

10/12/26

COMP 435:

Computer Security Concepts



Lecture 21: Intro to Crypto!



The University
of North Carolina
at Chapel Hill

<https://tinyurl.com/comp435-fa26>

Logistics Update



- Lab 2 Released! Due 10/14
- Details about final project posted
 - First milestone due 10/21

Plan for Class Today



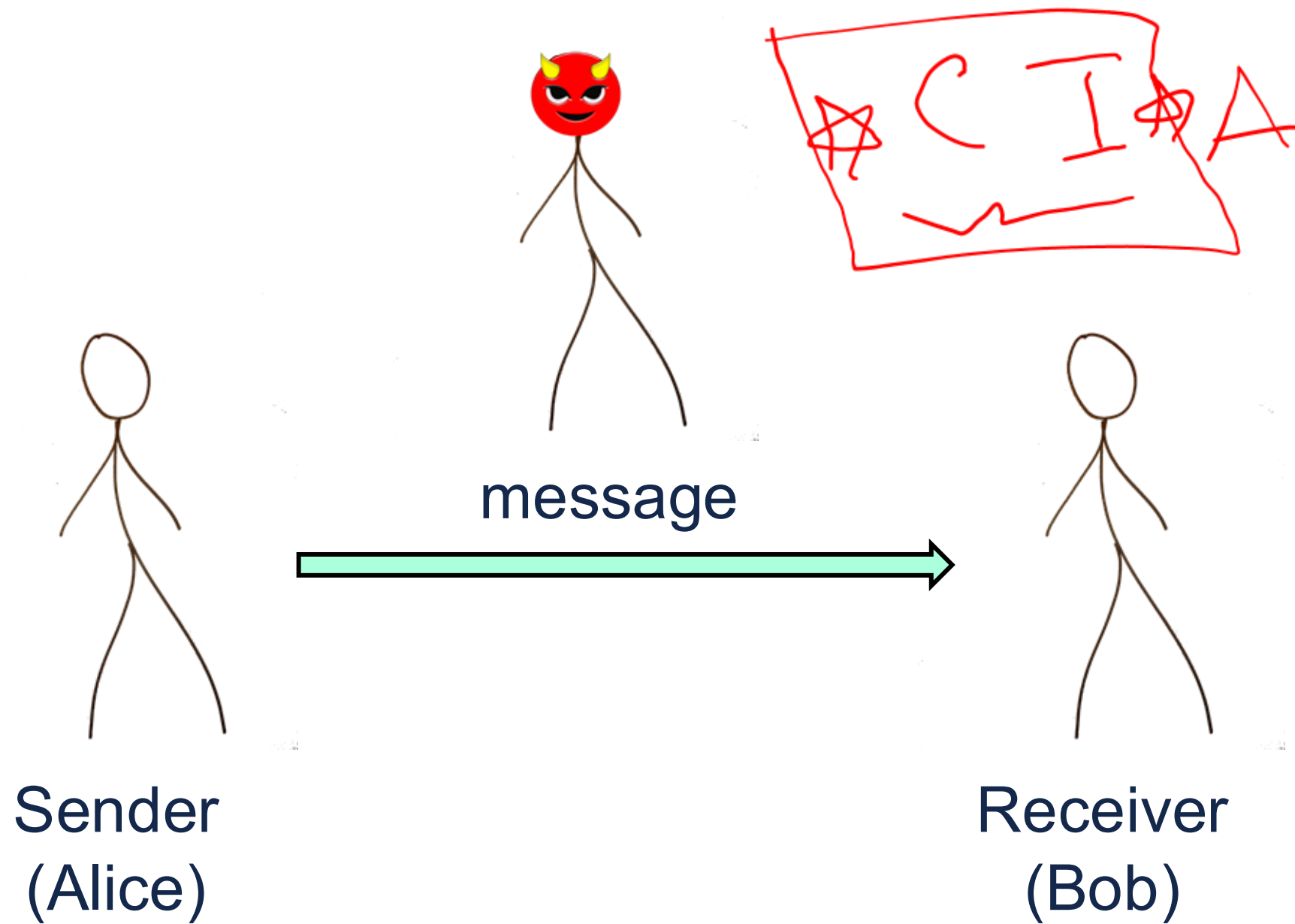
- 10:10-10:25 – Finish Authentication Concepts/Definitions
- 10:35-10:50 – Biometrics
- 10:50-11:00 – Maybe intro to crypto!

Guiding Question 😊

How do we
authenticate our
systems?



Introducing the Cryptographic Toolbox



✱ authentication 

✱ accountability



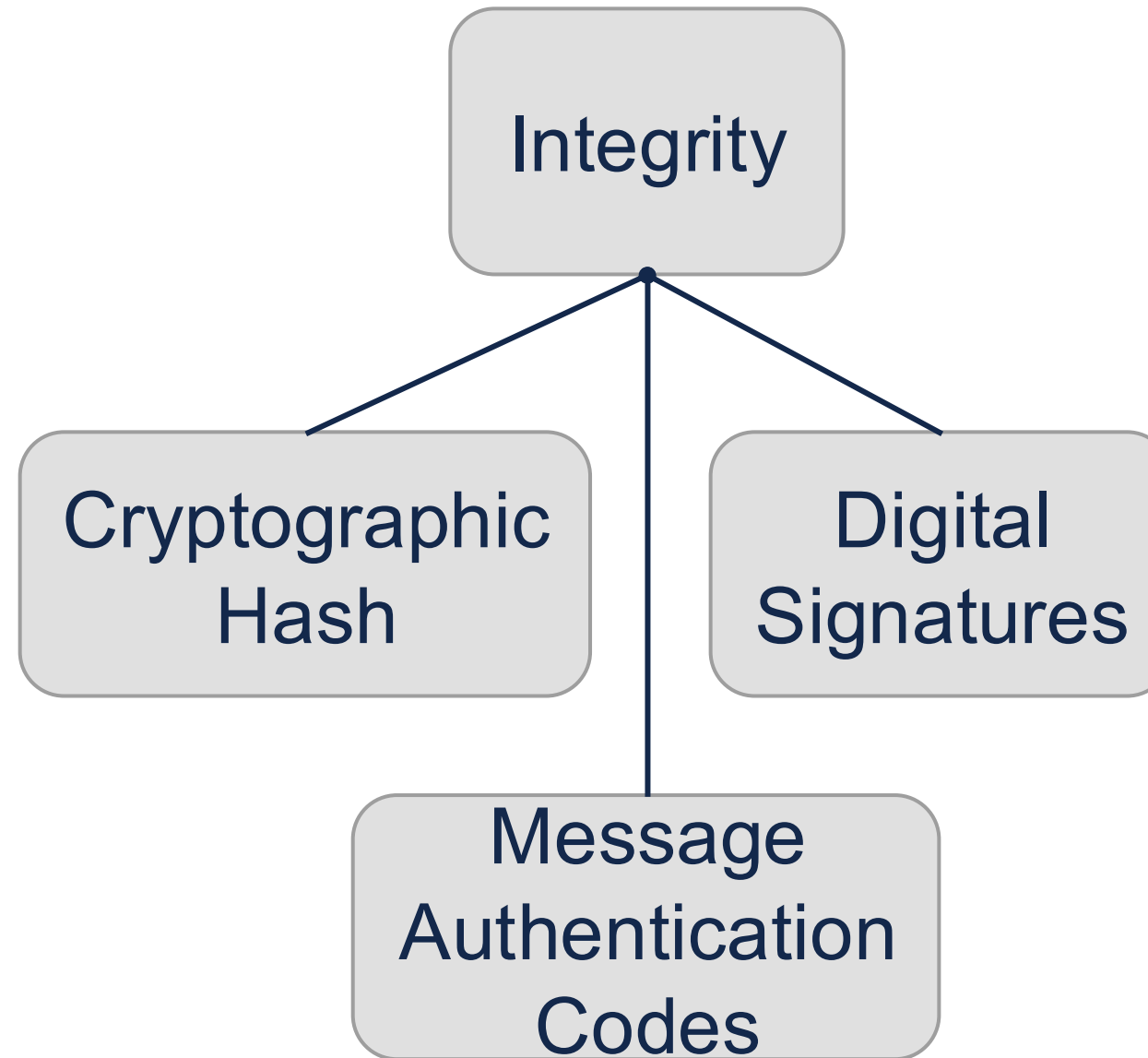
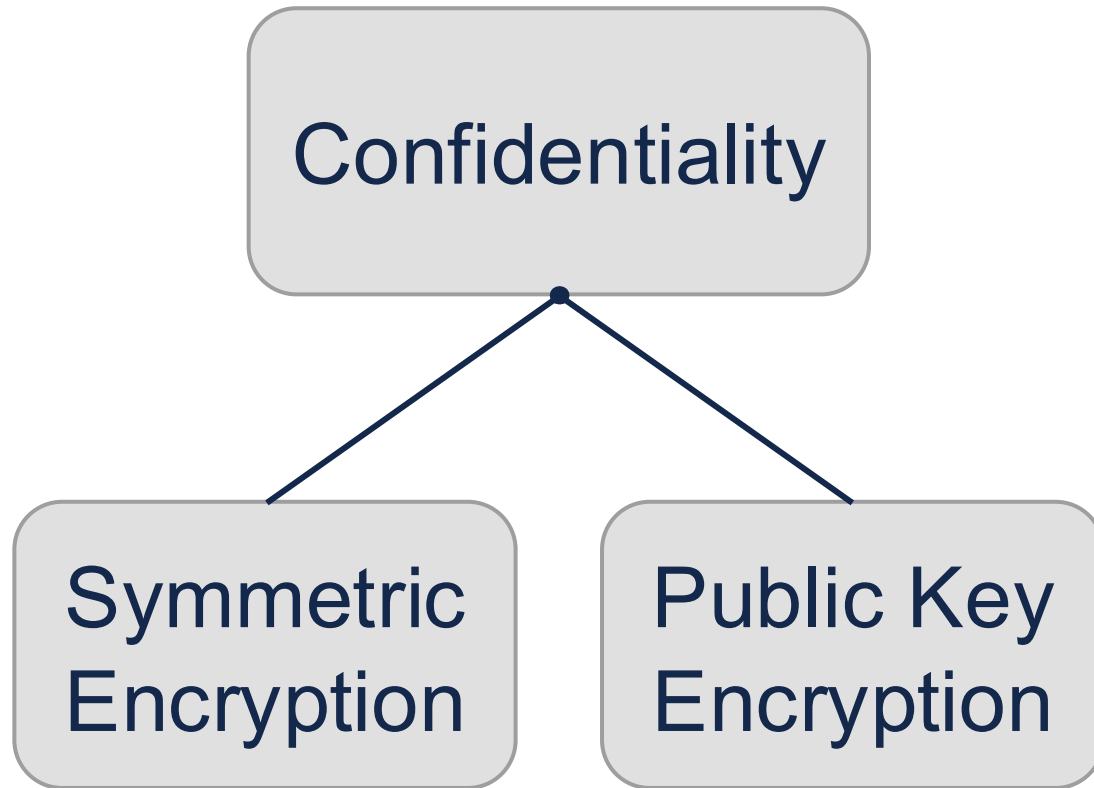
Sender
(Alice)

message

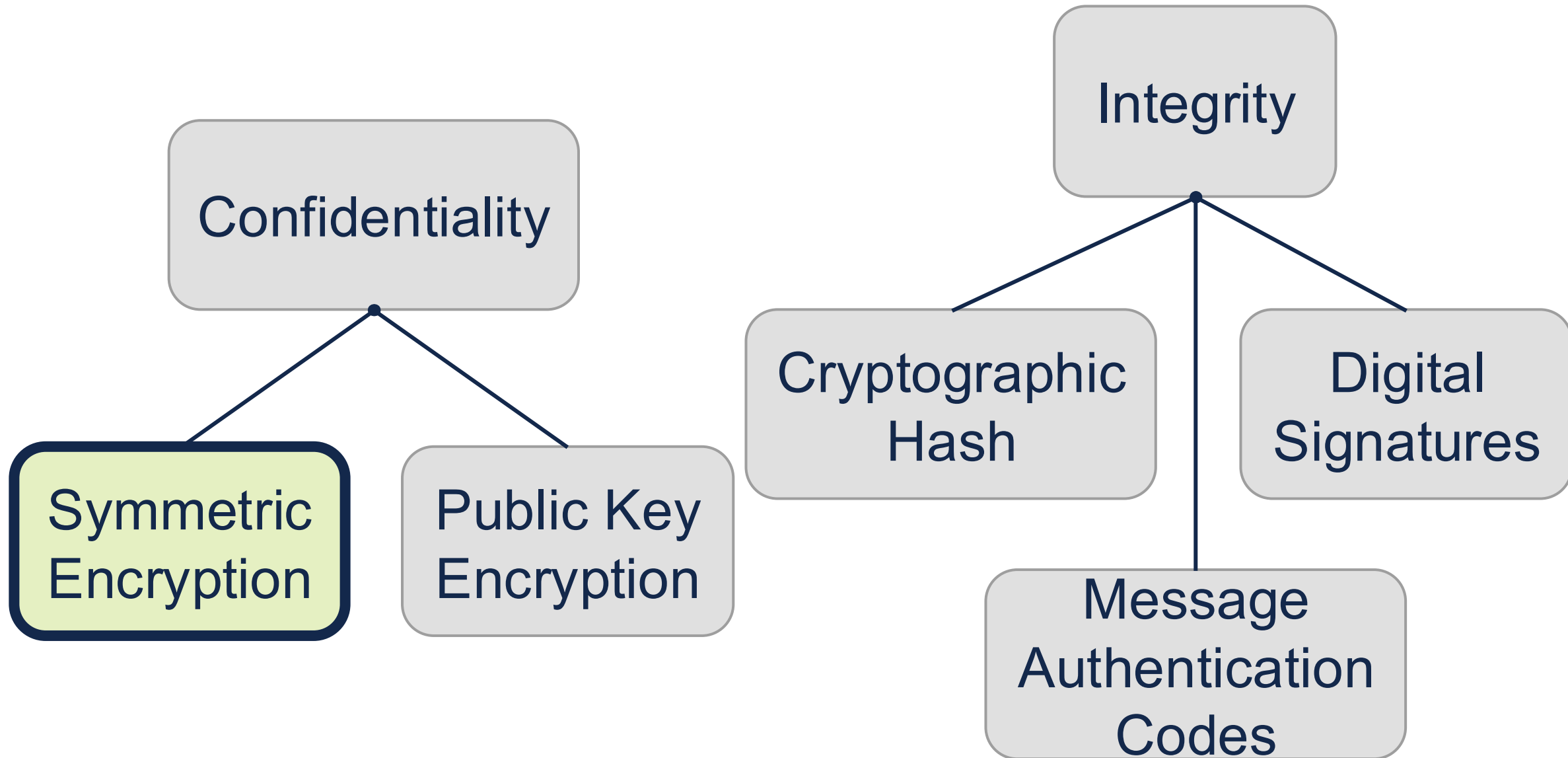


Receiver
(Bob)

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***
- Plaintext
- Ciphertext
- Cipher



Decode an encrypted message to reveal its original meaning. (Also, decipher)



Encryption Terminology

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



Original message

Encryption Terminology



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



Encrypted message

Encryption Terminology



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



Algorithm for encrypting or
decrypting

Encryption and Decryption

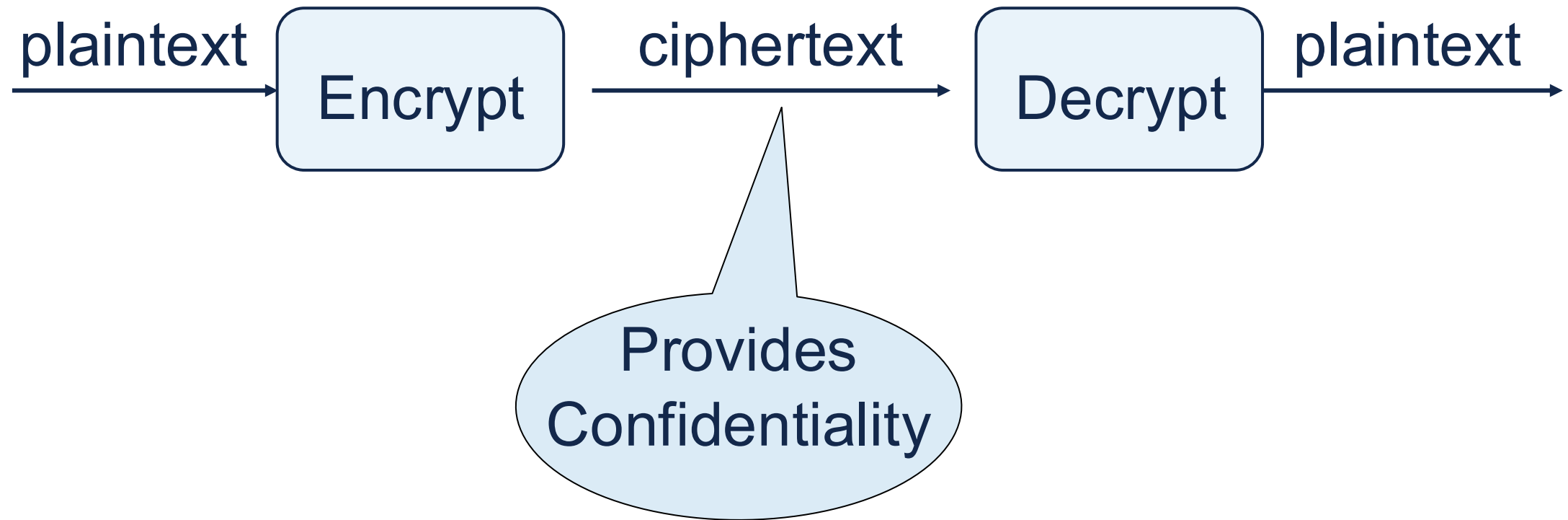


Encryption and Decryption

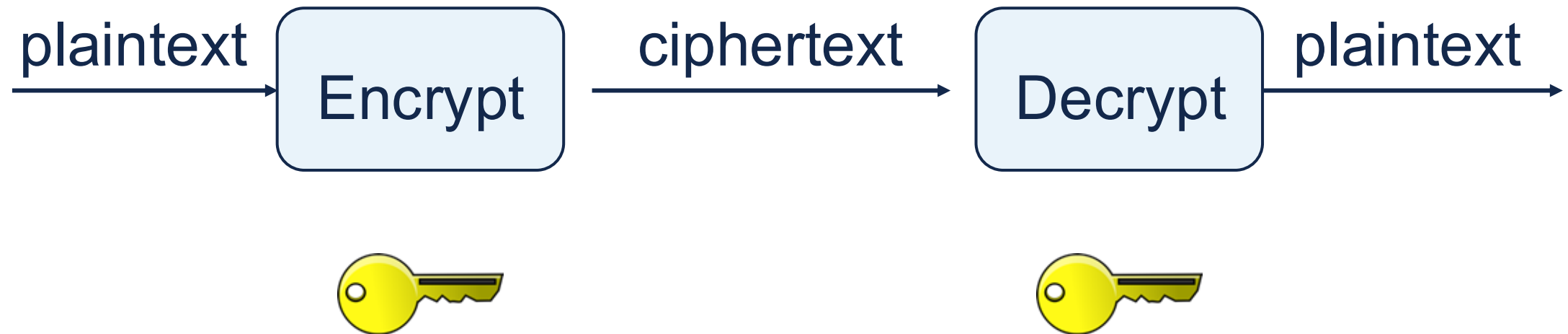


$$\text{msg} = \text{Dec}(\text{Enc}(\text{msg}))$$

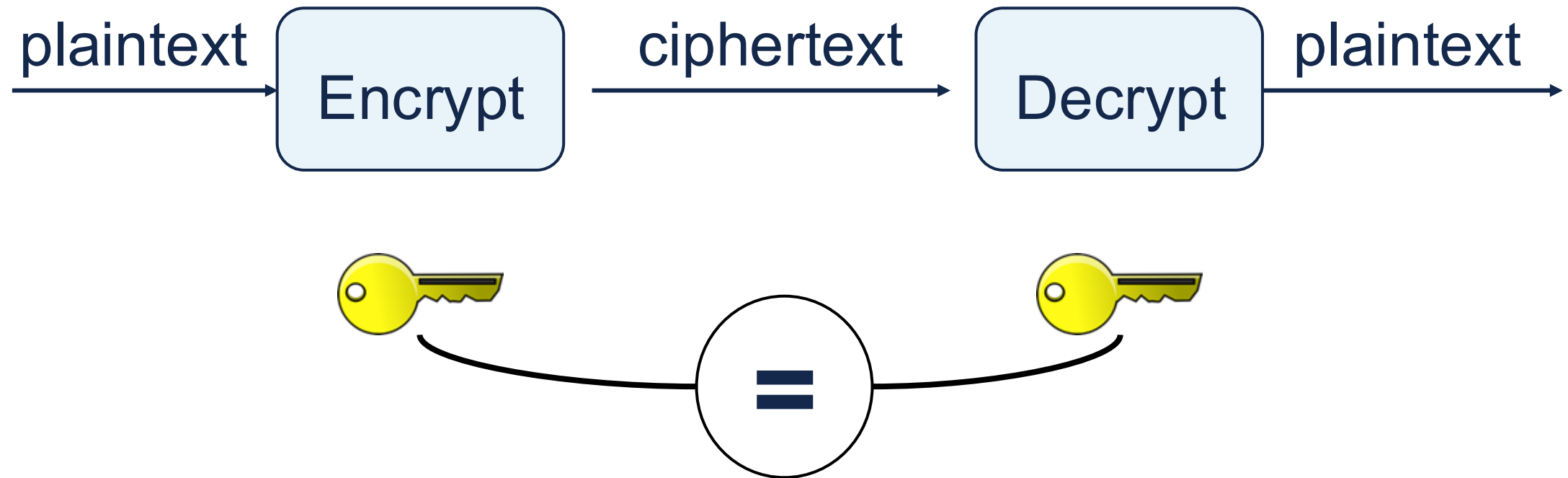
Encryption and Decryption



Encryption and Decryption



Encryption and Decryption

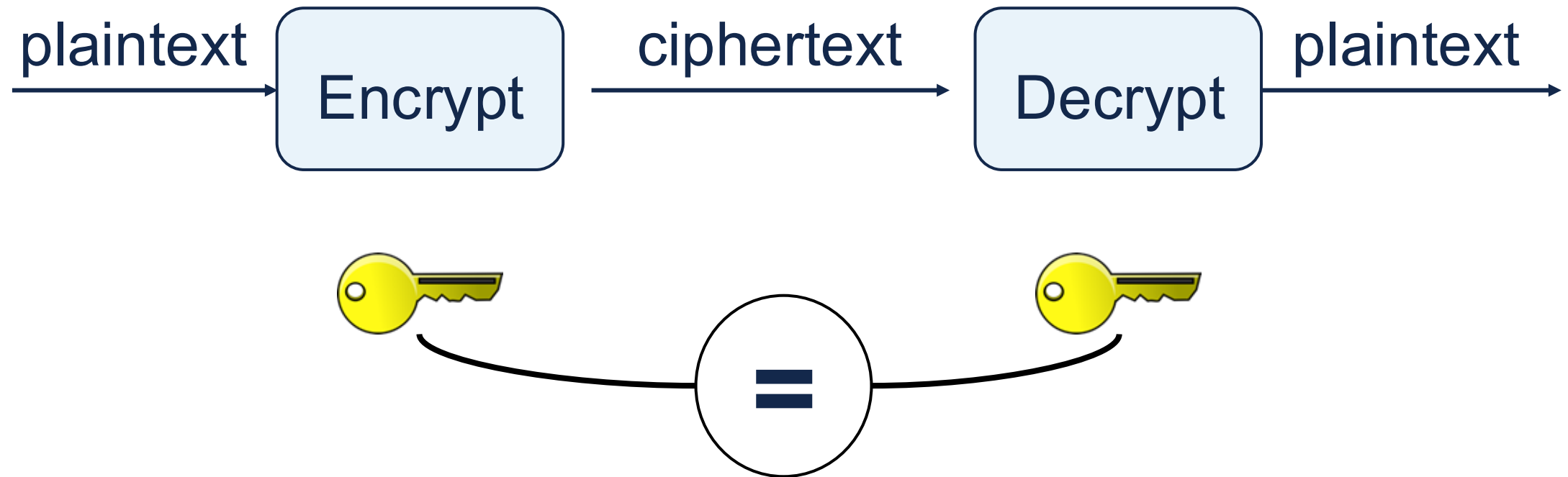


Encryption and Decryption



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

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



Cryptosystem



Def'n: A system for encryption and decryption

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



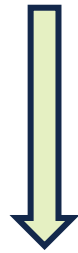
Kerckhoffs' Principle

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

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

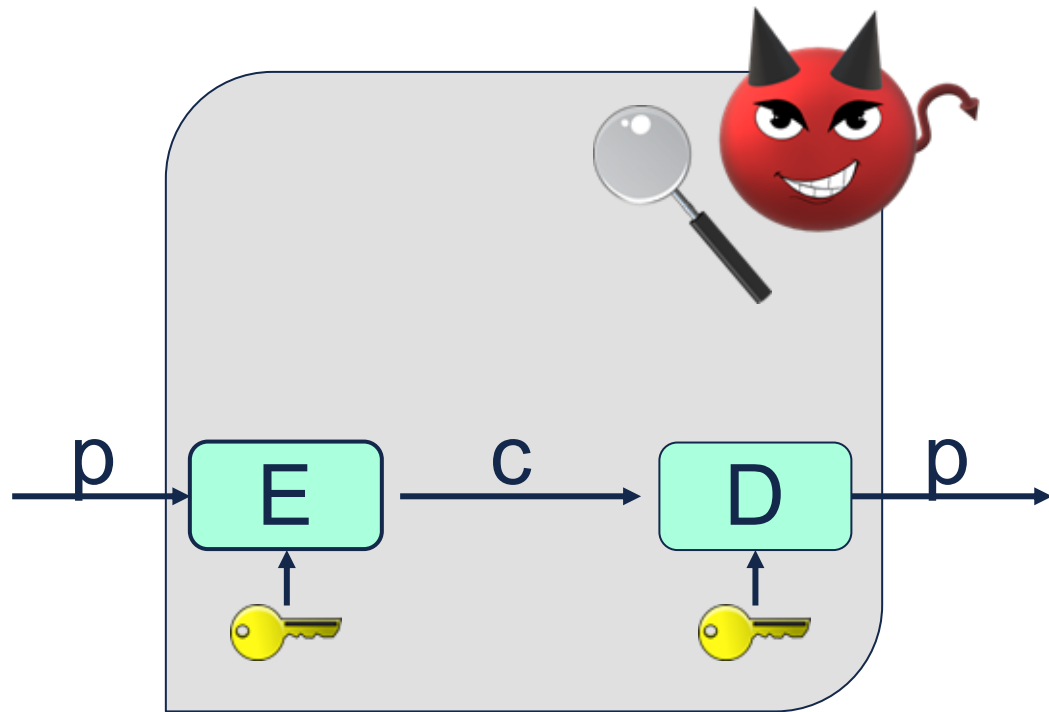


$$c = \text{Enc}_k(\text{msg})$$

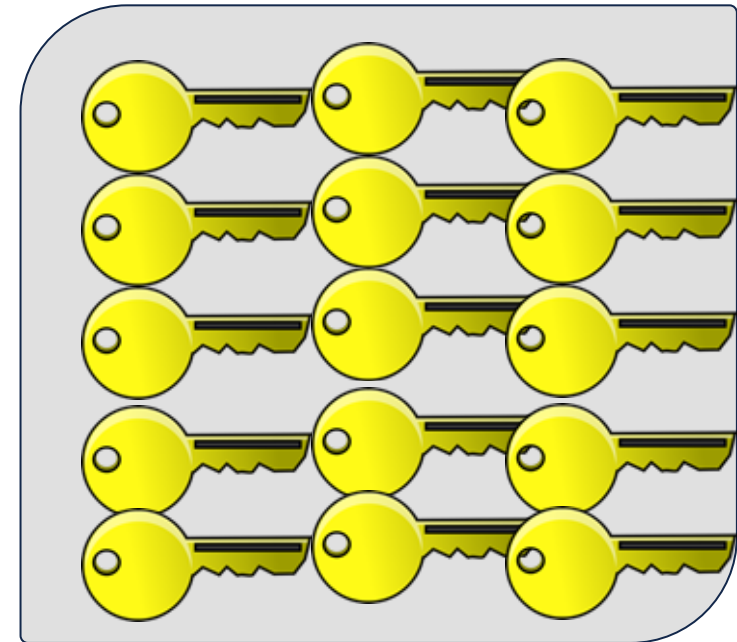
$$\text{msg} = \text{Dec}_k(c)$$



Breaking Encryption



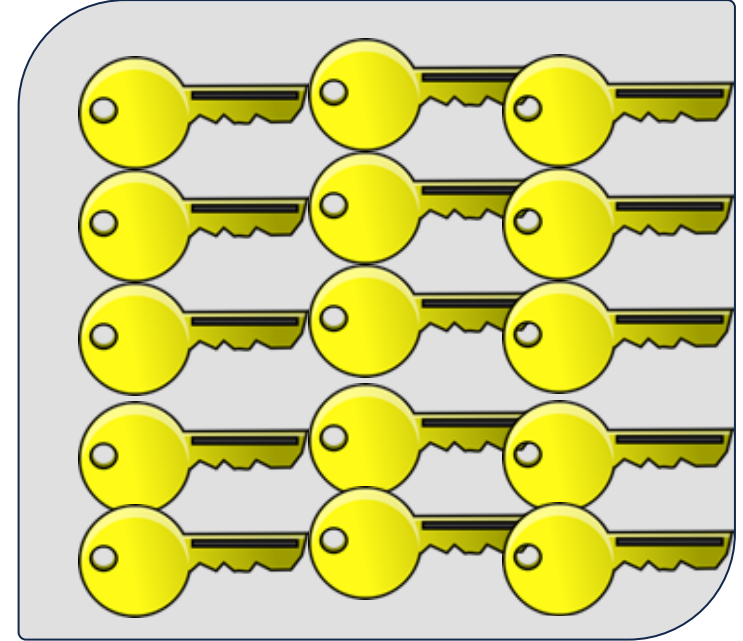
Cryptanalysis



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!

2^{56}

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

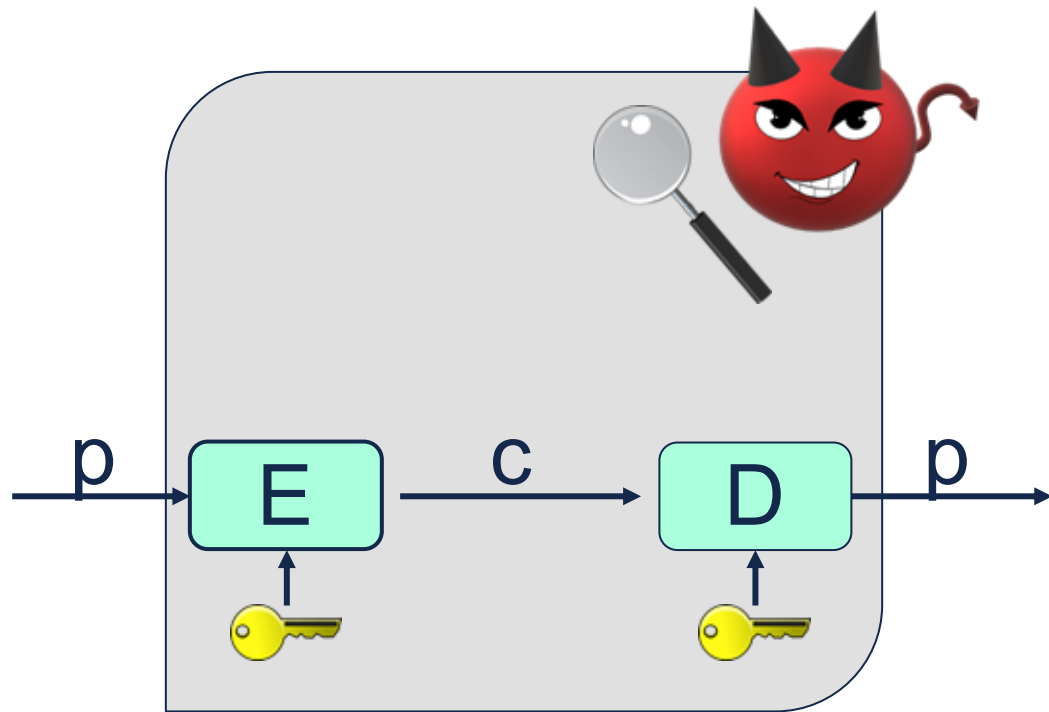
E.g., An encryption algorithm using 128-bit keys is computationally secure against exhaustive search.



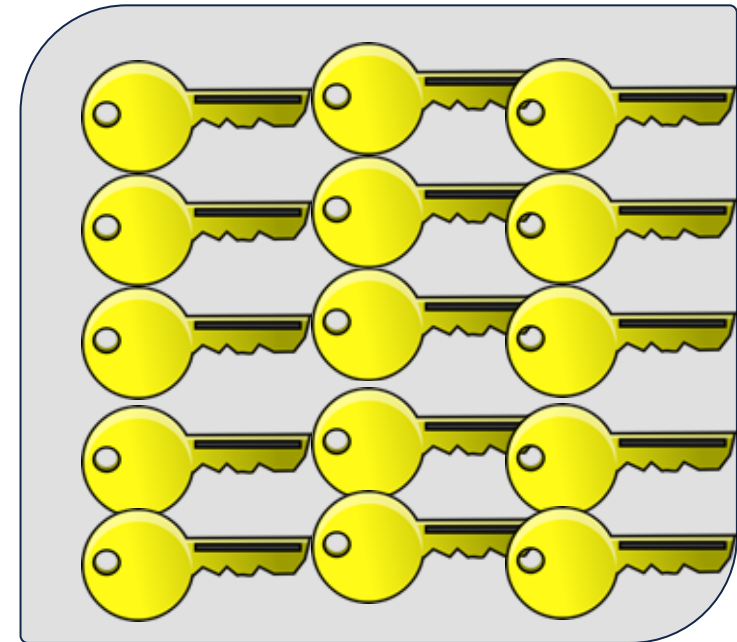
**Any secure encryption
scheme must have a key
space that is not
vulnerable to exhaustive
search**

--Sufficient Key Space Principle

Breaking Encryption



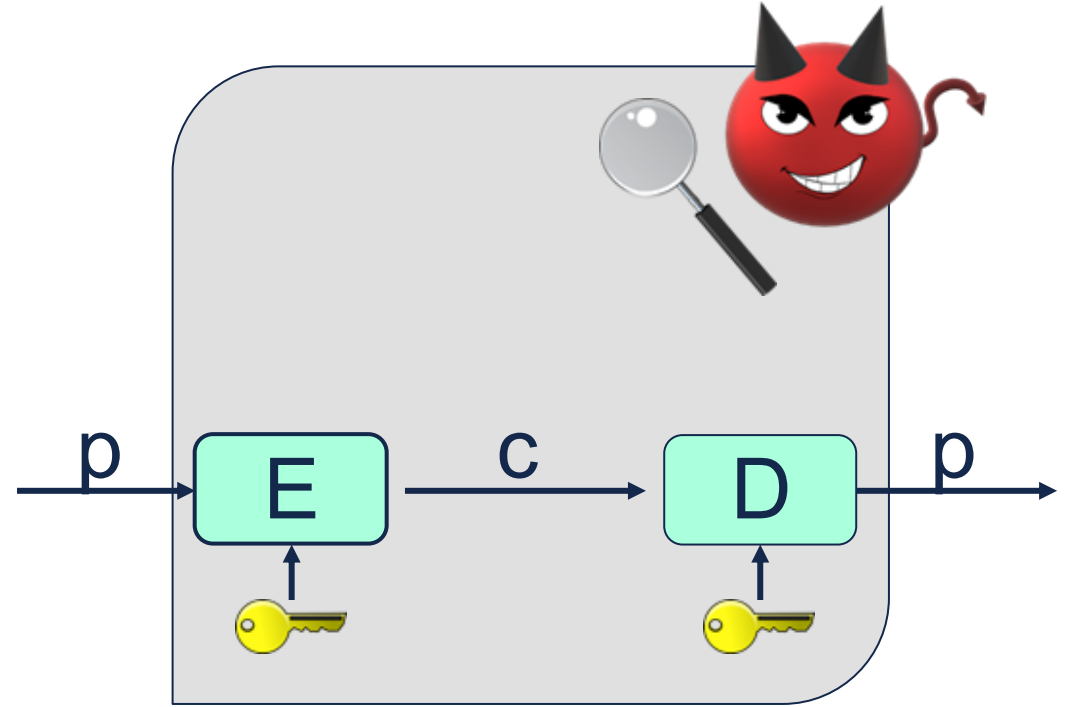
Cryptanalysis



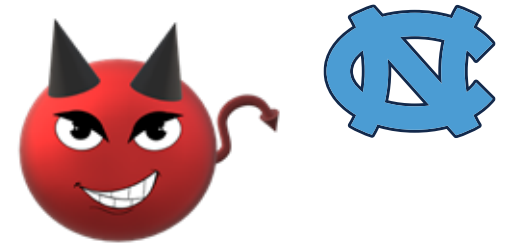
Exhaustive
Search



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



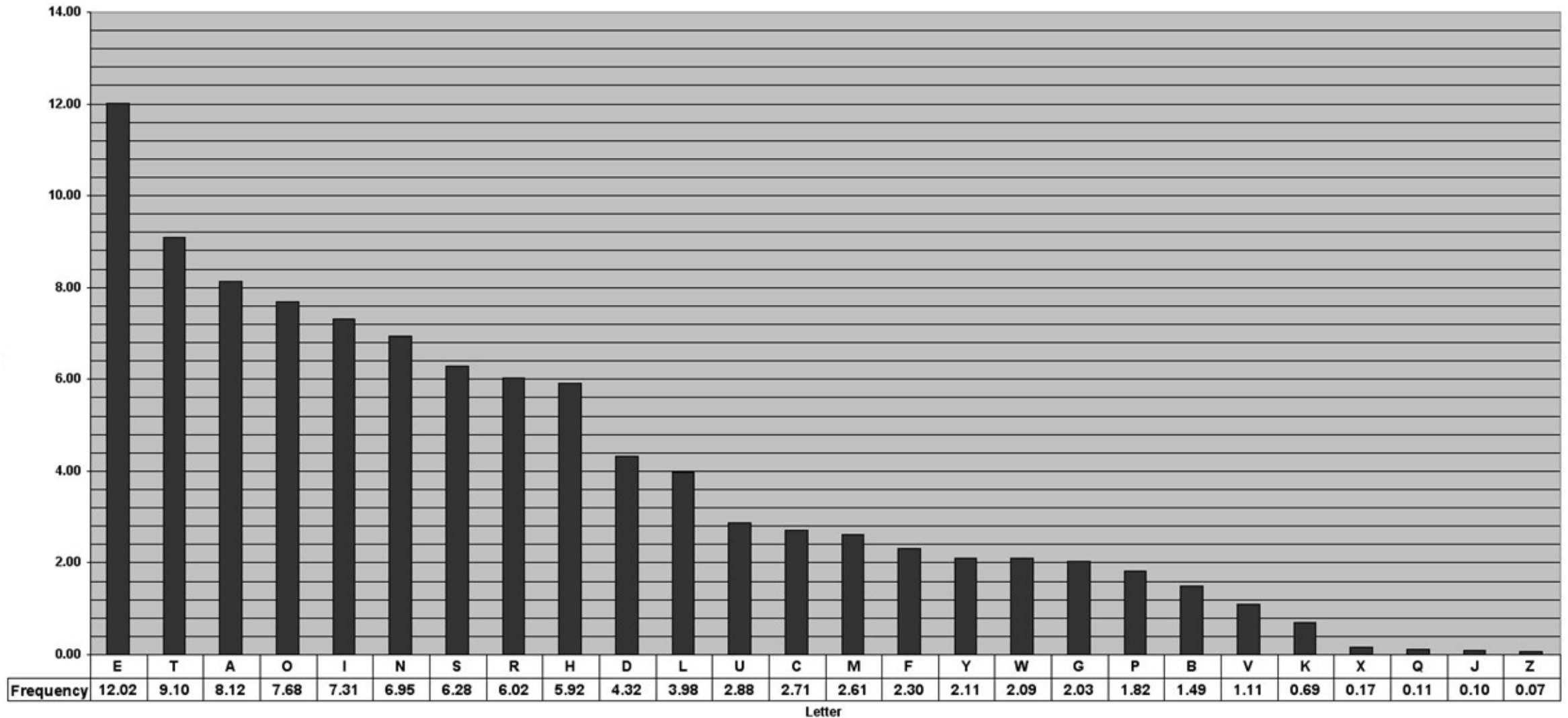
Mono-Alphabetic Substitution

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

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

msg $\in M$, key $\in K$

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



Cipher-text Only Attack

Set-up:

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

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

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

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



Given:

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

Goal:

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

Enc = Mono-alphabetic substitution

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

ct = xmqmqm

→

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

Example

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) ... (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

Set-up:

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

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

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

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

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

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



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

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

Enc = Shift Cipher

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



Attacker can recover the key and
decrypt any future message!

Example

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

ct₁ = **Enc**_{key}(m₁)

ct₂ = **Enc**_{key}(m₂) ...

ct_j = **Enc**_{key}(m_j)

Goal:

Recover the msg such that

msg = **Dec**_{key}(**ct**)

Set-up:

Enc, Dec, M = {0, 1}ⁿ 

K = {0, 1}^p

m ∈ M, key ∈ K

ct = **Enc**_{key}(m)



To be CPA-Secure, a scheme must be
non-deterministic.

Determinism



- **Deterministic encryption:** the same plaintext encrypted with the same key always produces the same ciphertext.
- For CPA, that is dangerous because an attacker can encrypt chosen messages and compare outputs to recognize or infer future messages.

Chosen-Plaintext Attack

Given:

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

ct₁ = **Enc**_{key}(m₁)

ct₂ = **Enc**_{key}(m₂) ...

ct_j = **Enc**_{key}(m_j)

Goal:

Recover the msg such that

msg = **Dec**_{key}(**ct**)

Set-up:

Enc, Dec, M = {0, 1}ⁿ 

K = {0, 1}^p

m ∈ M, key ∈ K

ct = **Enc**_{key}(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

Set-up:

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

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

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

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



Given:

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

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

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

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

Goal:

Recover the msg such that

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

To be CCA-Secure, a scheme must be **non-malleable**.

Malleability



- Malleable: an attacker can modify a ciphertext in a predictable way that causes a related change in the decrypted plaintext.
- An encryption scheme is non-malleable if:
 - If the adversary modifies a given ciphertext, the result is either an illegal ciphertext or one that encrypts a plaintext having no relation to the original one
- E.g., if the attacker is trying to break the encrypted ciphertext c , the attacker is free to flip just the first bit of c and send that new ciphertext c' to the oracle to be decrypted.
 - Therefore, to be secure against CCA attacks, it must be the case that c' is either an illegal ciphertext or is the ciphertext of something totally unrelated to the message that produced c .

Chosen-Ciphertext Attack

Given:

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

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

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

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

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

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)

