Using block ciphers

Security for many-time key

Example applications:

1. File systems: Same AES key used to encrypt many files.
2. IPsec: Same AES key used to encrypt many packets.

Semantic Security for many-time key

Key used more than once $\Rightarrow$ adv. sees many CTs with same key

Adversary's power: chosen-plaintext attack (CPA)

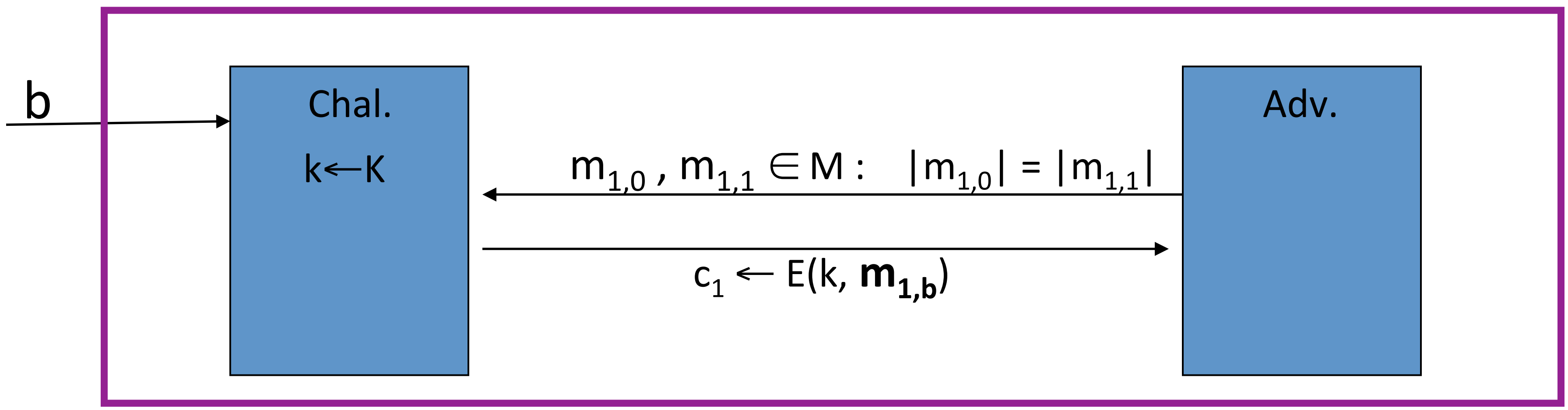
- Can obtain the encryption of arbitrary messages of his choice

(conservative modeling of real life)

Adversary's goal: Break sematic security

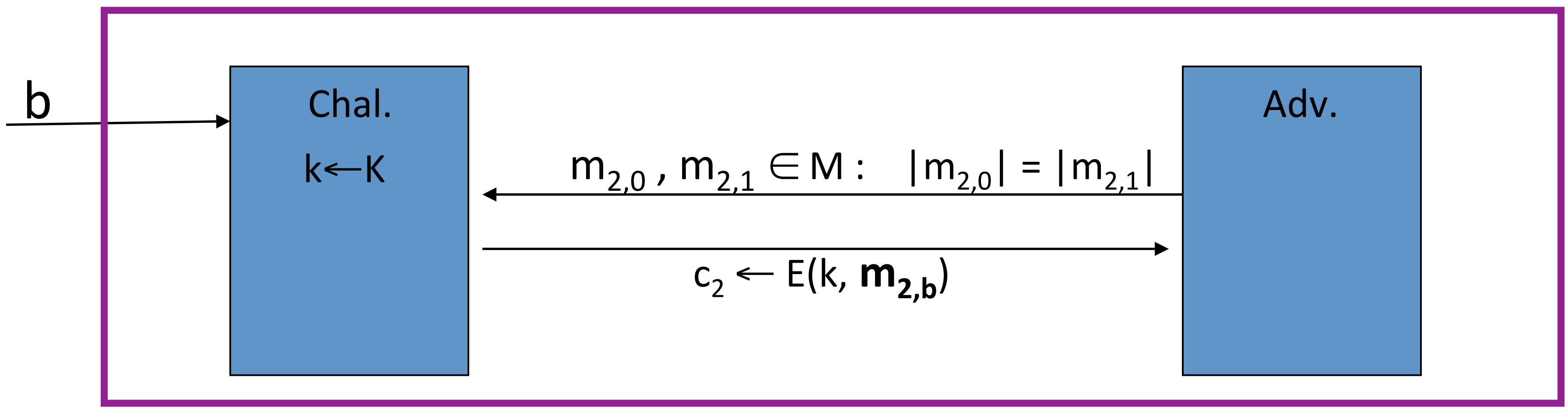
Semantic Security for many-time key

$\mathbb{E}$ = (E,D) a cipher defined over (K,M,C). For b=0,1 define EXP(b) as:



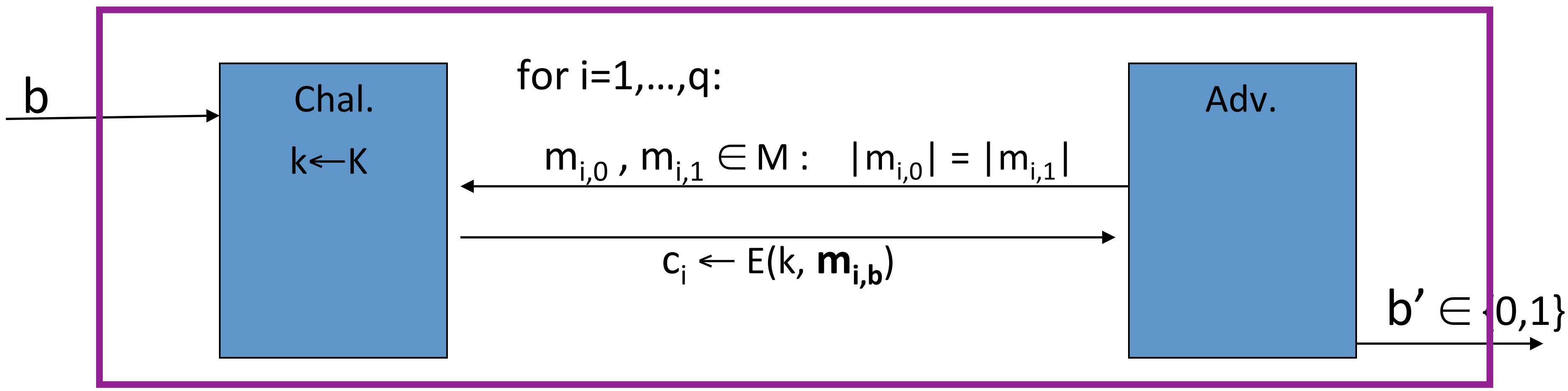
Semantic Security for many-time key

$\mathbb{E}$ = (E,D) a cipher defined over (K,M,C). For b=0,1 define EXP(b) as:



Semantic Security for many-time key (CPA security)

$\mathbb{E}$ = (E,D) a cipher defined over (K,M,C). For b=0,1 define EXP(b) as:



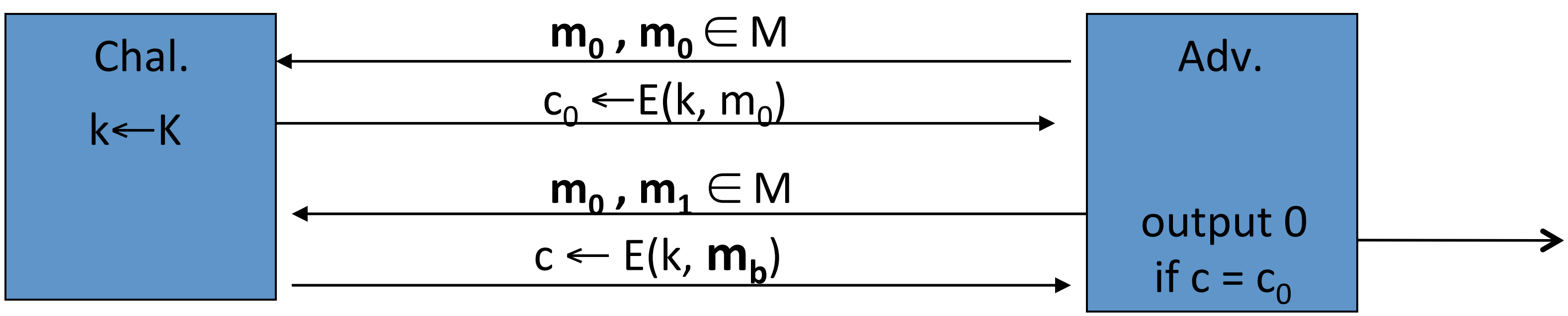
if adv. wants $c = E(k, m)$ it queries with $m_{j,0} = m_{j,1} = m$

Def: $\mathbb{E}$ is sem. sec. under CPA if for all "efficient" A:

$$\mathrm{Adv}_{\mathrm{CPA}}[A,\mathbb{E}] = \Big| \Pr[\mathrm{EXP}(0)=1] - \Pr[\mathrm{EXP}(1)=1] \Big| \quad \text{is "negligible."}$$

Ciphers insecure under CPA

Suppose E(k,m) always outputs same ciphertext for msg m. Then:

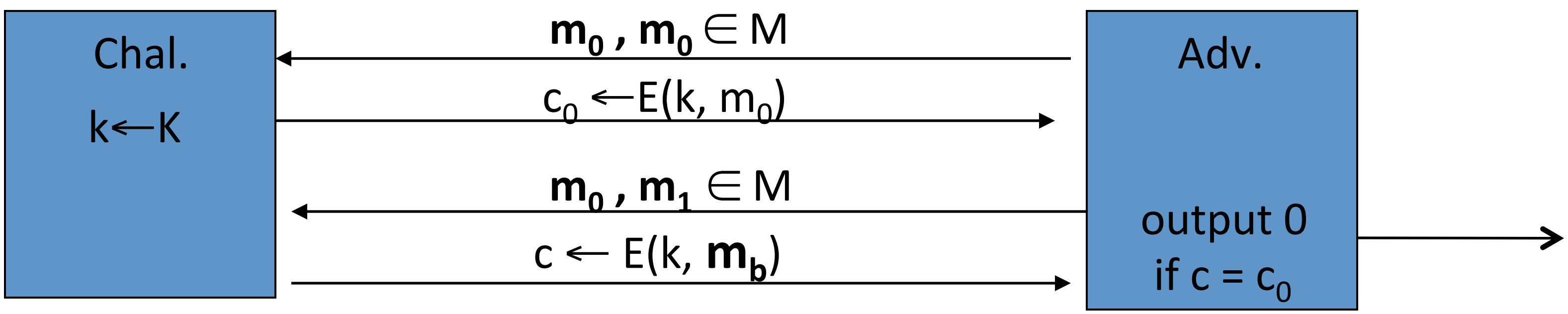


So what? an attacker can learn that two encrypted files are the same, two encrypted packets are the same, etc.

- Leads to significant attacks when message space M is small

Ciphers insecure under CPA

Suppose E(k,m) always outputs same ciphertext for msg m.   Then:

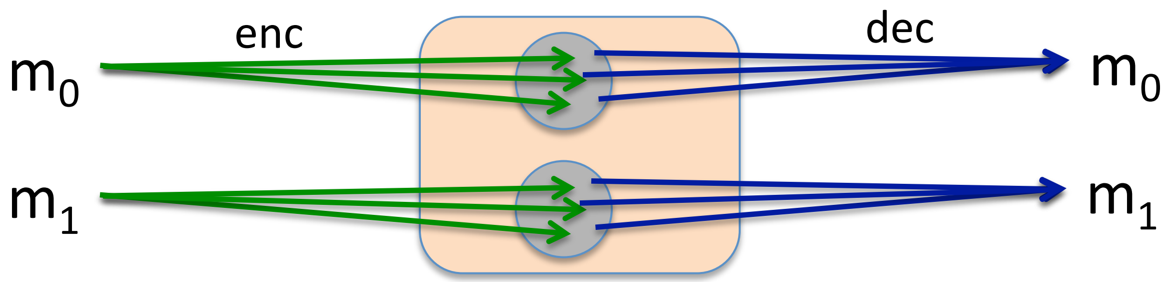


If secret key is to be used multiple times $\Rightarrow$

given the same plaintext message twice,
encryption must produce different outputs.

Solution 1: randomized encryption

- E(k,m) is a randomized algorithm:



$\Rightarrow$ encrypting same msg twice gives different ciphertexts (w.h.p)

$\Rightarrow$ ciphertext must be longer than plaintext

Roughly speaking: CT-size = PT-size + "# random bits"

Let $F: K \times R \longrightarrow M$ be a secure PRF.

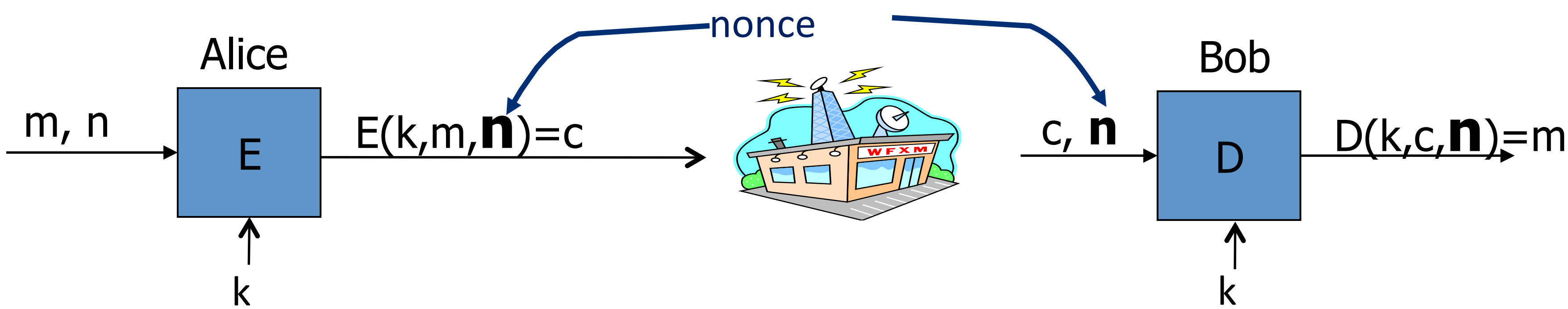
$\approx_p (r, f(r) \oplus m)$

For $m \in M$ define $E(k,m) = [\; r \xleftarrow{R} R$, output $(r, F(k,r) \oplus m) \;]$

Is E semantically secure under CPA?

- Yes, whenever F is a secure PRF
- No, there is always a CPA attack on this system
- Yes, but only if R is large enough so r never repeats (w.h.p)
- It depends on what F is used

Solution 2: nonce-based Encryption



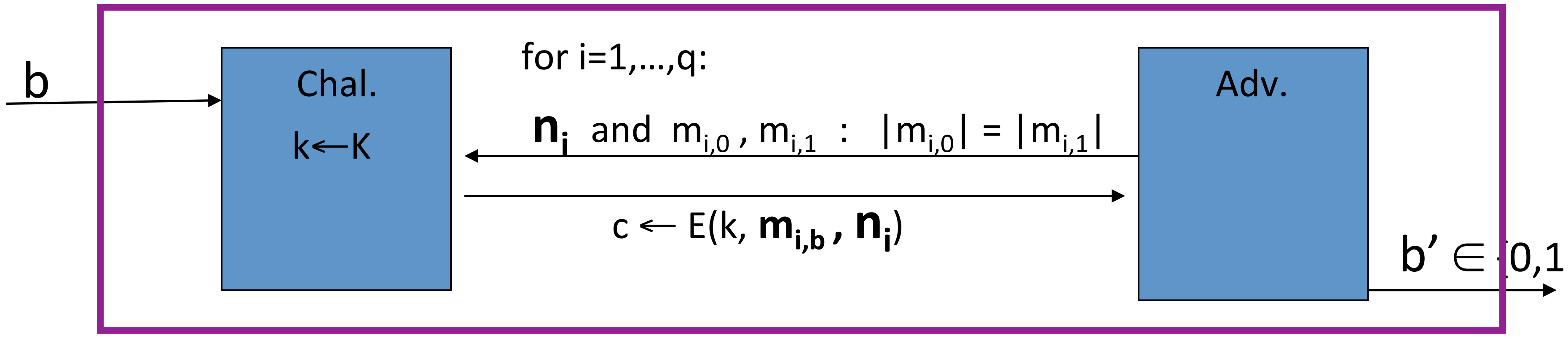
- nonce n: a value that changes from msg to msg.
  (k,n) pair <u>never</u> used more than once

- <u>method 1</u>: nonce is a **counter** (e.g. packet counter)
  - used when encryptor keeps state from msg to msg
  - if decryptor has same state, need not send nonce with CT

- <u>method 2</u>: encryptor chooses a **random nonce**, $n \leftarrow \mathcal{N}$

CPA security for nonce-based encryption

System should be secure when nonces are chosen adversarially.



All nonces $\{n_1, \ldots, n_q\}$ must be distinct.

Def: nonce-based E is sem. sec. under CPA if for all “efficient” A:

$$\mathrm{Adv}_{\mathrm{nCPA}}[A,E] = \Big|\Pr[\mathrm{EXP}(0)=1] - \Pr[\mathrm{EXP}(1)=1]\Big|$$ is “negligible.”

Let $F: K \times R \longrightarrow M$ be a secure PRF. Let $r = 0$ initially.

For $m \in M$ define $E(k,m) = [\ r{+}{+},\ \text{output } \mathbf{(r, F(k,r) \oplus m)}\ ]$

$\approx_p (r, f(r) \oplus m)$

Is E CPA secure nonce-based encryption?

- ○ Yes, whenever F is a secure PRF
- ○ No, there is always a nonce-based CPA attack on this system
- ○ Yes, but only if R is large enough so r never repeats
- ○ It depends on what F is used

End of Segment