaalyzvu Ohss.  
Zpujlylsf,  
Wyvm. Yfhu



ONE TIME PAD

# One Time Pad (Vernam Cipher)



$$c_i = p_i \oplus k_i$$

$$p_i = c_i \oplus k_i$$

# One Time Pad (Vernam Cipher): Encryption

plaintext		0	1	0	0	0	0	0	1
	$\oplus$								
key		0	1	0	1	0	1	0	1
		<hr/>							
ciphertext		0	0	0	1	0	1	0	0



# One Time Pad (Vernam Cipher): Decryption

ciphertext		0	0	0	1	0	1	0	0
	$\oplus$								
key		0	1	0	1	0	1	0	1
		<hr/>							
plaintext		0	1	0	0	0	0	0	1

# One Time Pad

- *Symmetric encryption algorithm*
- Stream cipher
- Substitution cipher



single key used for  
encryption and  
decryption

# One Time Pad

- Symmetric encryption algorithm
- Stream cipher
- ***Substitution cipher***



each unit of the  
plaintext is  
replaced with a  
unit of ciphertext

# One Time Pad Keys

- Key material is as long as message
- Key material is never reused
- Key material is kept secret
- Key material is chosen uniformly at random

uniformly at random



sampled from a uniform  
distribution

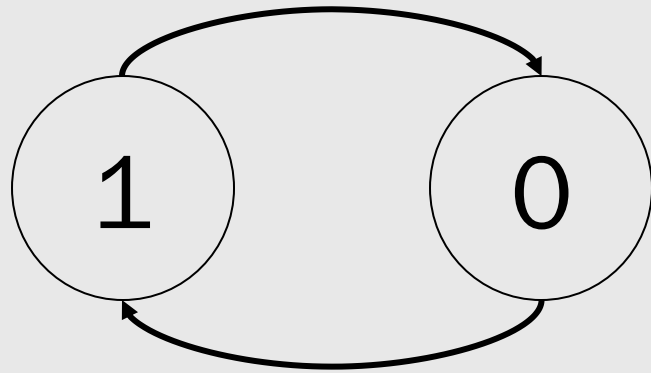
uniformly at random

```
graph TD; A[uniformly at random] --> B([without bias]); A --> C([nondeterministic]);
```

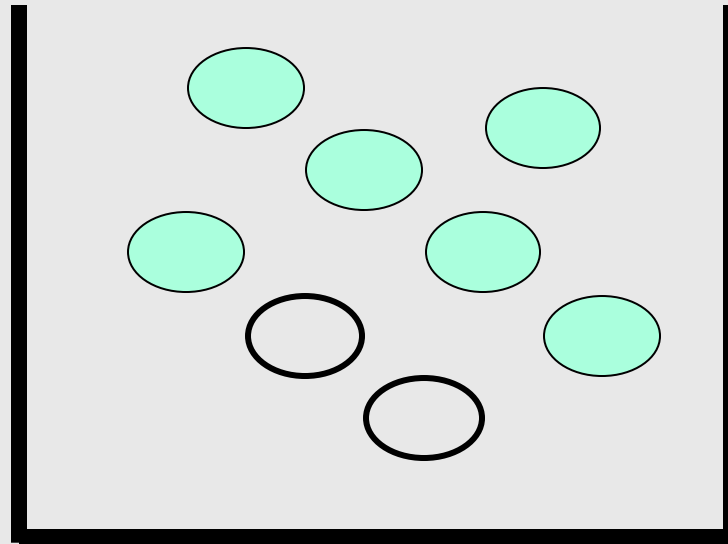
without bias

nondeterministic

# Unbiased, Deterministic



# Biased, Nondeterministic





# One Time Pad

- Key material is as long as message
- Key material is never reused
- Key material is kept secret
- Key material is chosen uniformly at random

OTP offers *information-theoretical security*

# Information-Theoretical Security

Def'n: Security derives from information theory

OTP is information-theoretically secure:

- Attacker cannot recover plaintext without the key
- Not susceptible to cryptanalysis
- Not susceptible to exhaustive search

# Worksheet Q2-5

# OTP

- Provides **perfect secrecy**
- Does not provide integrity
- Difficult to use in practice