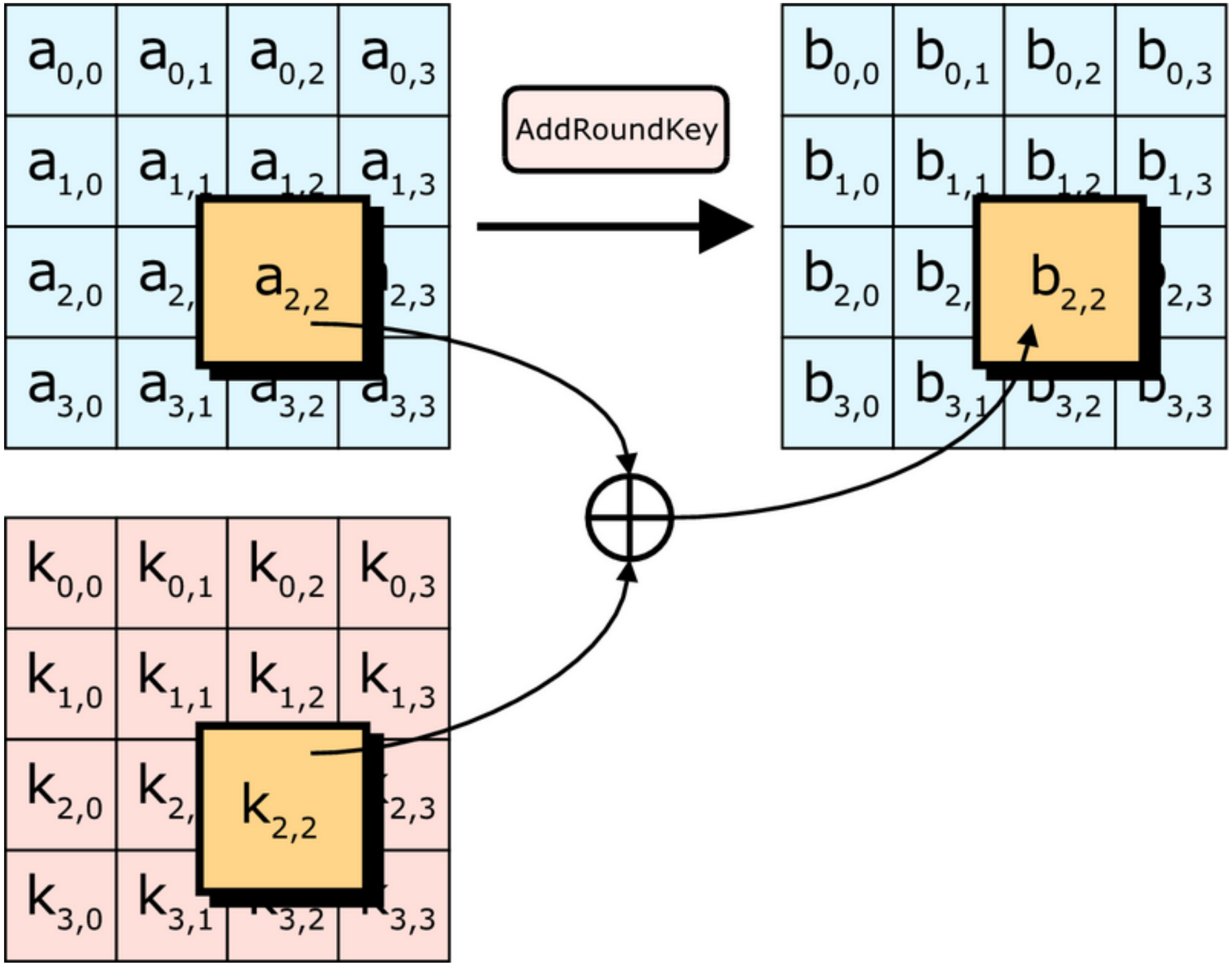


Session 4
Chiffrement par Bloc: Utilisation et Analyse

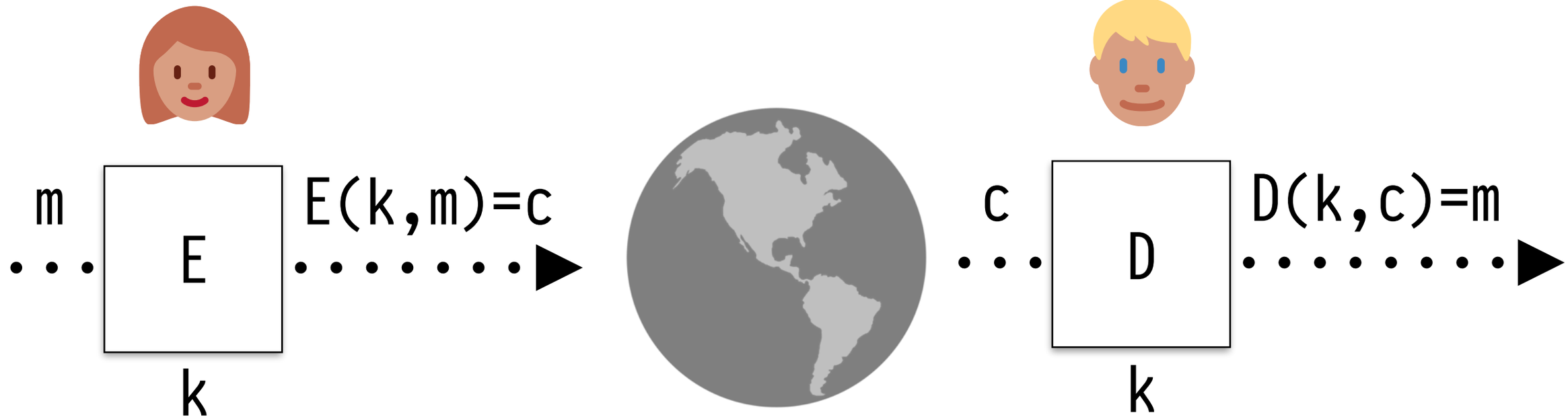
Introduction à la Cryptographie

Nadim Kobeissi

Chiffrement par Bloc

- Méthode de chiffrement d'un texte, utilisant une clé.
- La taille du texte et de la clé sont **fixes** et divisés en *blocs*.
- Cela permet plus de contrôle sur le propriétés du chiffrement.
- Cette session est pratique — le cours prochain sera un traitement plus théorique des chiffrements par bloc.

Rappel: Chiffrement Symétrique



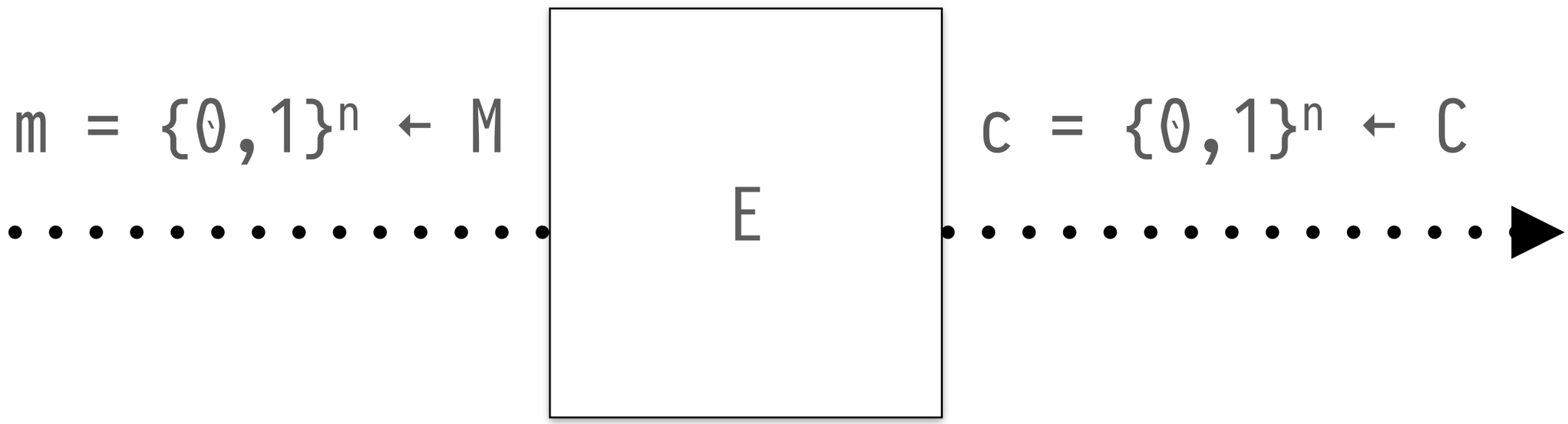
Pas d'espionnage,
Pas d'alteration.

m: Message, k: Clé
E, D: Fonctions de chiffrement (connus)
c: Message chiffré

E et D sont *efficaces, inversibles*.

m, k, et c sont de taille déterminée!

Un Regard Plus Proche



$k = \{0,1\}^x \leftarrow K$

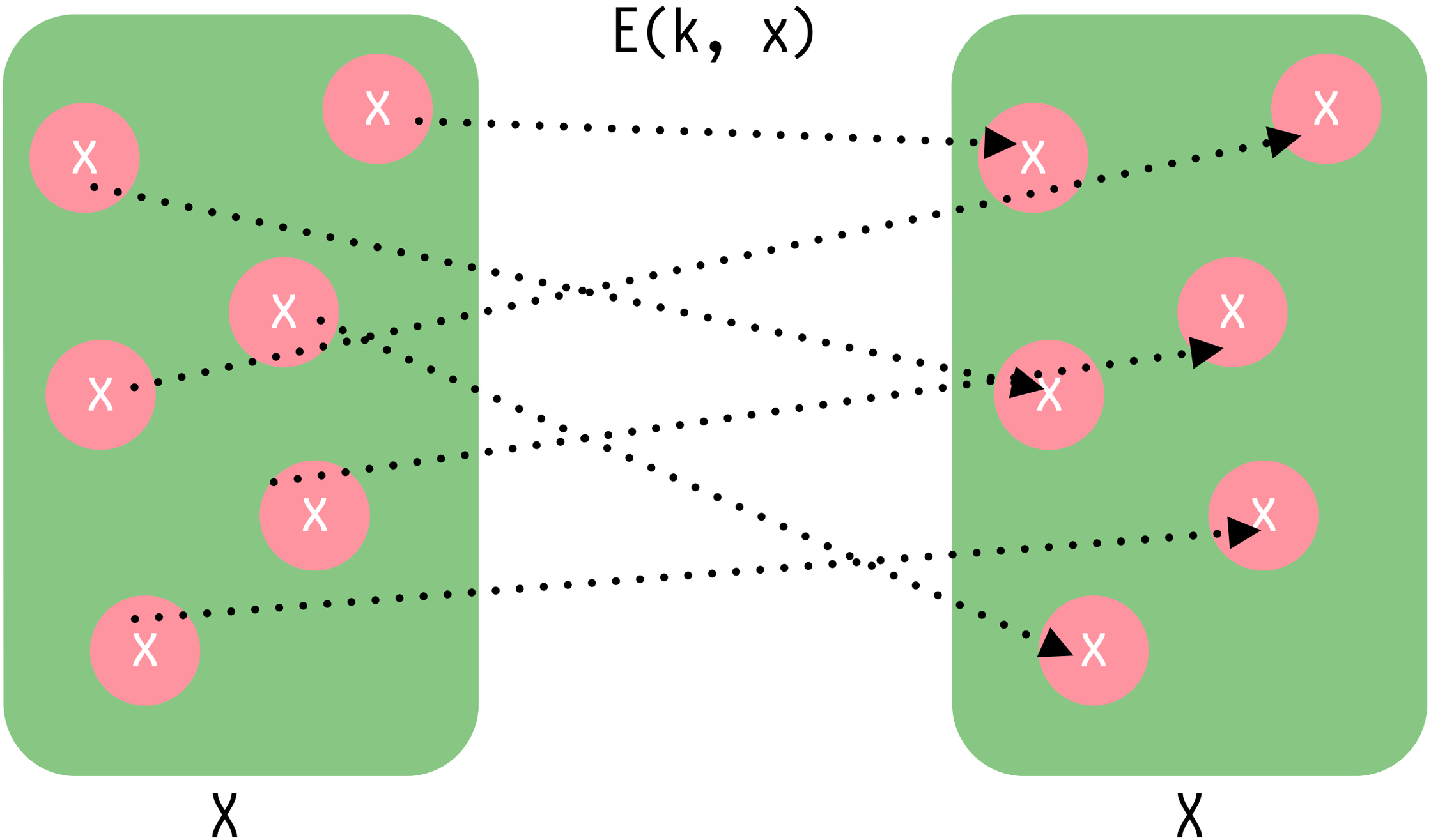
Exemples:

- 3DES: n = 64, x = 168
- AES: n = 128, x = 128, 192, 256

Fonctions Aléatoires: Toujours Notre But

- `X(U) → V`
- Exemple: `X(U,I) → V`
- *Nouvelle definition!* Permutation Aléatoire:
- `E(K,X) → X`, tel que:
  - Il existe une façon “efficace” de évaluer cette permutation.
  - Il existe un algorithme d’inversion `D(K,Y)` “efficace” aussi.
  - La fonction `E(K,·)` est une fonction “one-to-one” (bijection)

Bijection (fonction “one-to-one”)



Attaques Linéaires et Différentiels

- D*efinition de base*: donné plusieurs paires de input/ output d'un chiffrement, on peut récupérer la clés plus rapidement qu'une recherche exhaustive.

- (Plus rapidement que 2^{56} dans le cas to DES.)

- Une "attaque" implique qu'on a *brisé* un chiffrement.

“Brisé”? Vraiment?

- Avoir “brisé” un chiffrement veut dire quelque chose de different pour un:
  - *Chercheur en cryptographie théorique*: Une manière de trouver la clé plus rapide q’une recherche exhaustive.
  - *Ingénieur en cryptographie appliquée*: Une manière de “pratiquement” et “rapidement” récupérer une clé, un message clair, passer au dessus d’une verification d’intégrité…

Attaques Linéaires

- $Pr[m[i_{1 \ldots} \oplus_{\ldots r}] \oplus c[j_{j \ldots} \oplus_{\ldots v}] = k[l_{1 \ldots} \oplus_{\ldots u}]] = 0.5 + \varepsilon$

- ε est une linéarité. Une relation linéaire qui se manifeste come un bias.

- Un ε indique que $m[i_{1 \ldots} \oplus_{\ldots r}] \oplus c[j_{j \ldots} \oplus_{\ldots v}]$ produira la clé une majorité du temps.

Attaques Linéaires: DES

- $Pr[m[i_{1 \ldots} \oplus_{\ldots r}] \oplus c[j_{j \ldots} \oplus_{\ldots v}] = k[l_{1 \ldots} \oplus_{\ldots u}]] = 0.5 + \varepsilon$

- Dans DES, il existe un $\varepsilon = 0.0000000477$ (a cause d'une linéarité dans le 5eme S-Box).

- Avec 2^{42} paires input/output, on peut determiner 14 bits de k en 2^{42}. Donc il reste 43 bits.

- *Question*: Quel est le nombre d'operations total pour obtenir la clé?

$\sim 2^{43}$

Attaques Quantiques

- Une recherche exhaustive avec une espace de clés K demande:
  - *Sur un ordinateur conventionnel:* $|K|$ operations.
  - *Sur un ordinateur quantique:* $|K|^{0.5}$ operations.
- AES: $|K| = 2^{64,96,128}$, DES: $|K| = 2^{28}$
- Il est inconnu si les ordinateurs quantiques sont pratiquement possibles a construire.

Comment Chiffrer Avec Plusieurs Blocs?

- La plupart des chiffrements par bloc ont une taille de blocs très petite (16 bytes).

- Voici 16 bytes: Bonjour, mon nom

- On doit pouvoir utiliser les chiffrements par blocs pour chiffrer des messages plus longues que ça!

Sécurité Sémantique Pour Clés Re-utilisés

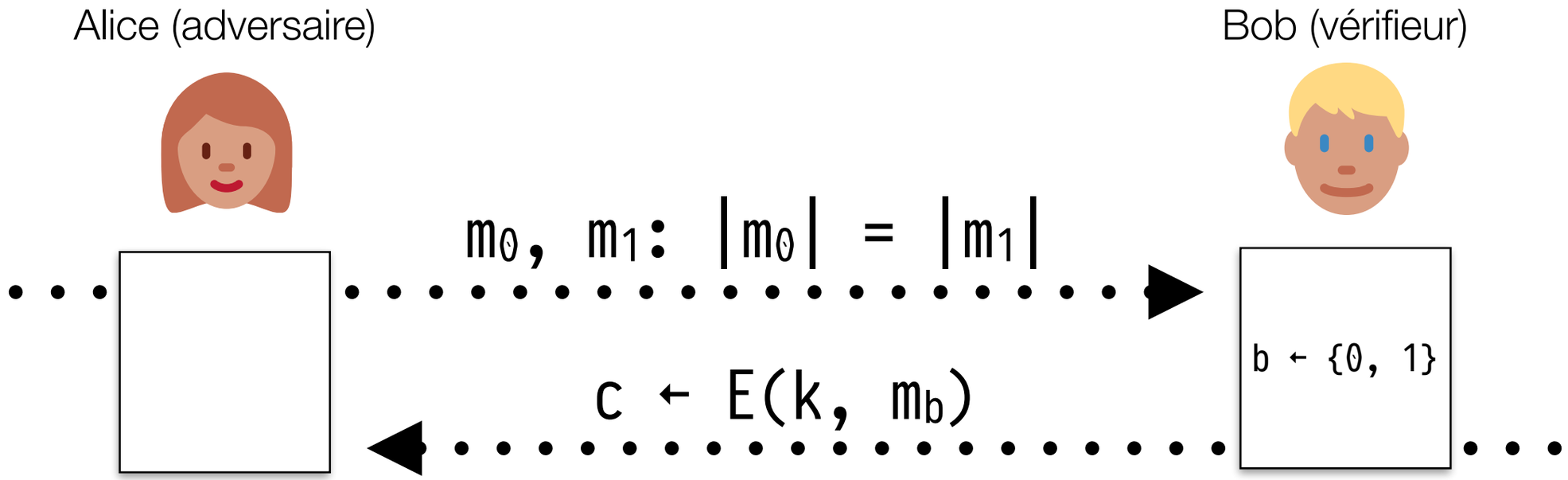
- Utiliser la même clé plusieurs fois → l'adversaire a access a plusieurs chiffrements sous la même clé.

- Comment définir un modèle de sécurité sous cette contrainte?

- **Pouvoir de l'adversaire:** il peut demander des chiffrements d'un nombre arbitraire de messages clairs de son choix. (Modèle CPA ("chosen plaintext attack"))

Sécurité Sémantique

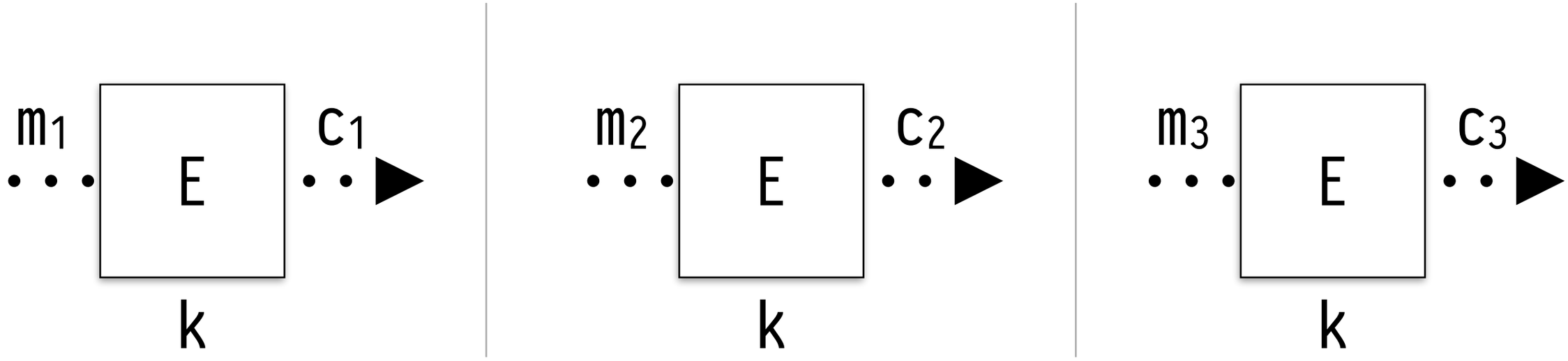
On définit deux experiments `Exp(0)` et `Exp(1)`, tel que:



`E` est *sémantiquement sur* si Alice a un avantage *negligible* avec lequel deviner la valeur de `b` en utilisant `c`.

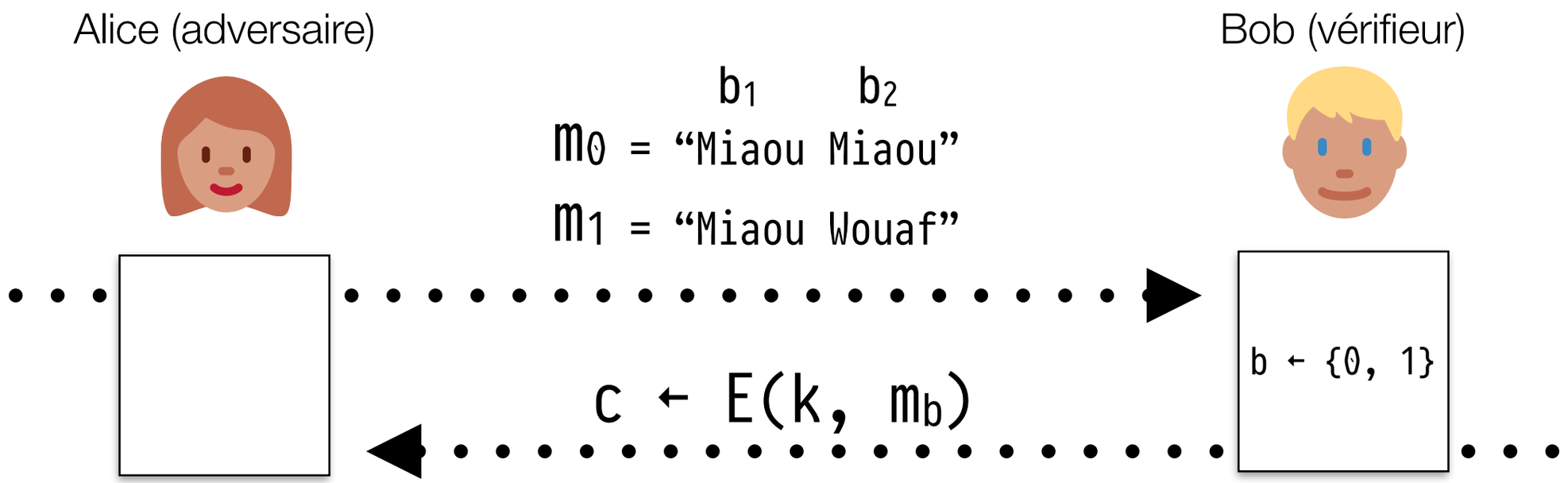
ECB: Electronic Codebook Mode

- On divise le texte clair en plusieurs blocs.
- On applique le chiffrement avec la même clé sur chaque bloc.



Sécurité Sémantique de ECB

ECB n’est pas sémantiquement sur pour tous les applications avec plus que un bloc.

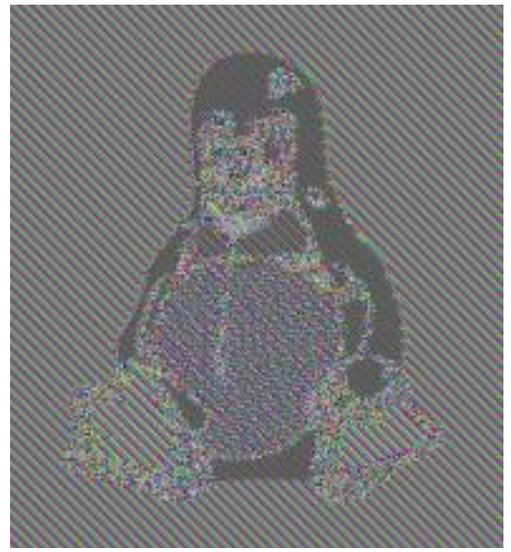


Alice saura distinguer b_2 ∋ m_1 de b_2 ∋ m_2 chaque fois.

ECB: Electronic Codebook Mode

- En photos:

Message clair



Message chiffré

Vous devrez jamais utiliser ECB.

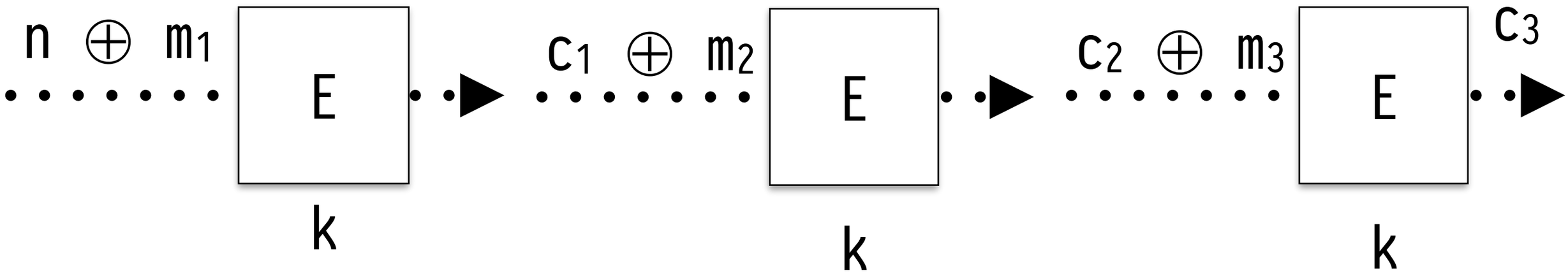
Comment Satisfaire La Sécurité CPA?

- …si on re-utilise la même clé?
- On a intérêt a que les résultats des chiffrements soient different même si on re-utilise la même clé avec le même message.

Exemple Simple: CBC (Cipher Block Chaining)

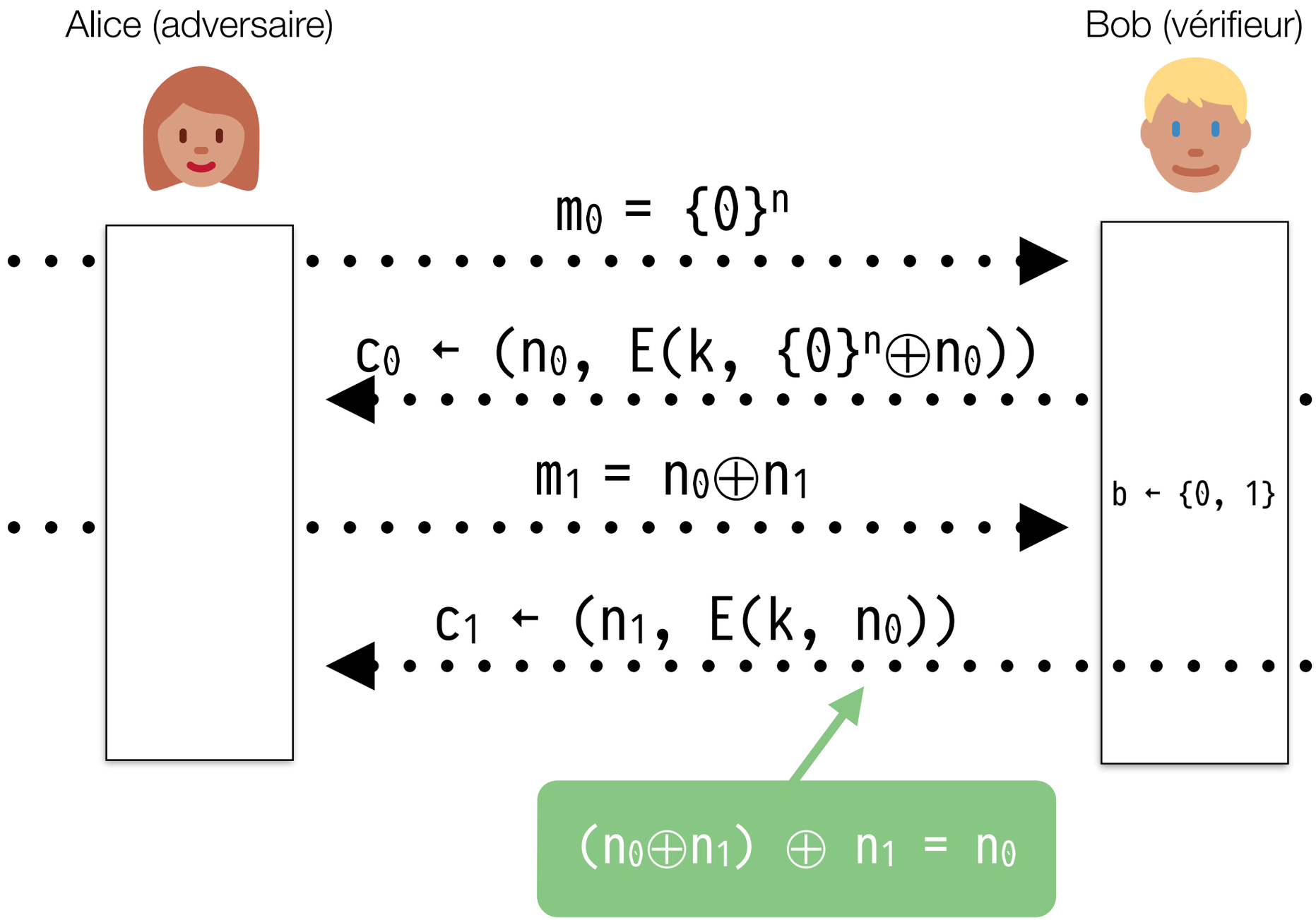
- On utilise un **n** pseudo-aléatoire pour générer un chiffrement avec des notions pseudo-aléatoires même si on re-utilise **k**.

```
n = 0a4e7ad3aa3890fa0a4e7ad3aa3890fa
```



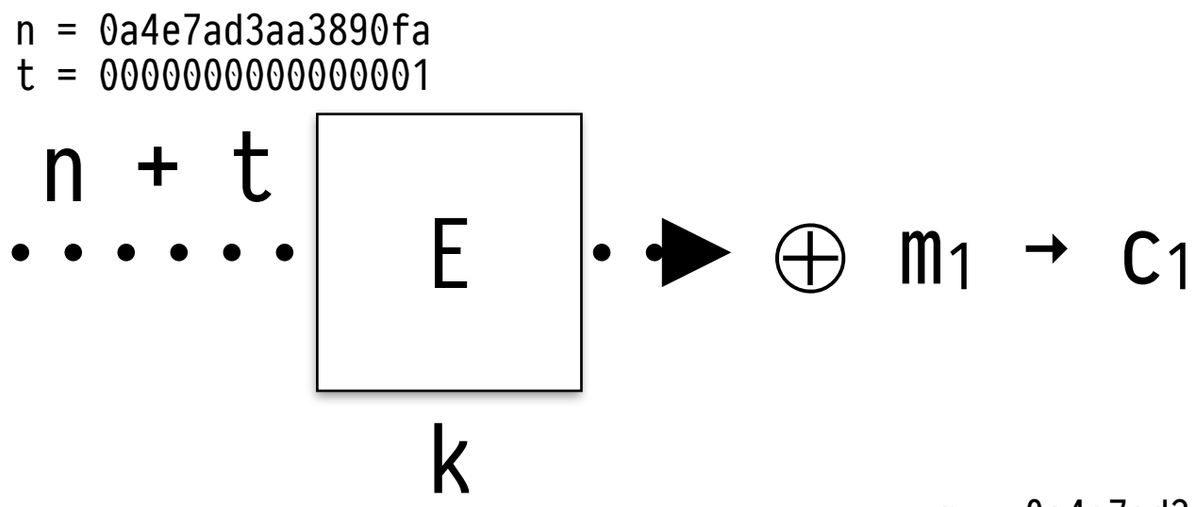
Mode CBC: n Doit Être Imprévisible

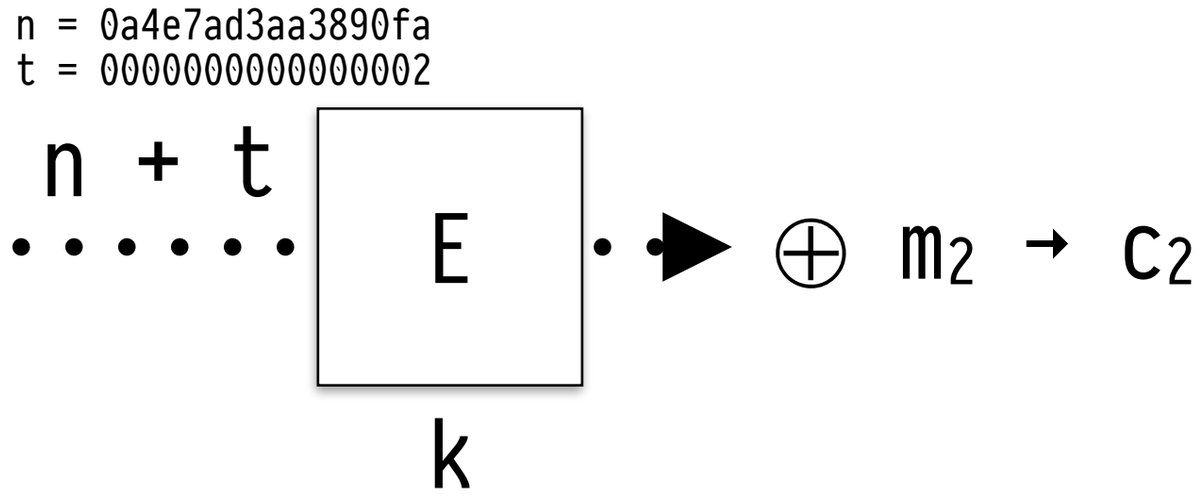
Si Alice peut prédire le prochain n, CBC n'est pas CPA-sur.



CTR: Counter Mode

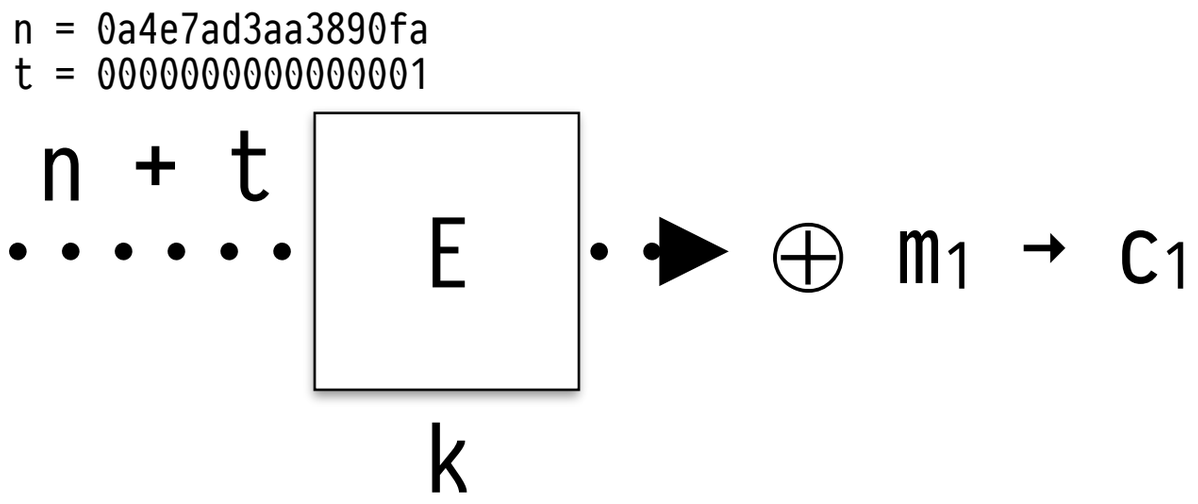
- On utilise le chiffrement par bloc pour bâtir un chiffrement de flux!





CTR: Counter Mode

- On envoie `(c, n)` comme notre message chiffré.
- Mais il existe une faille potentielle…
- Et si on utilise le même **n** et **k** pour deux messages?



- $c_1 = E(k, n) \oplus m_1$
- $c_2 = E(k, n) \oplus m_2$
- $c_1 \oplus c_2 = m_1 \oplus m_2$

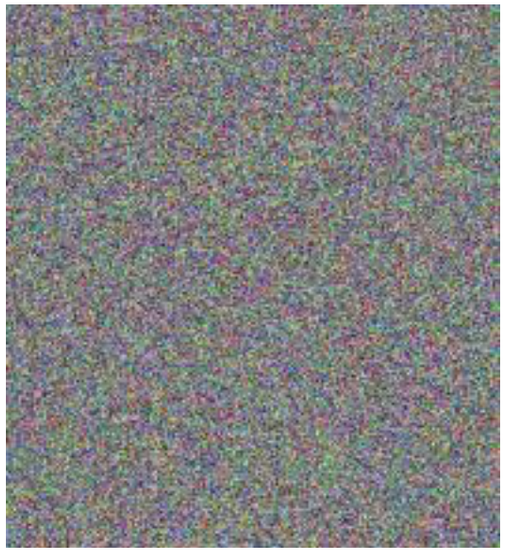
CTR: Counter Mode

- *Question*: Est-ce que le chiffrement avec les mode CTR est sur contre un attaquant dans le modèle CPA?
  - Oui…
  - …mais seulement si l'espace de **n** possibles est suffisamment large pour satisfaire que **n** ne se répète jamais.

Modes CBC et CTR

- En photos:

Message clair



Message chiffré

Souvenez-vous que un XOR avec une clé uniformément distribuée donne un résultat uniformément distribué.

Suivez le Cours En Ligne

- http://courscrypto.org
  - Matériaux.
  - Devoirs/TPs.
  - Slides et vidéos.
  - **Deuxième partie commence le 22 Aout!**

Je conseille vivement lire ce livre