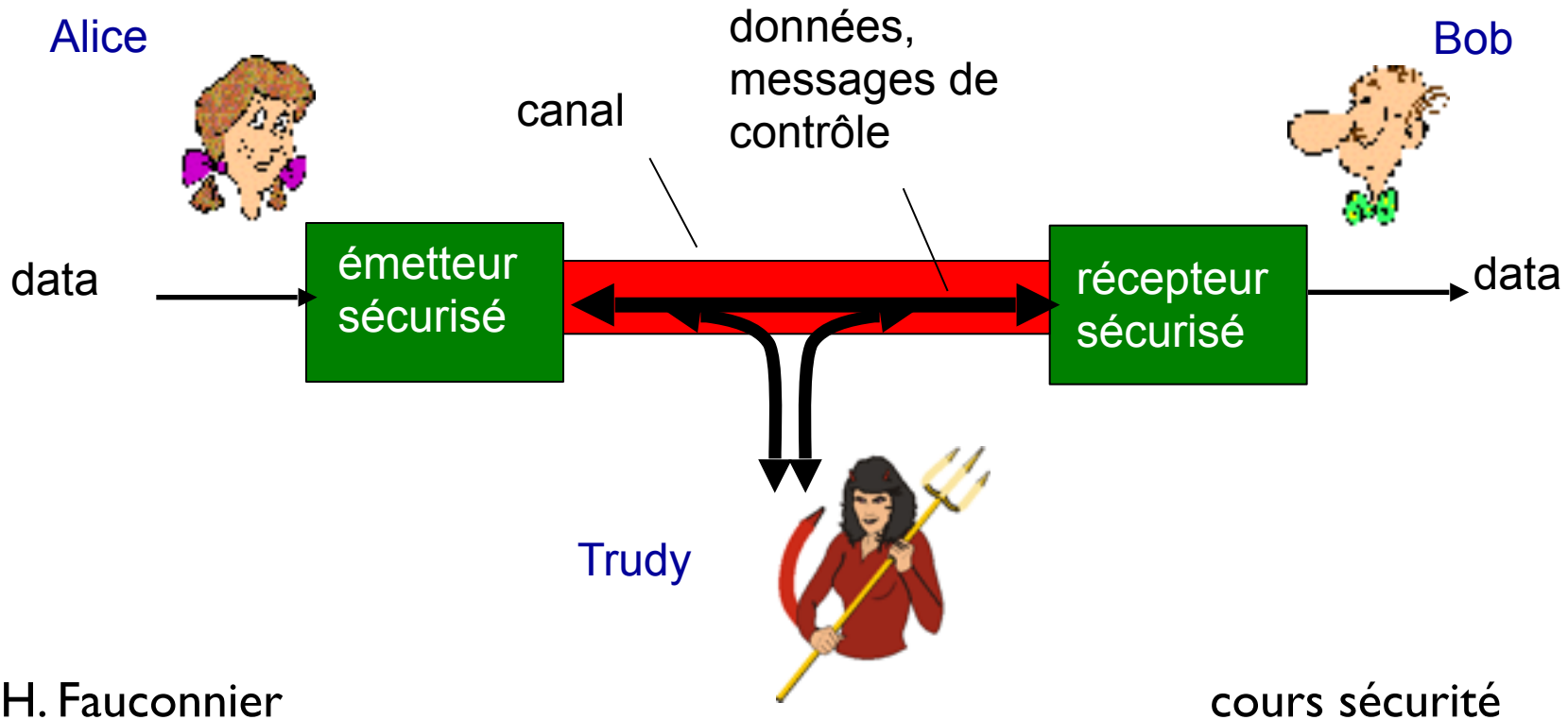
Cours sécurité informatique: clés symétriques et asymétriques signatures numériques

Modèle de base pour la communication

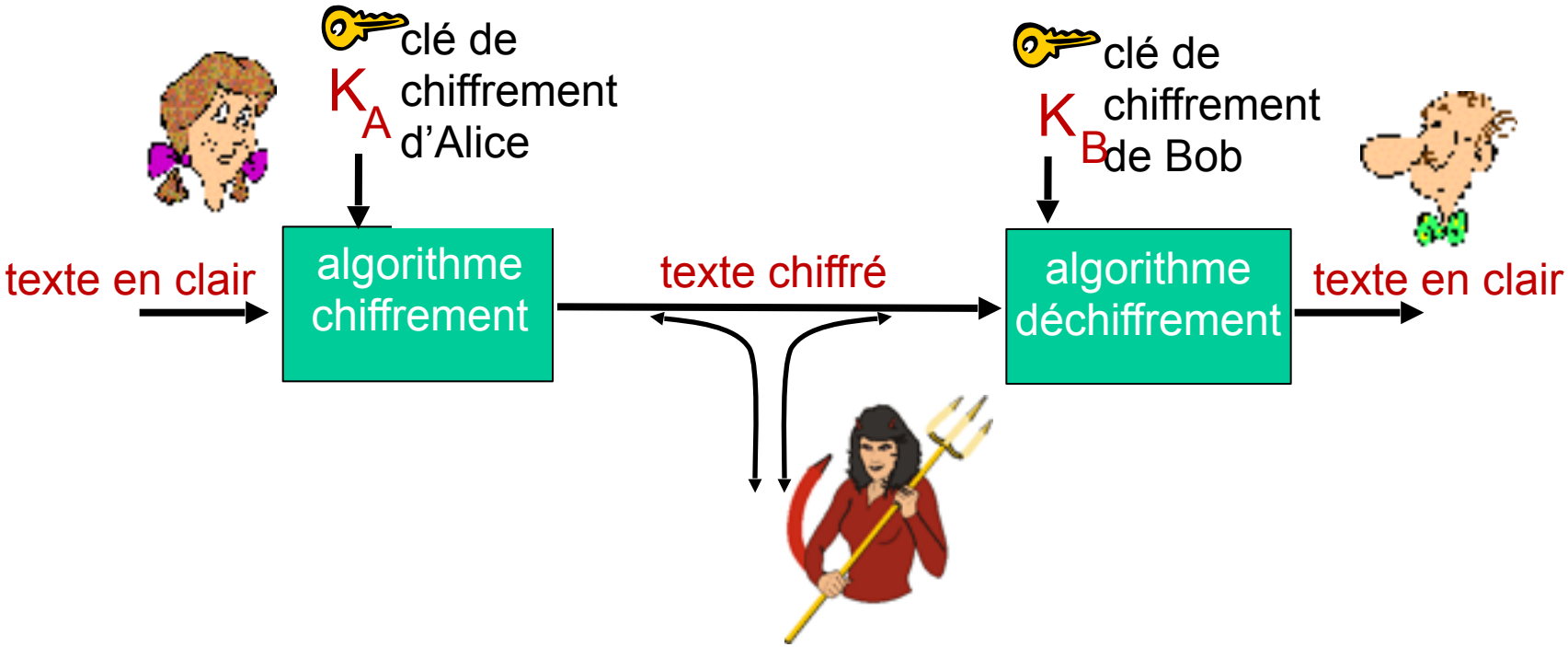
- Bob, Alice veulent communiquer de façon sûre
- Trudy (l'intruse) peut intercepter, supprimer, ajouter des messages



Que peut faire un bad guy (réseau)

- *espionner (eavesdrop):* intercepter des messages
- *insérer* des messages
- *imposture (impersonation):* mettre de fausses adresses sources dans les paquets (parodie - spoof)
- *pirater (hijacking) :* prendre la place de l'émetteur ou du récepteur
- *dénis de service:* empêcher le service de fonctionner (surcharge des ressources)

Le langage de la cryptographie



m message en clair

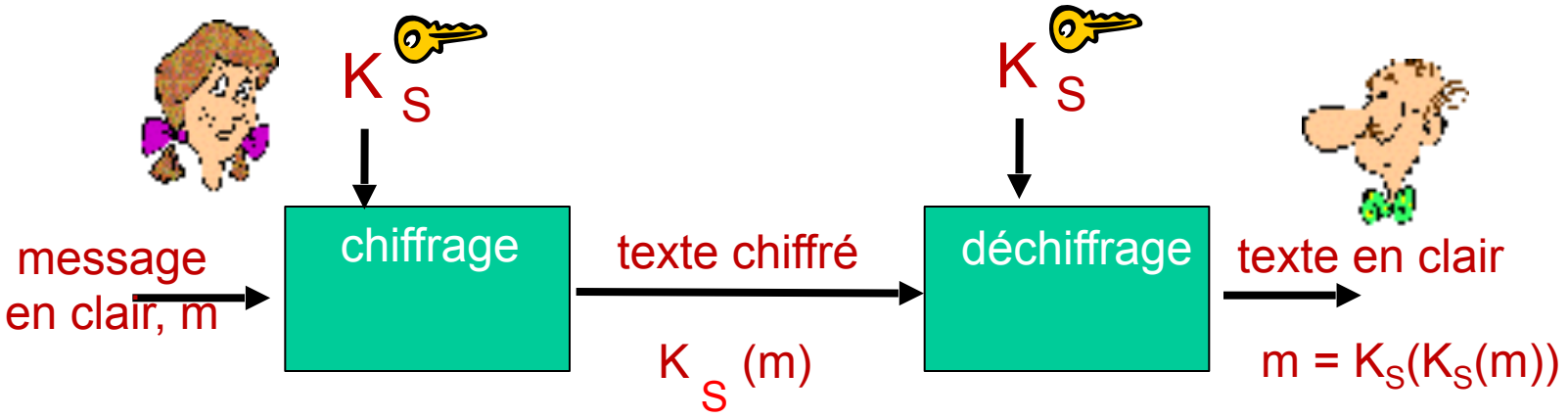
$K_A(m)$ message chiffré avec la clé K_A

$m = K_B(K_A(m))$

Comment casser ce schéma?

- attaque avec le texte chiffré seul: Trudy a le texte chiffré qu'elle peut analyser
- deux approches:
  - force brute: essayer toutes les clés
  - analyse statistique du texte
- attaque avec du texte en clair: Trudy a un texte en clair correspondant a un texte chiffré
- attaque avec du texte en clair choisi: Trudy peut obtenir des textes chiffrés à partir de textes en clair

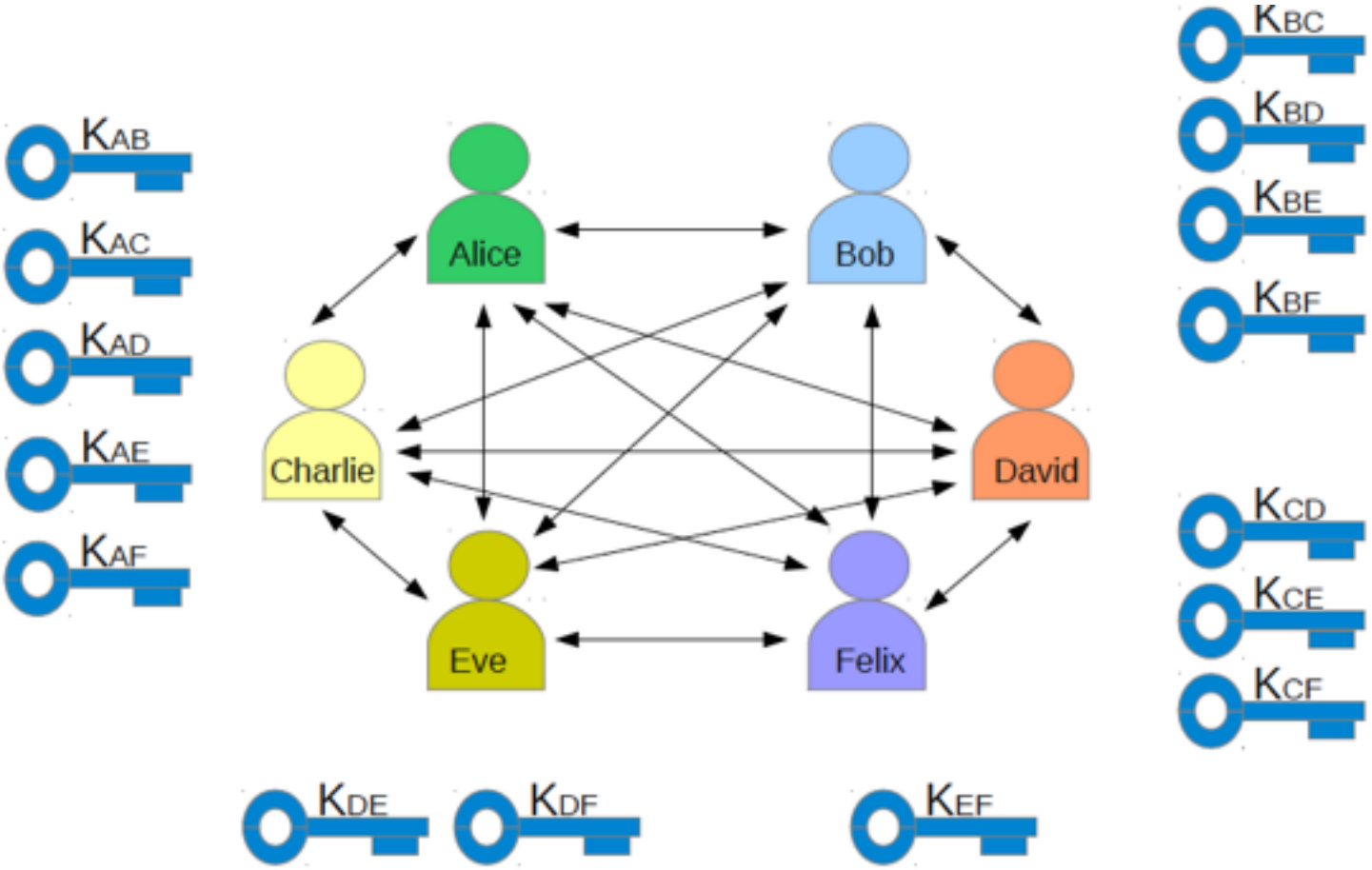
Cryptographie symétrique



cryptographie à clés symétriques: Bob et Alice partagent la même clé (symétrique) K_S

Problème: Comment Bob et Alice obtiennent la clé?

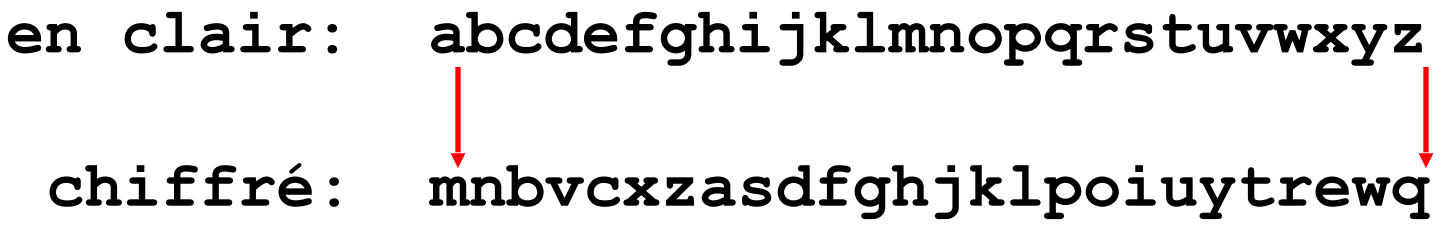
Des clefs:


KAB
KAC
KAD
KAE
KAF
Alice
Bob
Charlie
David
Eve
Felix
KBC
KBD
KBE
KBF
KCD
KCE
KCF
KDE
KDF
KEF

Un chiffrement très simple

chiffrement par substitution:

- chiffrement mono-alphabétique (César): substituer une lettre par une autre



e.g.:

```
en clair: bob. i love you. alice
  chiffré: nkn. s gktc wky. mgsbc
```

Chiffrement clé de codage: application d'un ensemble de 26 lettres dans un ensemble de of 26 lettres

Plus élaboré

- n codes à substitution: $M_1, M_2, \ldots, M_n$
- motifs cycliques:
  - exemple n=4: M_1, M_3, M_4, M_3, M_2; M_1, M_3, M_4, M_3, M_2; ..
- pour chaque nouveau symbole utiliser cycliquement le motif de substitution suivant:
- dog: d de M_1, o de M_3, g de M_4

Clé de chiffrement: n codes à substitution, et un motif de substitution

- la clé n'est pas seulement un motif de n bits

DES: chiffrement symétrique

DES: Data Encryption Standard

- US standard [NIST 1993]
- Clef symétrique 56-bit, texte en clair de 64-bit
- chiffrage par blocs avec chainage des chiffrages
- DES est-il sûr?
  - DES Challenge: 56-bit-key-encrypted phrase déchiffrée(brute force brute) en moins d'un jour
  - pas de « bonne » attaque analytique connue
- DES plus sûr:
  - 3DES: chiffrement 3 fois avec 3 clés de chiffrement

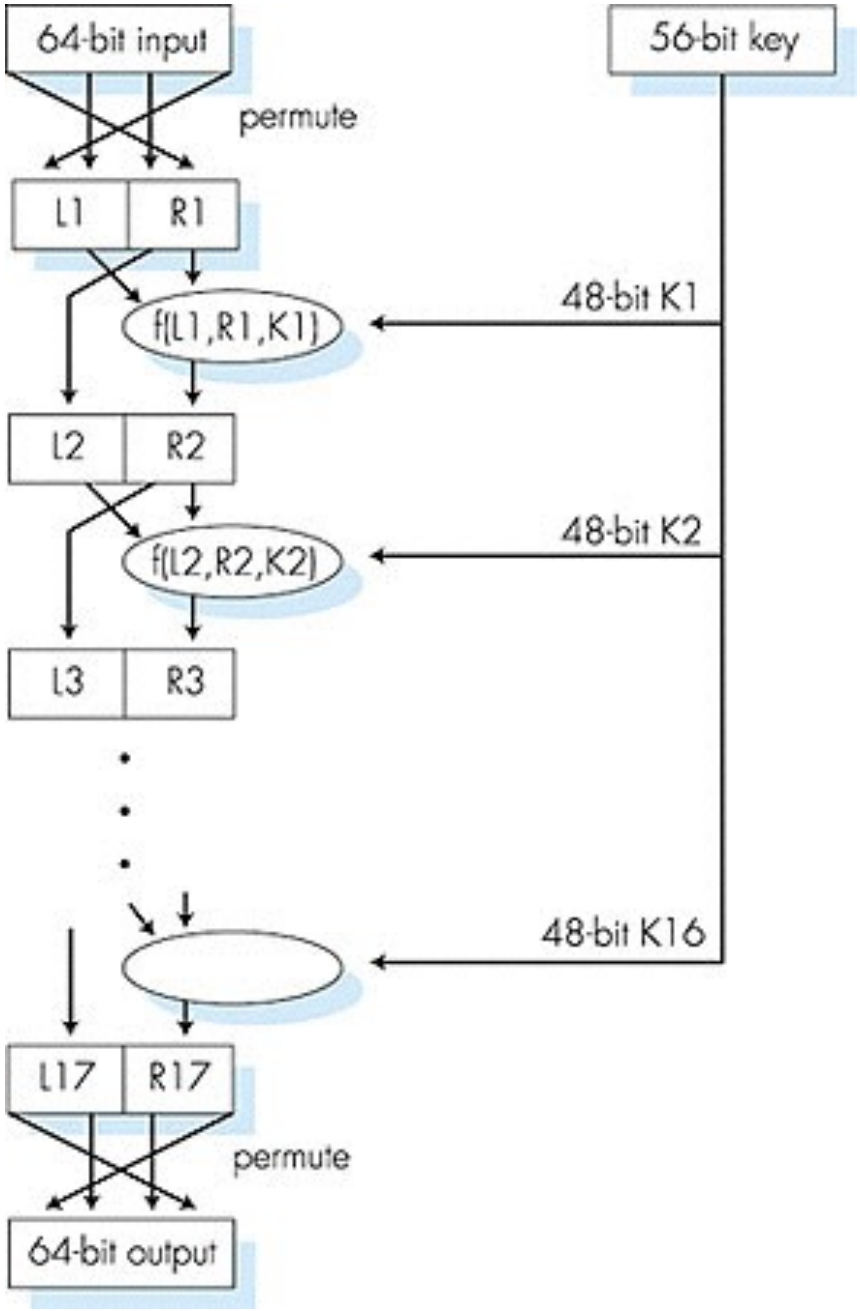
DES: code symétrique

DES:

permutation initiale

16“rondes” identiques d’application de fonctions, chacune utilisant 48 bits différents de la clé

permutation finale



AES: Advanced Encryption Standard

- symmetric-key le standard NIST à clé symétrique qui remplace DES (Nov 2001)
- données traitées par blocs de 128 bits
- clés de 128, 192 ou 256 bits
- un décryptage en force brute (essayer chaque clé) prenant 1 sec avec DES, prend 149 trillion d'année pour AES

Cryptographie à clés publiques

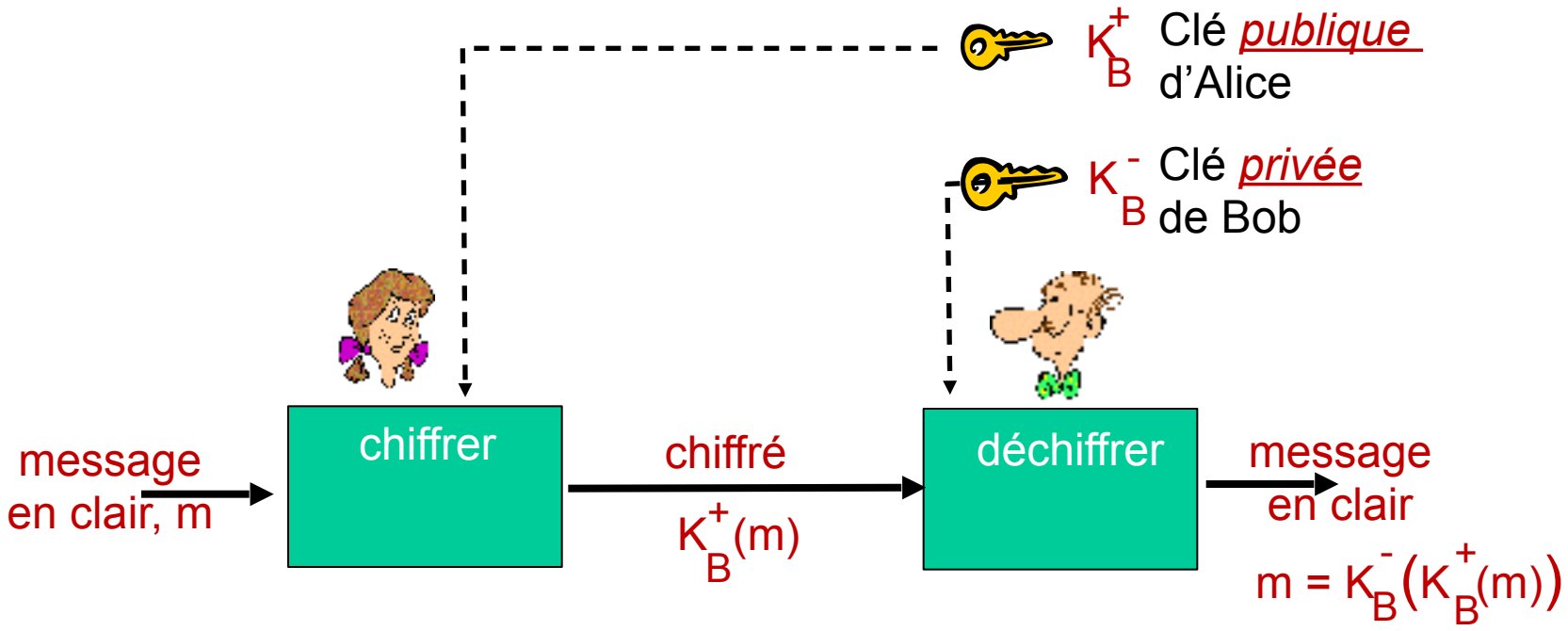
clés symétriques

- Alice et Bob partagent une clé secrète
- Comment obtenir cette clé secrète?

clés publiques

- approche différente [Diffie-Hellman76, RSA78]
- Alice et Bob ne partagent pas de clé secrète
- clé *publique* est connue de tous
- clé *privée* pour déchiffrer n'est connue que de Bob

Cryptographie à clés publiques (Cryptographie asymétrique)



Algorithme de chiffrement

Principe:

1. $K_B^+( )$ et $K_B^-( )$ vérifient

$$K_B^-(K_B^+(m)) = m$$

2. A partir de la clé publique K_B^+, il est « impossible » de calculer la clé privée K_B^-

($K_B^+()$ et $K_B^-()$ sont inverses l'une de l'autre, $K_B^+()$ est facile à calculer mais à partir de $K_B^+()$ il est très difficile de calculer $K_B^-()$ *one-way functions)*

RSA: Rivest, Shamir, Adelson

RSA: création des clés privée/publique

1. choisir deux grands nombres premiers p et q (par exemple 1024 bits)
2. calculer $n = pq$, $z = (p-1)(q-1)$
3. choisir e $(e<n)$ sans diviseur commun avec z (e, z sont premiers entre eux).
4. choisir d tel que $ed-1$ est divisible par z. (ed mod $z = 1$).
5. *clé publique* $\underbrace{(n,e)}_{K_B^+}$. *clé privée* $\underbrace{(n,d)}_{K_B^-}$.

RSA: chiffrement, déchiffrement,

0. soit (n,e) et (n,d) obtenus précédemment

1. message m ($<n$), c le message chiffré:

$$c = m^e \bmod n$$

2. pour déchiffrer c:

$$m = c^d \bmod n$$

On a $m = (\underbrace{m^e \bmod n}_{c})^d \bmod n$

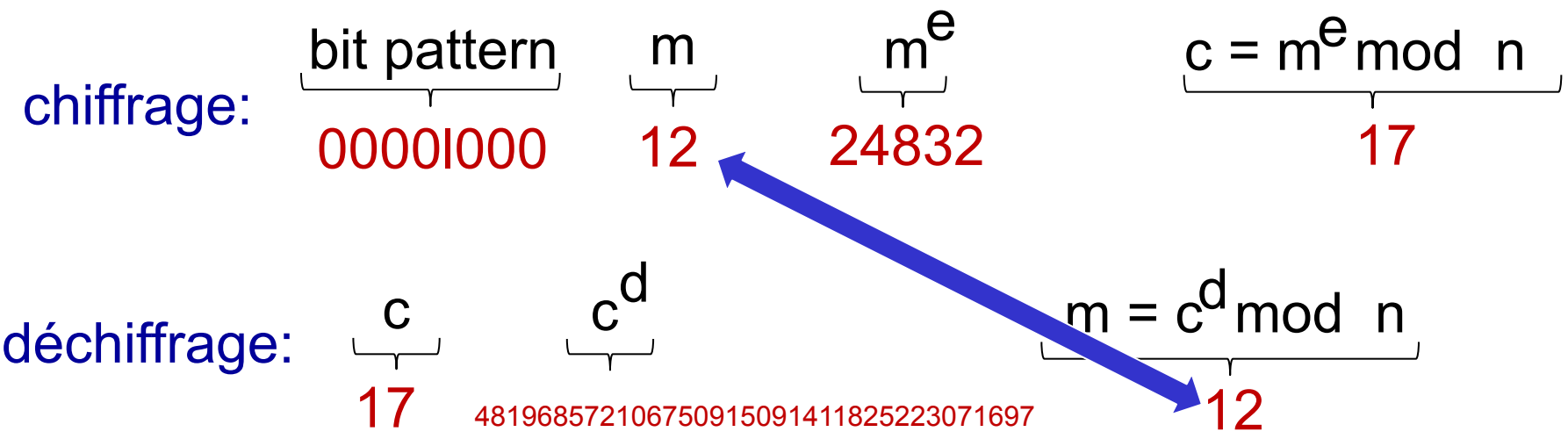
RSA exemple:

Bob choisit *p=5, q=7*. Alors *n=35, z=24*.

e=5 (*e, z* premiers entre eux).

d=29 (*ed-1* divisible par z).

chiffrement message de 8-bits



Chiffrement à clés publiques

RSA vérifie aussi:

$$\underbrace{K_B^-(K_B^+(m))}_{\text{d'abord la clé publique ensuite la clé privée}} = m = \underbrace{K_B^+(K_B^-(m))}_{\text{d'abord la clé privée ensuite la clé publique}}$$

Pourquoi RSA est sûr?

- A partir de la clé publique de Bob (n,e). Il est difficile de trouver d
- « nécessite » de factoriser n sans connaître p et q.
  - On considère que factoriser un grand nombre est difficile.

RSA dans la réalité

- l'exponentiation utilisée dans RSA est coûteuse
- DES est 100 fois plus rapide que RSA
- En pratique:
  - utiliser un système à clés publiques pour établir une communication sûre et s'entendre sur une clé symétrique, utiliser cette clé pour chiffrer-déchiffrer les communications.

Clé de session, K_S

- Bob et Alice utilisent RSA pour échanger une clé symétrique K_S
- avec K_S, codage symétrique

Signatures numériques

technique analogue à la signature manuelle:

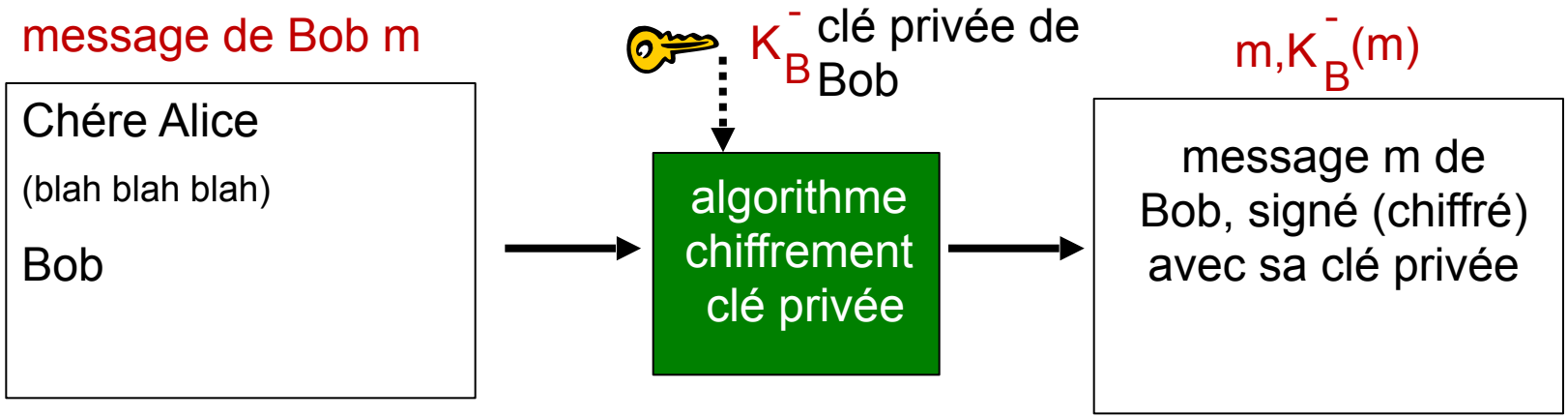
- Bob signe numériquement le document, établissant ainsi qu'il est le créateur/propriétaire du document.
- *vérifiable, infalsifiable (unforgeable):* Alice peut prouver à un tiers que personne d'autres que Bob n'a signé le document

- attention… on suppose ici que « tout le monde » sait que la clé publique de Bob est bien la clé publique de Bob et seul Bob connaît sa clé privée

Signatures numériques

simple signature numérique pour le message m:

- Bob signe m en le codant avec sa clé privée K_B^-, créant le message signé , $K_B^-(m)$



Signatures numériques

- si Alice reçoit m, avec la signature: m, $K_B^-(m)$
- Alice vérifie que m est signé par Bob avec la clé publique de Bob K_B^+ : $K_B^+(K_B^-(m)) = m$.
- Si $K_B^+(K_B^-(m)) = m$, celui qui a signé avait la clé privée de Bob

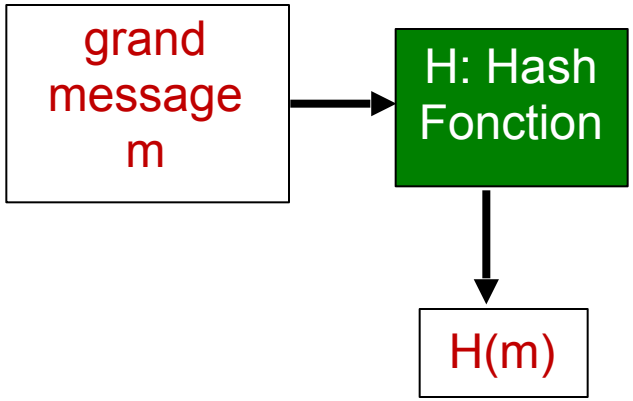
Alice vérifie:

- Bob a signé m
- personne d'autre n'a signé m
- Bob a signé m et pas m'

non-répudiation:

- ✓ Alice peut aller en justice prendre m, et la signature $K_B^-(m)$ et prouver que Bob a signé m

« Message digest »



le chiffrement de longs messages avec clé publique est très coûteux

Mais: mais on peut facilement chiffrer des empreintes (digest) de taille fixe ("fingerprint")

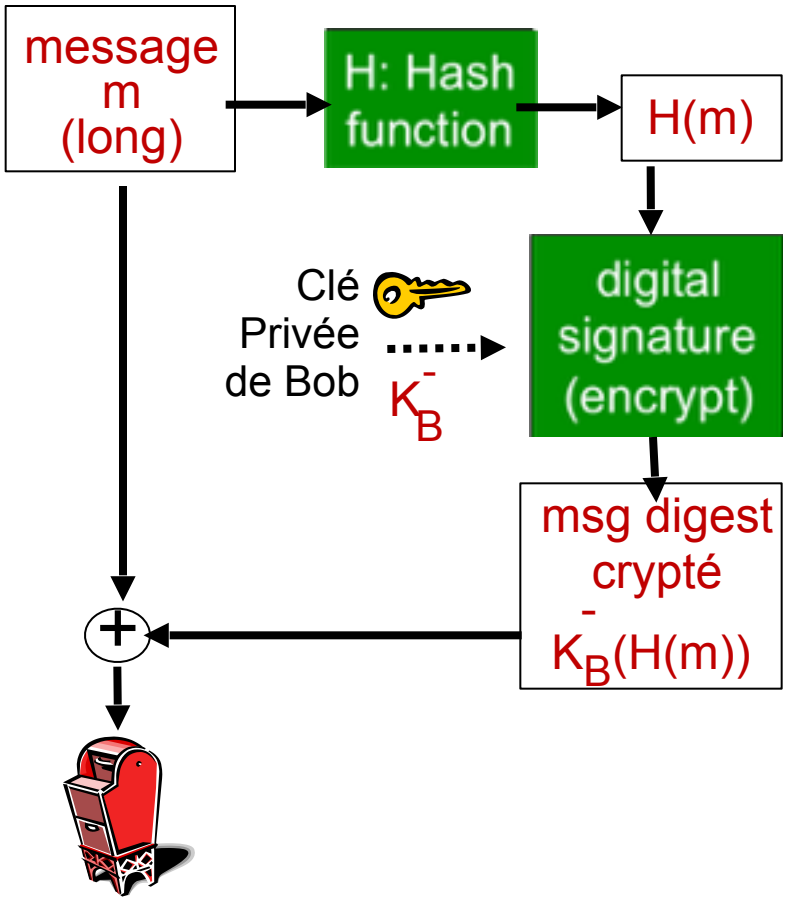
- en appliquent H fonction de hachage à m, on obtient un digest *H(m)*.

Propriétés des fonctions de hachage:

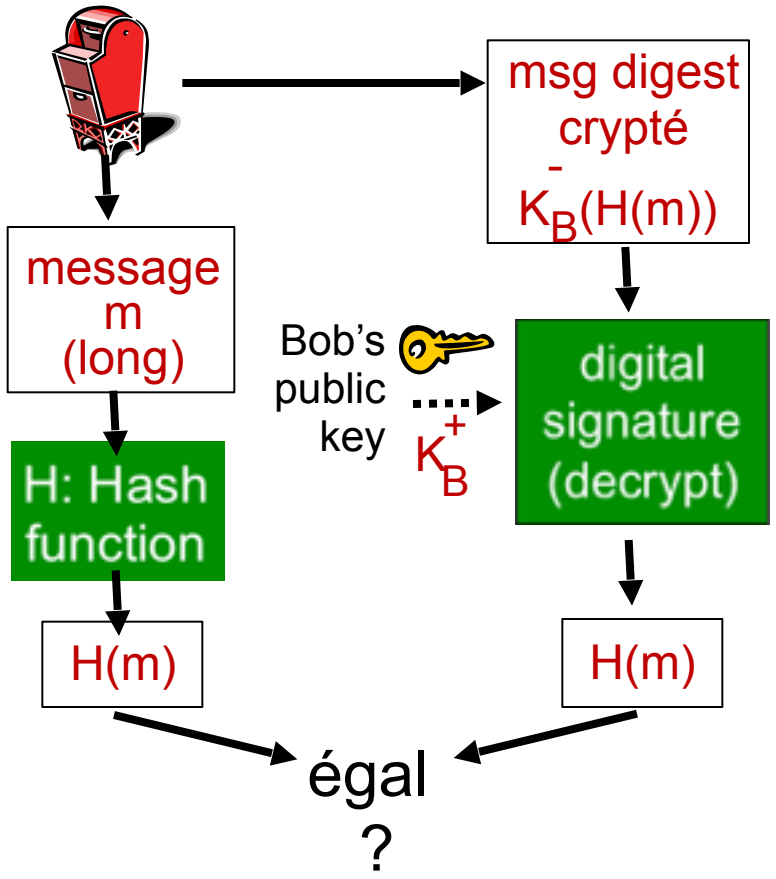
- E=H(M): « impossible » de trouver M connaissant E
- connaissant M et E « impossible » de trouver M' tel que H(M')=H(M)=E
- « impossible » de trouver M et M' tels que H(M)=H(M')

Digest signé comme signature numérique

Bob envoie le message signé :



Alice vérifie la signature,
l'intégrité du message signé:

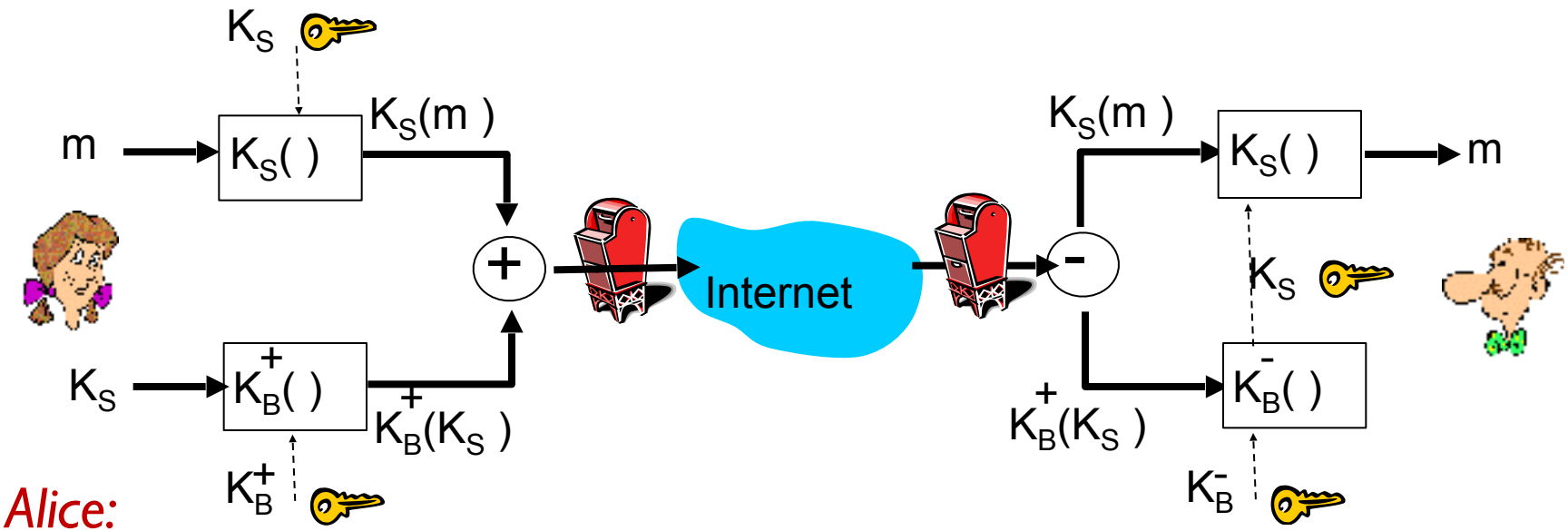


algorithmes de Hachage

- MD5 (RFC 1321)
  - calcule un « digest » de128-bit
  - à partir d'une chaîne x de128-bits, il est difficile de construire un msg m pour lequel le hachage par MD5 est égal à x
  - « cassé » en 2004
- SHA-256 SHA-512

e-mail sécurisé

- Alice veut envoyer un mail confidentiel à Bob

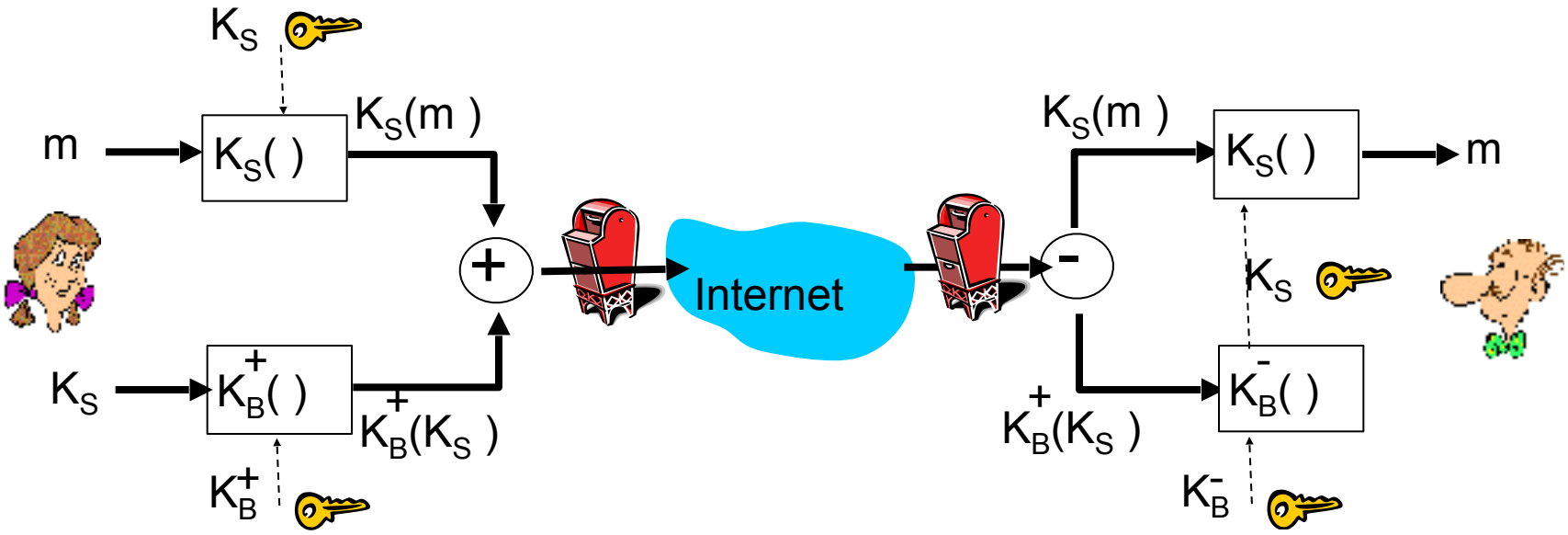


Alice:

- génère une clé *symétrique* privée, K_S
- chiffre le message avec K_S (pour l'efficacité)
- chiffre aussi K_S avec la clé publique de Bob
- envoie $K_S(m)$ et $K_B^+(K_S)$ to Bob

e-mail sécurisé

- Alice veut envoyer un mail confidentiel m à Bob.

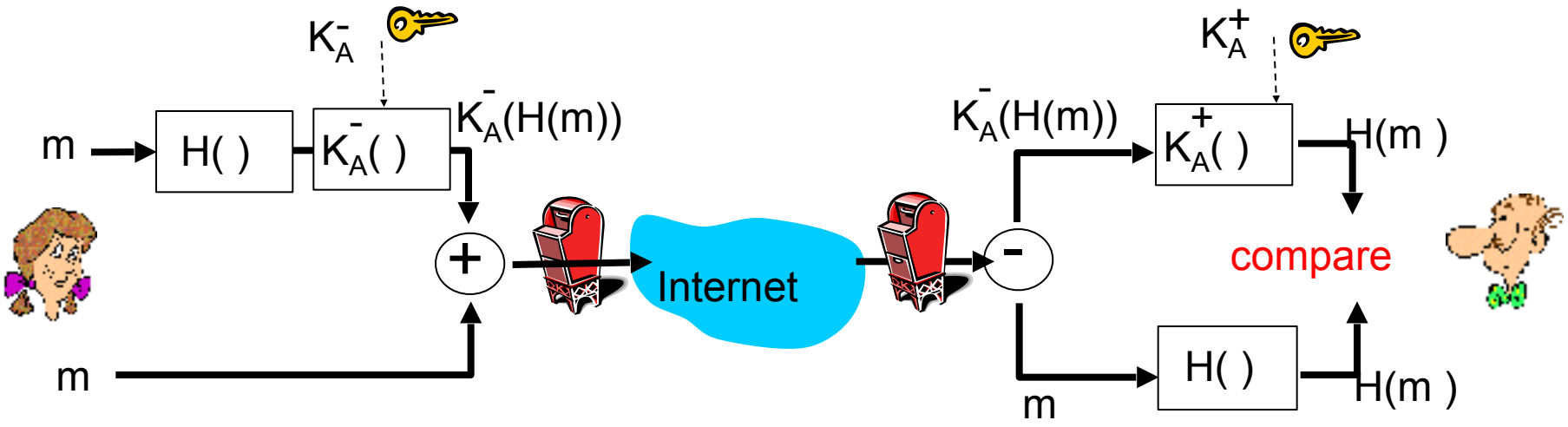


Bob:

- utilise sa clé privée pour déchiffrer et obtient K_S
- utilise K_S pour déchiffrer $K_S(m)$ pour obtenir m

e-mail sécurisé (suite)

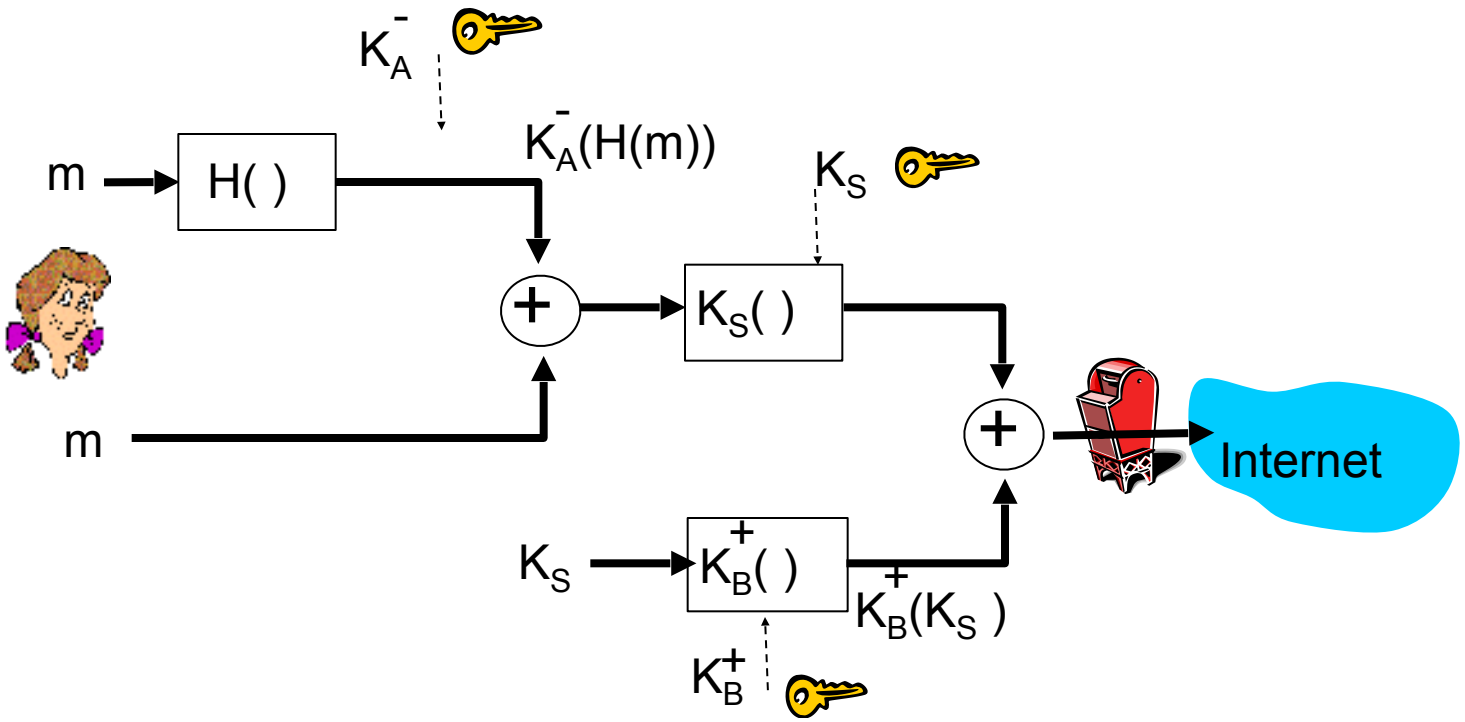
- Alice veut en plus prouver l'intégrité et authentifier le message



- Alice signe le message (signature numérique)
- envoie à la fois le message en clair et la signature numérique

e-mail sécurisé (suite)

- Alice veut tout: secret authentification de l'émetteur et intégrité



Alice utilise 3 clés: sa clé privée, la clé publique de Bob, et une nouvelle clé symétrique