Cryptographie assymétrique

IF202 - partie 2

2020-2021

corentin travers

Cryptographie symétrique

- **Enc, Dec :** algorithmes de chiffrement/déchiffrement
- **k :** clé **secrète partagée** par Alice et Bob

Clés secrètes ?

- **Partage d'une clé entre un client et un serveur ?**
  - **canal fiable ?**

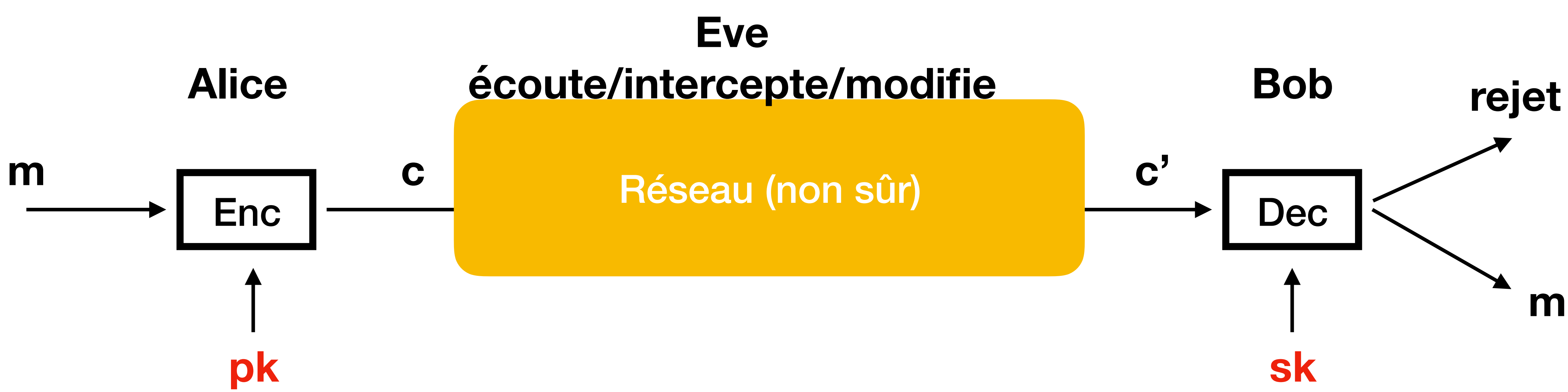
Gestion des clés ?

- N participants, canaux sûrs point-à-point : $O(N^2)$ clés, $O(N)$ clés par participants

- Distribution, partage des clés entre paires de participants ?

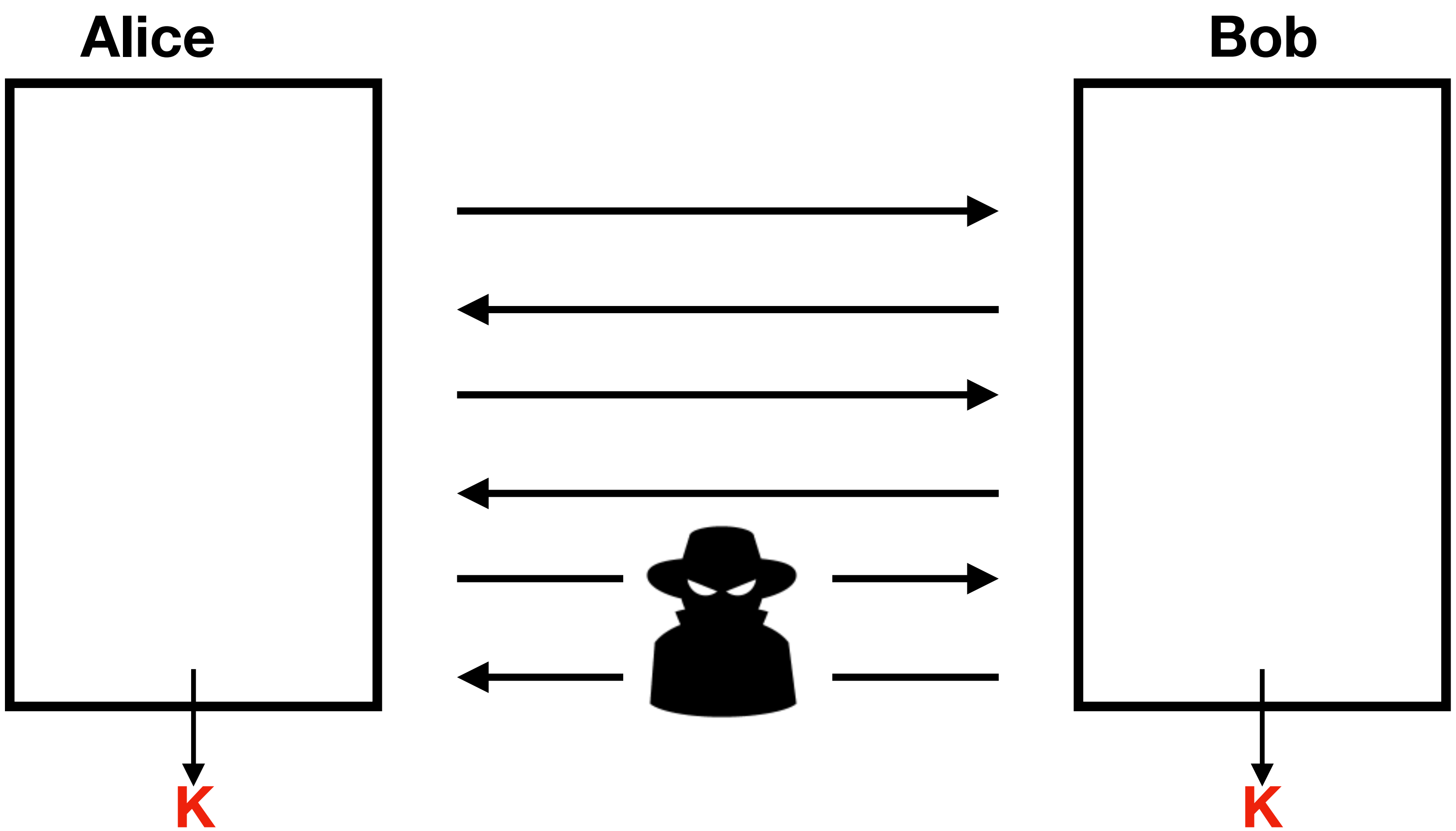
- Stockage des clés et utilisation des clés ? Que faire si une clé est compromise ?

Cryptographie assymétrique



- **Enc, Dec :** algorithmes de chiffrement/déchiffrement
- **k =(pk,sk)** clé de **Bob**
    - **sk clé secrète : connue uniquement de Bob**
    - **pk clé publique : connue de tous (y compris l'attaquant)**

Protocole d'échange de clés

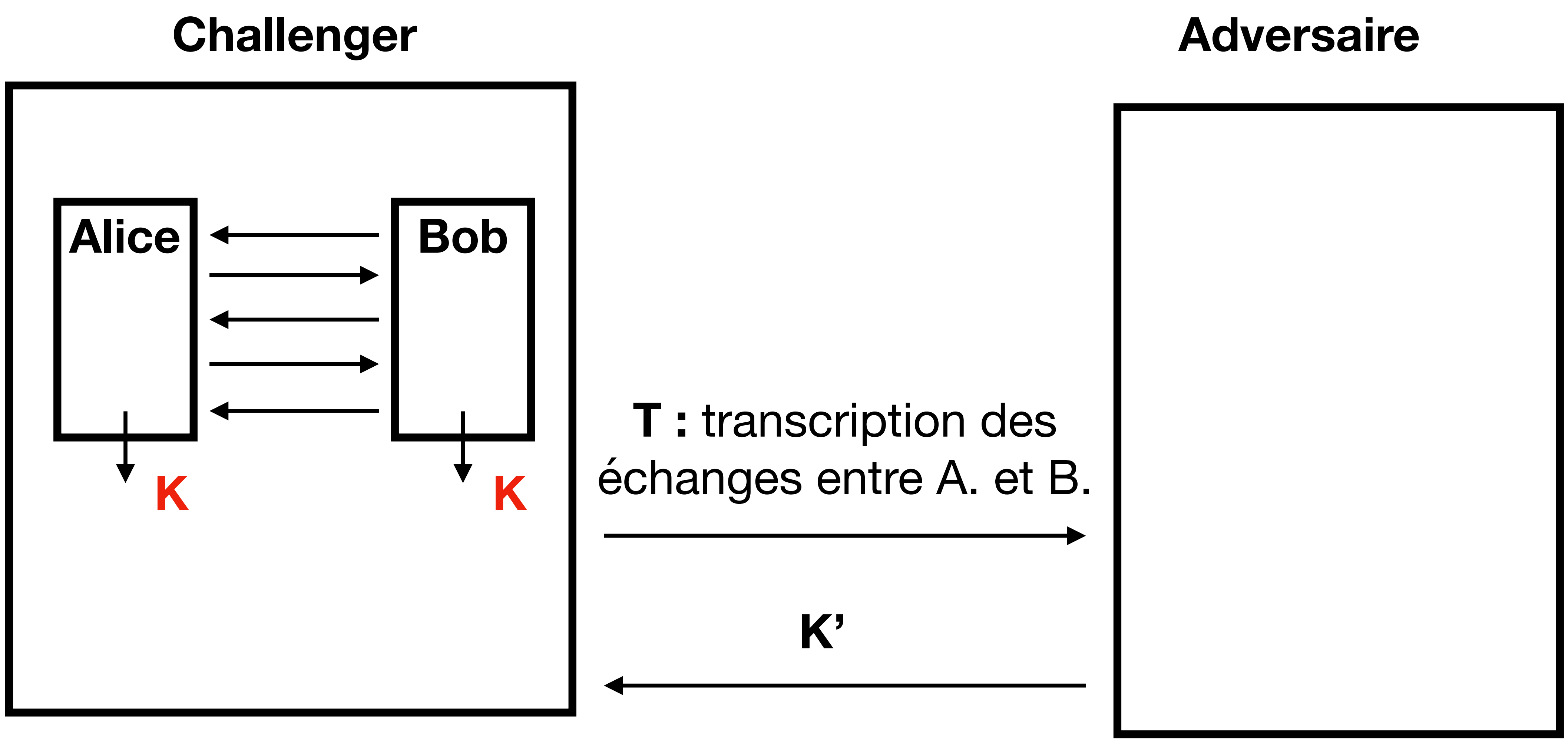


- protocole entre Alice et Bob
- pas de secret initialement partagé entre Alice et Bob
- attaquant écoute, voire modifie les messages échangés

But : établissement d'une clé secrète K entre Alice et Bob

- attaquant ne peut retrouver K en temps raisonnable

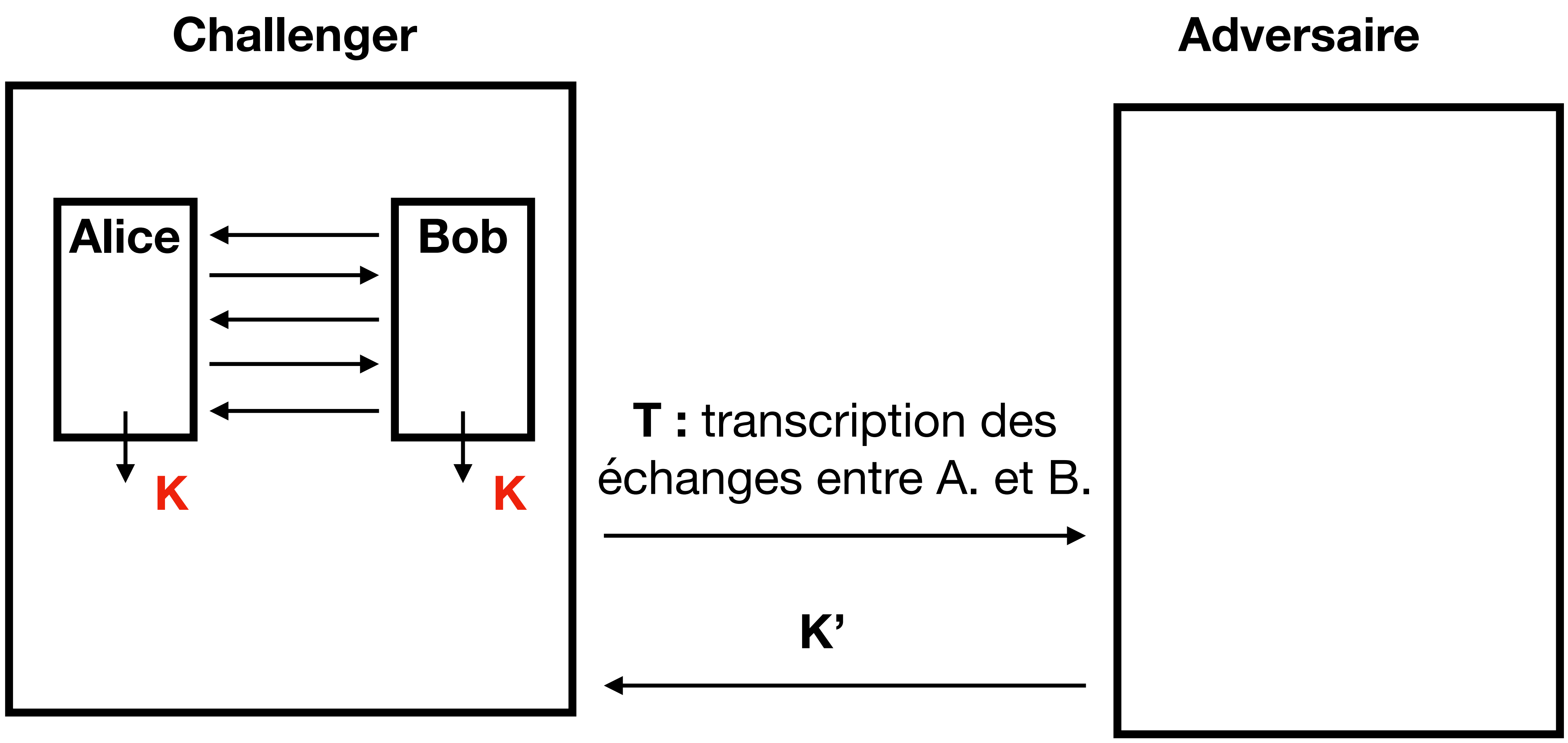
échange de clés : sécurité



W : évènement **K' = K**

Protocole sûr : pour tout adversaire efficace, $\Pr[W] = \text{negl}$.

échange de clés : sécurité



W : évènement **K' = K**

Protocole sûr : pour tout adversaire efficace, $\Pr[W] = \text{negl}$.

limites :
- **adversaire** doit retrouver l'intégralité de la clé **K**
- **adversaire** ne peut modifier/intercepter/injecter messages du protocole

échange de clés : Puzzles de Merkle (1974)

Soit $S = (\mathrm{Enc}, \mathrm{Dec}, K, M, C)$ **un cryptosystème à clés secrétes (ie, AES, DES, etc.)**

Supp. $K = \{0{,}1\}^N$ **clé de N bits** **Supp.** $M = \{0{,}1\}^L, L \geq N$

puzzle $(\mathrm{Enc}(k, id), \mathrm{Enc}(k, SK), \mathrm{Enc}(k,0))$

clé $k \in K$

id du puzzle

solution du puzzle

résoudre le puzzle $P = (c_1, c_2, c_3)$ trouver k **tel que** $\mathrm{Dec}(k, c_3) = 0$ complexité $\mathbf{O(N)}$

échange de clés : Puzzles de Merkle (1974)

Alice | **Bob**

1. générer N puzzles :

$KT[1..N] \leftarrow [\ \bot\ , \ldots, \ \bot\ ]$

pour $i \in \{1,\ldots,N\}$ **faire**

$k_i \leftarrow^r \{0,1\}^N$

$SK_i \leftarrow^r \{0,1\}^L$

$KT[i] \leftarrow SK_i$

$P_i \leftarrow (\mathrm{Enc}(k_i, i), \mathrm{Enc}(k_i, SK_i), \mathrm{Enc}(k_i, 0))$

échange de clés : Puzzles de Merkle (1974)

Alice **Bob**

1./<u>génère N puzzles :</u>

$KT[1..N] \leftarrow [\ \perp, \ldots, \perp\ ]$

pour $i \in \{1,\ldots,N\}$ **faire**

$k_i \leftarrow^r \{0,1\}^N$

$SK_i \leftarrow^r \{0,1\}^L$

$KT[i] \leftarrow SK_i$

$P_i \leftarrow (\mathrm{Enc}(k_i, i), \mathrm{Enc}(k_i, SK_i), \mathrm{Enc}(k_i, 0))$

2./ <u>envoi des N puzzles</u> dans un ordre aléatoire

$$\xrightarrow[P_{\pi(1)}, \ldots, P_{\pi(N)}]{}$$

échange de clés : Puzzles de Merkle (1974)

Alice **Bob**

$$\xrightarrow[P_{\pi(1)}, \ldots, P_{\pi(N)}]{}$$

3./ Résout l'un des puzzles $P_j = (c_1, c_2, c_3)$

$id \leftarrow \mathrm{Dec}(k_j, c_1); SK \leftarrow \mathrm{Dec}(k_j, c_2);$

$0 \leftarrow \mathrm{Dec}(k_j, c_3)$

4./ Envoi identifiant id de P_j

$$\xleftarrow[id]{}$$

5./ Produit **K** = $KT[id]$

5'./ Produit **K** = SK

échange de clés : Puzzles de Merkle (1974)

Alice **Bob**

1./génère N puzzles :

2./ envoi des N puzzles dans un ordre aléatoire

$$\xrightarrow[P_{\pi(1)}, \ldots, P_{\pi(N)}]{}$$

3./ Résout l'un des puzzles $P_j = (c_1, c_2, c_3)$

4./ Envoi identifiant id de P_j

$$\xleftarrow[id]{}$$

5./ Produit **K** clé associée à l'identifiant id

5'./ Produit **K** clé **SK** du puzzle P_j

échange de clés : Puzzles de Merkle (1974)

Alice **Bob**

1./génère N puzzles :

2./ envoi des N puzzles dans un ordre aléatoire

$$\xrightarrow[P_{\pi(1)}, \ldots, P_{\pi(N)}]{}$$

3./ Résout l'un des puzzles $P_j = (c_1, c_2, c_3)$

4./ Envoi identifiant id de P_j

$$\xleftarrow[id]{}$$

5./ Produit **K** clé associée à l'identifiant id

5'./ Produit **K** clé **SK** du puzzle P_j

Complexité ?

échange de clés : Puzzles de Merkle (1974)

Alice **Bob**

1./génère N puzzles :

2./ envoi des N puzzles dans un ordre aléatoire

$$\xrightarrow[P_{\pi(1)}, \ldots, P_{\pi(N)}]{}$$

3./ Résout l'un des puzzles $P_j = (c_1, c_2, c_3)$

4./ Envoi identifiant id de P_j

$$\xleftarrow[id]{}$$

5./ Produit **K** clé associée à l'identifiant id

5'./ Produit **K** clé **SK** du puzzle P_j

Complexité ?

- **Alice : génération et envoi des puzzles O(N)**
- **Bob : résolution d'un puzzle O(N)**

échange de clés : Puzzles de Merkle (1974)

Attaquant :

- **Pas de modification des messages**
- **écoute uniquement des messages entre Alice et Bob**
- **But retrouver la clé partagée K**

- **Entrée** $(P_1, \ldots, P_N), id$
- **Sortie K**
- **Algorithme : résoudre les puzzles** $P_1, P_2, \ldots, P_N$

jusqu'à trouver celui dont l'identifiant $= id$

Complexité ?

échange de clés : Puzzles de Merkle (1974)

Attaquant :

- **Pas de modification des messages**
- **écoute uniquement des messages entre Alice et Bob**
- **But retrouver la clé partagée K**

- **Entrée** $(P_1, \ldots, P_N), id$
- **Sortie K**
- **Algorithme : résoudre les puzzles** $P_1, P_2, \ldots, P_N$

jusqu'à trouver celui dont l'identifiant $= id$

Complexité ?

- **Attaquant : résolution des** N **puzzles O(**N^2**)**

Puzzles de Merkle (1974)

Complexité ?

- **Alice : génération et envoi des puzzles O(N)**
- **Bob : résolution d'un puzzle O(N)**

- **Attaquant : résolution des N puzzles O(N^2)**

En pratique, non utilisable : N doit être choisit grand pour rendre l'attaque infaisable
Cependant, **saut quadratique** de complexité entre utilisateurs légitimes et attaquant

Puzzles de Merkle (1974)

Complexité ?

- **Alice : génération et envoi des puzzles O(N)**
- **Bob : résolution d'un puzzle O(N)**

- **Attaquant : résolution des N puzzles O(N^2)**

En pratique, non utilisable : N doit être choisit grand pour rendre l'attaque infaisable
Cependant, **saut quadratique** de complexité entre utilisateurs légitimes et attaquant

Souhait protocole d'échange de clés tel que :

- **Alice, Bob** $O(N^d)$
- **Attaquant** $O(\text{super} - \text{poly}(N))$

Protocole d'échange de clés de Diffie Hellman

- Soit $(\mathbb{G}, \cdot\ )$ un groupe cyclique de générateur g
- typiquement $\cdot = \times \mod p$ avec p (grand) nombre premier
- $\mathbb{G} = \{1, g, g^2, \ldots, g^{q-1}\}$ où g d'ordre q avec $q \,|\, p-1$

Protocole d'échange de clés de Diffie Hellman

paramètres publics : p, g, q

Alice		**Bob**
$\alpha \xleftarrow{r} \{0,\ldots,q-1\}$		$\beta \xleftarrow{r} \{0,\ldots,q-1\}$
$u \leftarrow g^{\alpha} \mod p$		$v \leftarrow g^{\beta} \mod p$
	$\xrightarrow{\hspace{2em}}$ $u = g^{\alpha} \mod p$	
	$\xleftarrow{\hspace{2em}}$ $v = g^{\beta} \mod p$	
$K \leftarrow v^{\beta} \mod p$		$K \leftarrow u^{\alpha} \mod p$

Protocole d'échange de clés de Diffie Hellman

paramètres publics : p, g, q

Alice

$\alpha \xleftarrow{r} \{0, \ldots, q-1\}$

$u \leftarrow g^{\alpha} \mod p$

$\xrightarrow{\hspace{3em}}$

$u = g^{\alpha} \mod p$

$\xleftarrow{\hspace{3em}}$

$v = g^{\beta} \mod p$

$K \leftarrow v^{\beta} \mod p$

$\downarrow$

K = $(g^{\beta})^{\alpha} = g^{\alpha\beta} \mod p$

Bob

$\beta \xleftarrow{r} \{0, \ldots, q-1\}$

$v \leftarrow g^{\beta} \mod p$

$K \leftarrow u^{\alpha} \mod p$

$\downarrow$

K = $(g^{\beta})^{\alpha} = g^{\alpha\beta} \mod p$

Protocole d'échange de clés de Diffie Hellman

Alice **paramètres publics :** p, g, q **Bob**

$\alpha \leftarrow^{r} \{0,\ldots,q-1\}$

$u \leftarrow g^{\alpha} \mod p$

$\beta \leftarrow^{r} \{0,\ldots,q-1\}$

$v \leftarrow g^{\beta} \mod p$

$$\xrightarrow{}$$

$$u = g^{\alpha} \mod p$$

$$\xleftarrow{}$$

$$v = g^{\beta} \mod p$$

$K \leftarrow v^{\beta} \mod p$

$K \leftarrow u^{\alpha} \mod p$

Complexité ? **Alice, Bob :** ***exponentiation modulaire*** $O(\log q)$

Attaquant ?

Protocole d’échange de clés de Diffie Hellman

Attaquant :

- **écoute des messages échangés entre Alice et Bob**
- **pas de modification/interception/injection**

Entrée :

- **Paramètres publics** p, q, g
- **retranscription des échanges** $u = g^{\alpha} \mod p, v = g^{\beta} \mod p$

Sortie :

- $K = g^{\alpha\beta} \mod p$

Logarithme discret

- $\mathbb{G}$ groupe fini de générateur g ($\mathbb{G} = \{g, g^2, \ldots, g^{|\mathbb{G}|}\}$)

- $\text{DLOG}_g : \mathbb{G} \to \{0, |\mathbb{G}| - 1\}$

- $\text{DLOG}_g(u) = \alpha$ t.q. $g^\alpha = u$

Groupe cyclique :

- $(\{1,\ldots,p-1\}, \times \mod p)$ avec p premier
- groupe défini à partir de courbes elliptiques

Problème Diffie Hellman version calcul

- $\mathbb{G}$ groupe fini de générateur g ($\mathbb{G} = \{g, g^2, \ldots, g^{|\mathbb{G}|}\}$)
- Diffie-Hellman calculatoire :
  - $\mathrm{CDH}_g : \mathbb{G} \times \mathbb{G} \to \mathbb{G}$
  - $\mathrm{CDH}_g(x = g^\alpha, y = g^\beta) \to z = g^{\alpha\beta}$

Problème Diffie Hellman version décision

- $\mathbb{G}$ groupe fini de générateur g ($\mathbb{G} = \{g, g^2, \ldots, g^{|\mathbb{G}|}\}$)
- Diffie-Hellman décisionnel :
  - $\mathrm{DDH}_g : \mathbb{G} \times \mathbb{G} \times \mathbb{G} \to \{0,1\}$
  - $\mathrm{CDH}_g(x = g^\alpha, y = g^\beta, z = g^\gamma) = 1$ si $g^\gamma = g^{\alpha\beta}$,$0$ sinon

Logarithme discrets, problèmes de Diffie-Hellman

$\mathbb{G}$ groupe fini de générateur g ($\mathbb{G} = \{g, g^2, \ldots, g^{|\mathbb{G}|}\}$

$\boxed{\mathsf{DDH}_g \leq_P \mathsf{CDH}_g \leq_P \mathsf{DLOG}_g}$

DDH **est** **<u>supposé</u>** **difficile dans**

- **certains sous-groupes de** $\mathbb{Z}_p^* = (\{1,\ldots,p-1\}, \times \mod p)$ **avec** p **premier**
- **des groupes construits à partir de courbes elliptiques**

Si DDH_g **est difficile dans** $\mathbb{G}$ **alors le protocole de Diffie-Hellman avec paramètres publics** $(\mathbb{G}, g, |\mathbb{G}|)$ **est sûr (pour les attaques à écoute seule)**

Fonctions à *sens unique* et à *brèche secrète*

one-way trapdoor functions

Intuitivement, $F_{pk,sk} : X \to Y$ est à <u>sens unique et à brèche secrète</u> si

- $F_{pk,sk} : X \to Y$ **facile** à calculer
- $F^{-1}_{pk,sk} : Y \to X$ **difficile** à calculer
- mais il existe un **algorithme efficace** I qui, étant donné $y = F_{pk,sk}(x)$ et sk produit x

- **facile** = existence d'un algorithme de complexité polynomiale
- **difficile** = tout alg. de cplxité pol. a une probabilité de succès negl .
- **efficace** = de complexité polynomiale
- complexité mesurée en fonction d'un **paramètre de sécurité** n
  (e.g. la taille en bits de pk, sk)

Fonctions à *sens unique* et à *brèche secrète*

one-way trapdoor functions

définition

Un triplet de 3 algorithmes (G, F, I) fournit une famille de fonctions à sens unique et à brèche secrète sur X, Y si :

- G est un algorithme probabiliste de génération de clés $G() \to (pk, sk)$

pk, sk sont appelées clés publiques et clés secrètes respectivement

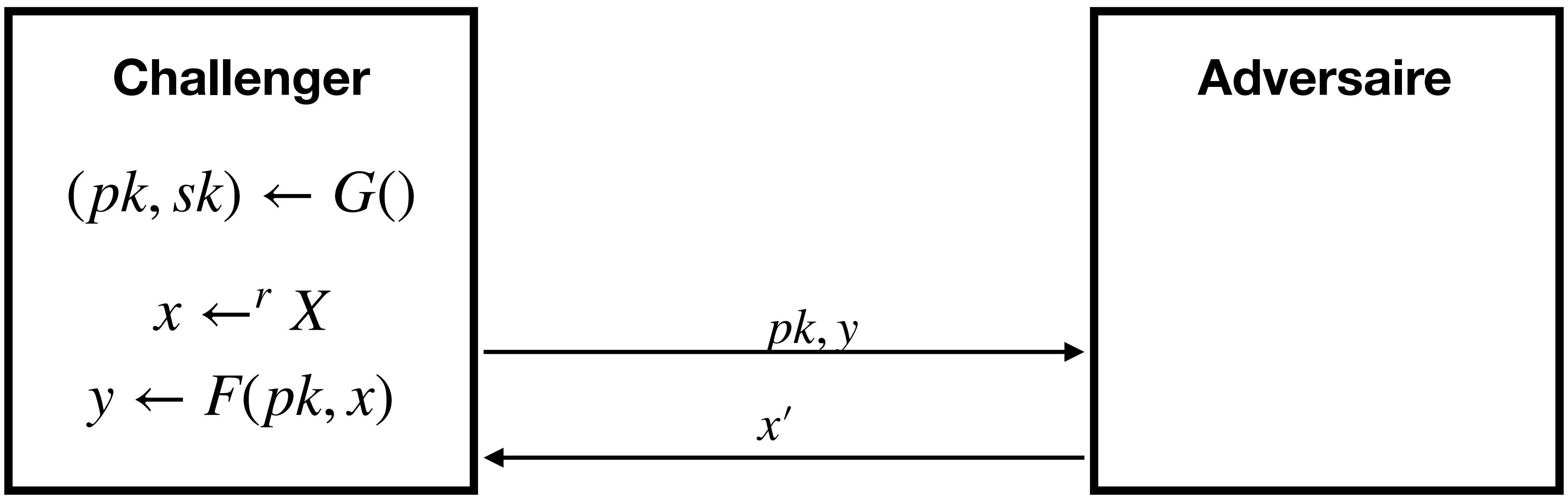
- F est un alg. déterministe efficace $F(pk, x) \to y$ avec pk clé publique $x \in X, y \in Y$
- I est un alg. déterministe efficace $I(sk, y) \to x$ avec sk clé secrète $y \in Y, x \in X$

satisfaisant

- $\forall (pk, sk) \leftarrow G(), \forall x \in X : I(sk, F(pk, x)) = x$

à sens unique : définition

Soit (G, F, I) **défini sur** X, Y **:**



W : Adversaire produit $x' = x$

définition

(G, F, I) **est à sens unique si pour tout adversaire efficace,** $\Pr[W] =$ negl .

Permutation à sens unique et à brèche secrète RSA

Rivest, Shamir, Adleman 1977

GenRSA$(\ell, e : \text{entiers})$ **e impair**

$p, q \leftarrow$ entiers premiers aléatoires de ℓ bits
tels que $pgcd(p-1, e) = pgcd(q-1, e) = 1$

$N \leftarrow pq$

$d \leftarrow e^{-1} \mod (p-1)(q-1)$

$pk \leftarrow (N, e); sk \leftarrow (N, d)$

retourner (pk, sk)

Permutation à sens unique et à brèche secrète RSA

Rivest, Shamir, Adleman 1977

GenRSA$(\ell, e : \text{entiers})$

$p, q \leftarrow$ entiers premiers aléatoires de ℓ bits
tels que $pgcd(p-1,e) = pgcd(q-1,e) = 1$

$N \leftarrow pq$

$d \leftarrow e^{-1} \mod (p-1)(q-1)$

$pk \leftarrow (N, e); sk \leftarrow (N, d)$

retourner (pk, sk)

F$(pk = (N, e), x)$

retourner $x^e \mod N$

I$(sk = (N, e), d)$

retourner $y^d \mod N$

Permutation à sens unique et à brèche secrète RSA

Rivest, Shamir, Adleman 1977

GenRSA$(\ell, e : \text{entiers})$

$p, q \leftarrow$ entiers premiers aléatoires de ℓ bits tels que $pgcd(p-1, e) = pgcd(q-1, e) = 1$

$N \leftarrow pq$

$d \leftarrow e^{-1} \mod (p-1)(q-1)$

$pk \leftarrow (N, e); sk \leftarrow (N, d)$

retourner (pk, sk)

F$(pk = (N, e), x)$

retourner $x^e \mod N$

I$(sk = (N, e), d)$

retourner $y^d \mod N$

Validité : $I((N, d), F((N, e), x)) = (x^e)^d = x^{1+k\varphi(N)} = x \mod N$

Permutation RSA : sécurité

- **Hypothèse** (non prouvée) **la permutation RSA est à sens unique**

- $N = pq$ **module RSA.** Existence d'un alg. efficace de factorisation => attaque de la permutation RSA

- Ordinateurs quantiques ?

$\mathbb{Z}_p$, $\mathbb{Z}_p^*$ quelques rappels d'arithmétique modulaire

Références

- Appendix IV section A dans l'ouvrage de Boneh et Shoup « A Graduate Course in Applied Cryptography »http://toc.cryptobook.us

- Notes en français ici http://ctravers.vvv.enseirb-matmeca.fr/IF202/rappelsarithmetique.pdf

Arithmétique modulaire

- $+, -, \times \mod N$
- pgcd, théorème de Bezout
- Inverse modulaire
- Groupe $\mathbb{Z}_N = (\{0,\ldots,N-1\}, + \mod N)$
- Groupe
  $\mathbb{Z}_N^* = (\{x : 1 \leq x \leq N-1 \wedge pgcd(x,N) = 1\}, \times \mod N)$
- Soit $(\mathbb{G}, \cdot)$ un groupe fini de cardinal m. $\forall g \in \mathbb{G}, g^m = 1_{\mathbb{G}}$.
  Conséquence : théorème d'Euler, petit théorème de Fermat

Arithmétique modulaire

- Fonction indicatrice d'Euler : $\varphi(N) = card(\mathbb{Z}_N^*)$
- Soit p, q premiers $\varphi(p) = p - 1, \varphi(pq) = (p-1)(q-1)$
- Soit $N = pq, pgcd(p, q) = 1$. Soit $\Psi : x \to (x \mod p, x \mod q)$. Ψ est un isomorphisme $\mathbb{Z}_N \to \mathbb{Z}_q \times \mathbb{Z}_q$ et de $\mathbb{Z}_N^* \to \mathbb{Z}_q^* \times \mathbb{Z}_q^*$ (a.k.a. Théorème des restes chinois)
- Si q premier, $\mathbb{Z}_q^*$ est un groupe cyclique