

A3 Cyberdéfense, Cyberlog et Cyberdata

Protocoles cryptographiques

Les enseignants :

- M. AISSANI
- M. ABBACHE

2023 / 2024

À propos du module

Intitulé: Protocoles cryptographiques

Responsables: M. Sofiane AISSANI et M. Bournane ABBACHE

Contacts: sofiane.aissani@univ-ubs.fr, bournane.abbache@univ-ubs.fr

Volume horaire: En fonction de la spécialité.

Pré-requis: Cryptographie symétrique et asymétrique

Evaluation: 50% CC et 50 % Examen

- Source1: Support Pr. ADI, université du Québec
- Source2: Burrows M, Abadi M, Needham R « Une logique d'authentification ». Actes de la Royal Society de Londres, série A. 426 (1871) : 233

PLAN DU COURS

- ① Généralités et rappels
- ② Introduction aux protocoles cryptographiques
- ③ Types des protocoles cryptographiques
- ④ Failles des protocoles cryptographiques
- ⑤ Validation formelle des PC (BAN)
- ⑥ Validation automatique des PC (Forme de projet)



Rappels

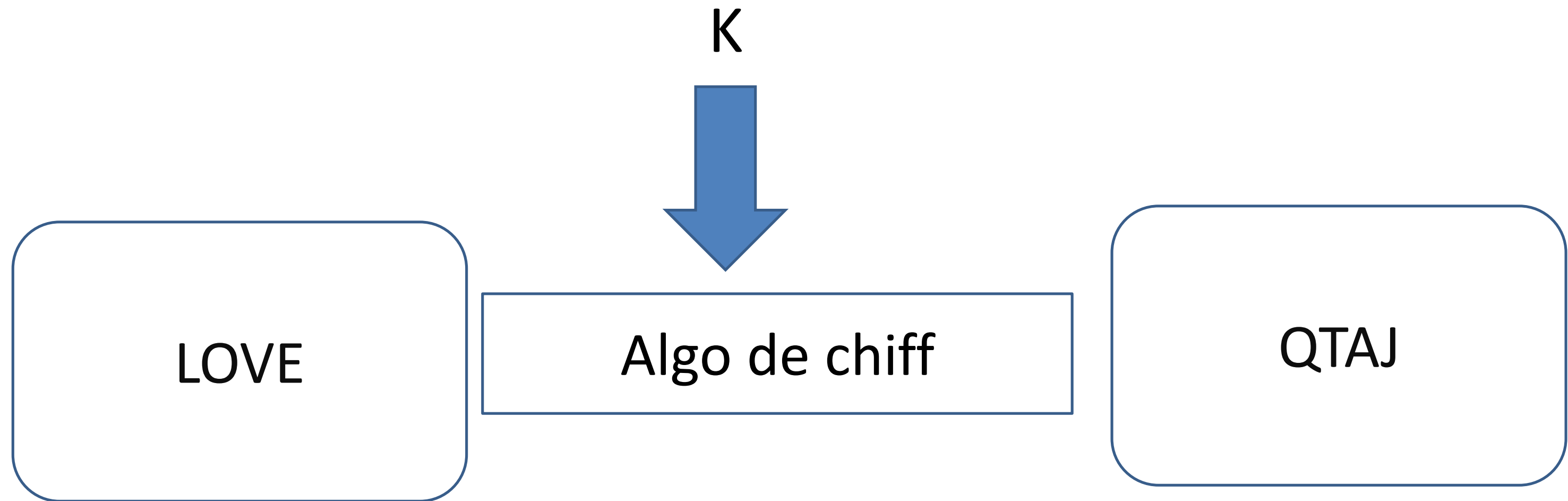
Les services de sécurité: Confidentialité , Authentication, non repudiation, intégrité, etc.

Cryptographie

Clé cryptographique

examples

Ex1



Ex2

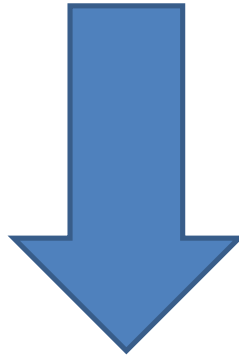
$\forall M_i$

$C_i = 5M_i + 11 \bmod 26$

(5,11)

Exemple crypto Asymétrique

(e, n)



M

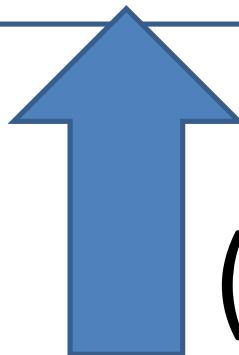
$$C = M^e \bmod n$$

C

M

$$C = M^d \bmod n$$

C



(d, n)

Rappel

- Cryptographie par bloc (AES, 3DES)
- Cryptographie par flux (RC4 et Salsa20)
- Cryptographie à seuil (ECC, Krawczyk)
- Cryptographie quantique (E91, BB84)

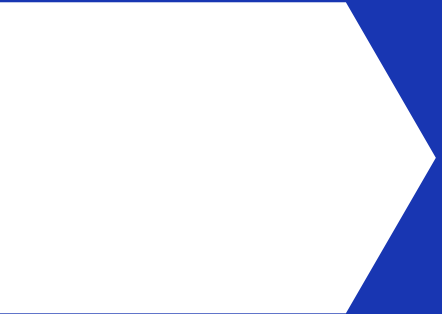
Fonction de hachage

- Taille fixe
- Déterministe
- Calcule rapide
- Irréversible
- Resistance aux collisions
- Effet de diffusion
- Résistance a la préimage


Que signifie la notion signature numérique?
À Quoi sert-elle?

Exemples de fonction de hachage: MD5, SHA1, SHA2,
SHA3

Exercice: Comment assurer les différents services de sécurité?



I. Introduction aux protocoles cryptographiques



Un protocole: Une séquence d'étapes de communication et de calcul.

Un protocole cryptographique: C'est un protocole qui se base sur la cryptographie pour assurer certains objectifs de sécurité.

Introduction (Notations)

Messages :

Identité d'un principal : A, B, C , etc.

Identité d'un serveur : S .

Nonces : Na, Nb , etc.

Message m crypté avec une clé k : $\{m\}_k$.

Message composé : m, m' .

Étapes de communication :

$i. A \rightarrow B : m$

Introduction (Notations)

Exemple :

1. $A \longrightarrow B : N_a$
2. $B \longrightarrow A : \{N_a\}k_b^{-1}$

Rôle vs. principal :

Principal : c'est un agent qui participe dans une session de l'exécution du protocole.

Rôle : c'est une abstraction du protocole où l'emphasis est mise sur un seul agent.

Introduction (Fraicheur)

Nonce, timestamp (estampille) : permet de vérifier la fraîcheur d'un message.

Exemple de message sans fraîcheur :

1. $A \rightarrow B : \textit{es-tu là}$
2. $B \rightarrow A : \{ \textit{oui, je suis là} \} k_b^{-1}$

Introduction (Fraicheur)

Exemple de message en vérifiant la fraîcheur :

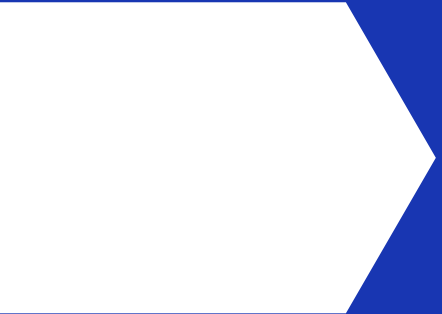
1. $A \rightarrow B : es-tu là, N_a$
2. $B \rightarrow A : \{ oui, je suis là, N_a \} k_b^{-1}$

1. $A \rightarrow B : es-tu là$
2. $B \rightarrow A : \{ oui, je suis là, T_b \} k_b^{-1}$


Introduction (Classification)

Les protocoles cryptographiques peuvent être classés selon plusieurs critères:

- Systèmes cryptographiques;
- Objectif du protocole;
- Nombre d'étapes;
- Utilisation ou non d'un serveur;
- Etc.



Types des protocoles cryptographiques



Protocoles d'authentification

Authentification de l'identité : Permettre à un principal de prouver son identité à un autre.

Sécurité : Ne pas permettre à un principal X *de prouver que son identité est Y* (alors que $X \neq Y$).

Authentification de messages : S'assurer que le contenu d'un message n'a pas été modifié.

Sécurité : Si un message est malicieusement ou accidentellement modifié en cours de route alors le récepteur devrait être capable de détecter cette modification.

Protocoles d'authentification

Approches d'authentification :

Quelque chose qui est en toi: Iris, empreintes digitales, ADN, etc.

Quelque chose que tu possèdes : Clé physique, carte d'accès, etc.

Quelque chose que tu connais : mot de passe, clé secrète, etc.

Protocoles d'authentification

Protocole de Woo et Lam (Exemple I) :

C'est une authentification unidirectionnelle. Seul le principal A qui a besoin de prouver son identité au principal B.

1. $A \longrightarrow B : A$
2. $B \longrightarrow A : N_b$
3. $A \longrightarrow B : \{N_b\}_{kas}$
4. $B \longrightarrow S : \{A, \{N_b\}_{kas}\}_{k_{bs}}$
5. $S \longrightarrow B : \{N_b\}_{k_{bs}}$

Protocoles d'authentification

Protocole de Woo et Lam (Exemple II) :

C'est une authentification bidirectionnelle.

1. $A \rightarrow B : A, N_a$
2. $B \rightarrow A : B, N_b$
3. $A \rightarrow B : \{A, B, N_a, N_b\}_{kas}$
4. $B \rightarrow S : \{A, B, N_a, N_b\}_{kas}, \{A, B, N_a, N_b\}_{kbs}$
5. $S \rightarrow B : \{B, N_a, N_b\}_{kas}, \{A, N_a, N_b\}_{kbs}$
6. $B \rightarrow A : \{B, N_a, N_b\}_{kas}$

Protocole de Needham et Shroeder (Exemple III) :

C'est une authentification bidirectionnelle.
Il se base sur des clés asymétriques.

1. $A \longrightarrow B : \{N_a, A\}_{k_b}$
2. $B \longrightarrow A : \{N_a, N_b\}_{k_a}$
3. $A \longrightarrow B : \{N_b\}_{k_b}$

Protocoles de distribution de clés

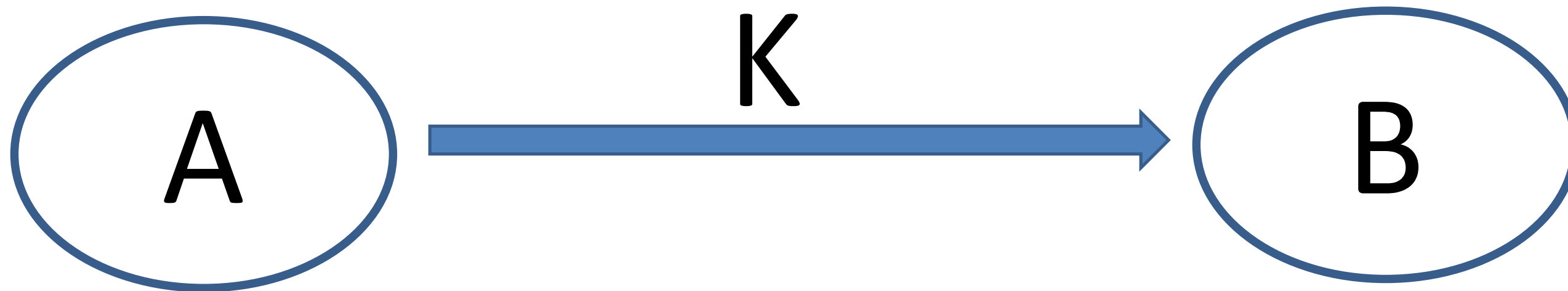
Objectif : distribuer de nouvelles clés aux principaux pour qu'ils s'en servent pendant leurs communications futures. (Inclus dans les protocoles de gestion de clés).

Sécurité :

Confidentialité

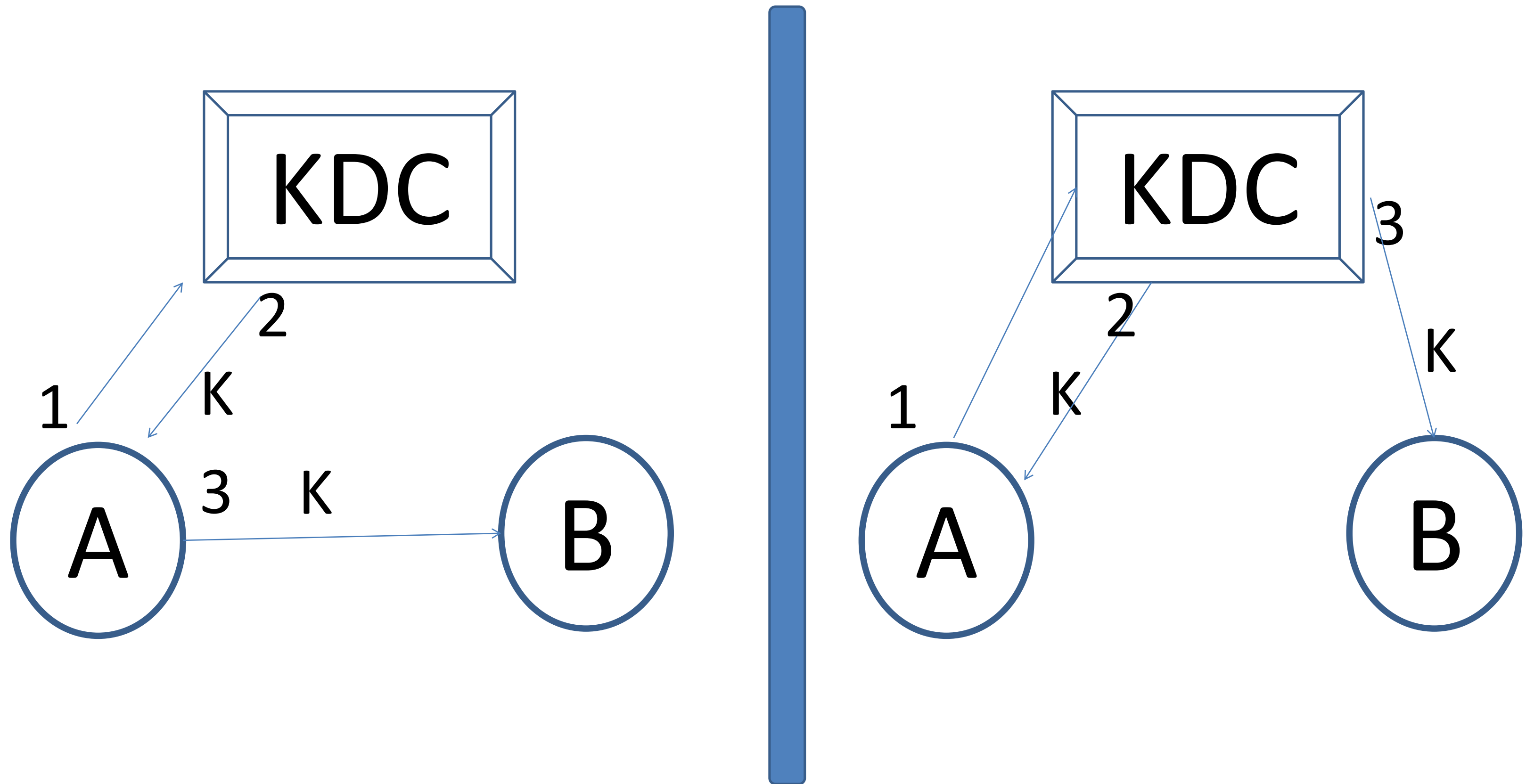
Intégrité

Possibilité 1



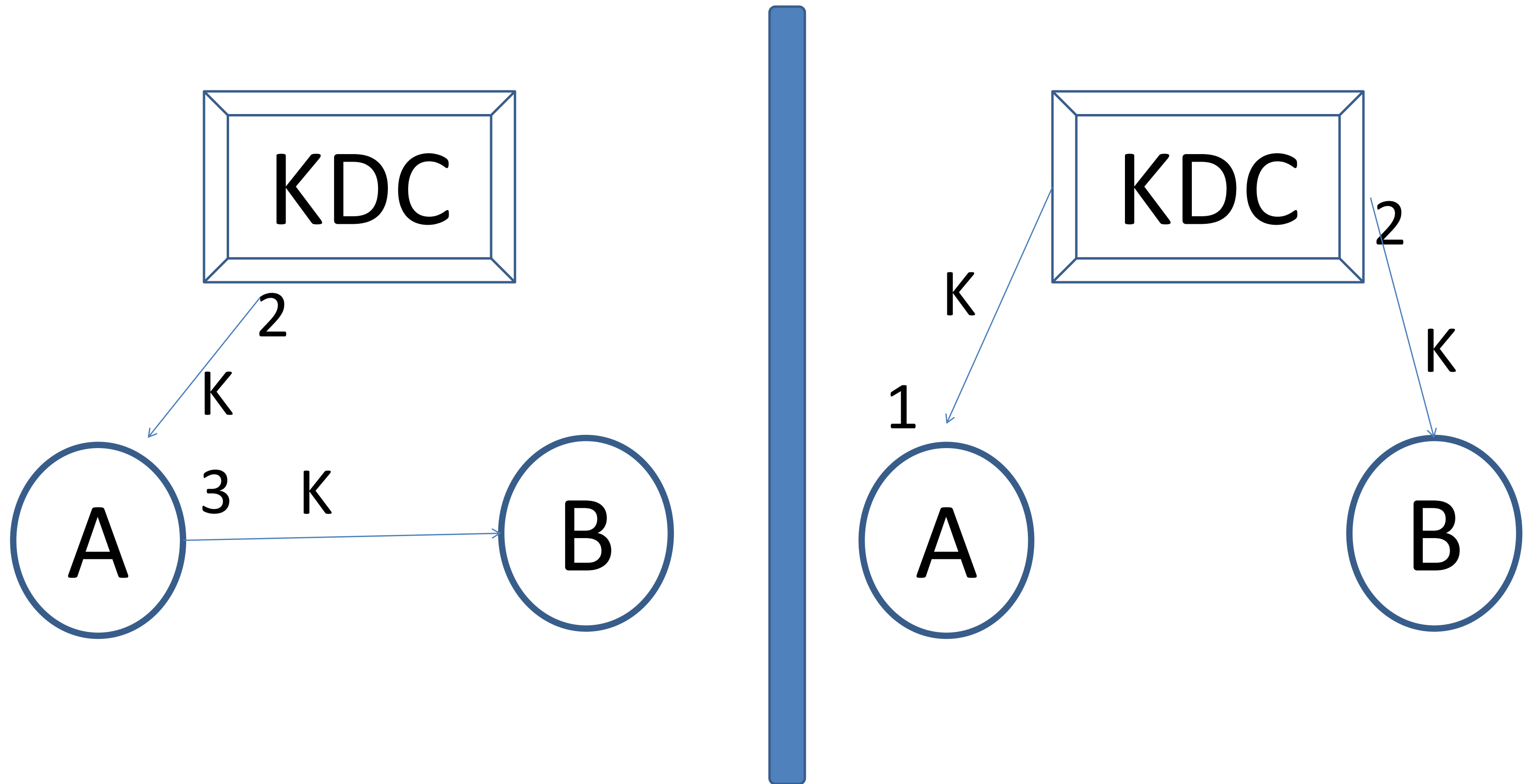
Protocoles de distribution de clés

Possibilité 2



Protocoles de distribution de clés

Possibilité 2



Protocoles de distribution de clés: Ex1

Protocole de Carlson:

Dans ce protocole S est un KDC.

1. $A \longrightarrow B : A, N_a$
2. $B \longrightarrow S : A, N_a, B, N_b$
3. $S \longrightarrow B : \{k_{ab}, N_b, A\}k_{bs}, \{N_a, B, k_{ab}\}k_{as}$
4. $B \longrightarrow A : \{N_a, B, k_{ab}\}k_{as}, \{N_a\}k_{ab}, N_b$
5. $A \longrightarrow B : \{N_b\}k_{ab}$

Protocoles de distribution de clés: Ex2

Protocole de Diffie-Hellman

Avant de commencer, Alice et Bob choisissent un nombre premier p et un générateur $g \mod p$, ($1 < g < p-1$) qui peuvent être publics.

Alice génère un nombre aléatoire a entre 1 et $p-1$

Alice calcule $A = g^a \mod p$.

Bob génère un nombre aléatoire b entre 1 et $p-1$

Bob calcule $B = g^b \mod p$.

1. Alice \rightarrow Bob: A

2. Bob \rightarrow Alice : B

Alice Calcule la clé $K = B^a \mod p$

Bob Calcule la clé $K = A^b \mod p$

Distribution de clés publiques

Distribution des clés publiques : Comment peut-on s'assurer qu'une clé publique est bien celle d'un principal donné ?



Un certificat
numérique

Les certificats

Qu'est-ce qu'un certificat numérique?

Qui délivre les certificats numériques?

Que contient un certificat numérique?

Les certificats

Qu'est-ce qu'un certificat numérique? C'est un fichier qui certifie la clé publique d'une entité;

Qui délivre les certificats numériques? Les autorités de certification (DigiCert, GlobalSign, Certigna, etc.);

Que contient un certificat numérique? (clé publique, information d'identification de l'entité, période de validité, nom de la CA, Signature de la CA, numéro de série, etc.).



Les failles des protocoles cryptographiques

Decorative geometric shapes at the bottom of the slide, including a purple triangle and a light blue trapezoid.

Failles (Introduction)

Un protocole est un algorithme distribué:

- Plusieurs participants.
- Chaque participant exécute ses actions (son rôle) sur sa machine.
- Un principal peut participer à plusieurs sessions d'un même protocole.
- Un principal peut changer de rôle d'une session à une autre.

Le rôle d'un principal

Rôle: abstraction du protocole ou l'emphasis est mise sur un principal donné.

Exemple: Protocole de carlson:

1. $A \rightarrow B : A$
2. $B \rightarrow A : N_b$
3. $A \rightarrow B : \{N_b\}k_{as}$
4. $B \rightarrow S : \{A, \{N_b\}k_{as}\}k_{bs}$
5. $S \rightarrow B : \{Nb\}k_{bs}$

Rôle de A

1. $A \rightarrow I(B) : A$
 2. $I(B) \rightarrow A : N_b$
- $$A \rightarrow I(B) : \{N_b\}k_{as}$$

Le rôle d'un principal

Exemple: Protocole de carlson:

1. $A \rightarrow B : A, N_a$
2. $B \rightarrow S : A, N_a, B, N_b$
3. $S \rightarrow B : \{k_{ab}, N_b, A\}k_{bs}, \{N_a, B, k_{ab}\}k_{as}$
4. $B \rightarrow A : \{N_a, B, k_{ab}\}k_{as}, \{N_a\}k_{ab}, N_b$
5. $A \rightarrow B : \{Nb\}k_{ab}$

Rôle de B

1. $I(A) \rightarrow B : A$
2. $B \rightarrow I(A) : N_b$
3. $I(A) \rightarrow B : \{N_b\}k_{as}$
4. $B \rightarrow I(S) : \{A, \{N_b\}k_{as}\}k_{bs}$
5. $I(S) \rightarrow B : \{Nb\}k_{bs}$

Rôle de S

4. $I(B) \rightarrow S : \{A, \{N_b\}k_{as}\}k_{bs}$
5. $S \rightarrow I(B) : \{Nb\}k_{bs}$

Rôle généralisé

Rôle Généralisé : abstraction d'un rôle où les parties d'un messages, dont la valeur n'est pas vérifiable, sont remplacées par des variables.

Rôle de A

1. $A \rightarrow I(B) : A$
2. $I(B) \rightarrow A : N_b$
3. $A \rightarrow I(B) : \{N_b\}k_{as}$

$R1: A \rightarrow I(B) : A$

$R2: I(B) \rightarrow A : X$

$R3: A \rightarrow I(B) : \{X\}k_{as}$

Rôle de B

1. $I(A) \rightarrow B : A$
2. $B \rightarrow I(A) : N_b$
3. $I(A) \rightarrow B : \{N_b\}k_{as}$
4. $B \rightarrow I(S) : \{A, \{N_b\}k_{as}\}k_{bs}$
5. $I(S) \rightarrow B : \{Nb\}k_{bs}$

Rôle de S

4. $I(B) \rightarrow S : \{A, \{N_b\}k_{as}\}k_{bs}$
5. $S \rightarrow I(B) : \{Nb\}k_{bs}$

Les failles

Rôle Généralisé : abstraction d'un rôle où les parties d'un messages, dont la valeur n'est pas vérifiable, sont remplacées par des variables.

Rôle de A

1. $A \rightarrow I(B) : A$
2. $I(B) \rightarrow A : N_b$
3. $A \rightarrow I(B) : \{N_b\}k_{as}$

Rôle de B

1. $I(A) \rightarrow B : A$
2. $B \rightarrow I(A) : N_b$
3. $I(A) \rightarrow B : \{N_b\}k_{as}$
4. $B \rightarrow I(S) : \{A, \{N_b\}k_{as}\}k_{bs}$
5. $I(S) \rightarrow B : \{Nb\}k_{bs}$

Rôle de S

4. $I(B) \rightarrow S : \{A, \{N_b\}k_{as}\}k_{bs}$
5. $S \rightarrow I(B) : \{Nb\}k_{bs}$

R1: $A \rightarrow I(B) : A$

R2: $I(B) \rightarrow A : X_1$

R3: $A \rightarrow I(B) : \{X_1\}k_{as}$

R1: $I(A) \rightarrow B : A$

R2: $B \rightarrow I(A) : N_b$

R3: $I(A) \rightarrow B : X_2$

R4: $B \rightarrow I(S) : \{A, X_2\}k_{bs}$

R5: $I(S) \rightarrow B : \{Nb\}k_{bs}$

R4: $I(B) \rightarrow S : \{A, \{X_3\}k_{as}\}k_{bs}$

R5: $S \rightarrow I(B) : \{X_3\}k_{bs}$

Trace d'exécution

Une trace est une exécution valide d'un protocole.

Exécution valide d'un protocole :

- tous les agents honnêtes, qui y participent se comportent conformément à la spécification du protocole.
- tous les messages envoyés par l'intrus, doivent appartenir à sa base de connaissances (connaissances initiales + messages reçus).

Trace d'exécution

Une trace est une exécution valide d'un protocole.

Exécution valide d'un protocole :

- tous les agents honnêtes, qui y participent se comportent conformément à la spécification du protocole.
- tous les messages envoyés par l'intrus, doivent appartenir à sa base de connaissances (connaissances initiales + messages reçus).

Trace d'exécution

R1: $A \rightarrow I(B) : A$

R2: $I(B) \rightarrow A : X_1$

R3: $A \rightarrow I(B) : \{X_1\}k_{as}$

R1: $I(A) \rightarrow B : A$

R2: $B \rightarrow I(A) : N_b$

R3: $I(A) \rightarrow B : X_2$

R4: $B \rightarrow I(S) : \{A, X_2\}k_{bs}$

R5: $I(S) \rightarrow B : \{Nb\}k_{bs}$

R4: $I(B) \rightarrow S : \{A, \{X_3\}k_{as}\}k_{bs}$

R5: $S \rightarrow I(B) : \{X_3\}k_{bs}$

1.1: $I(A) \rightarrow B : A$

1.2: $B \rightarrow I(A) : N_b$

1.3: $I(A) \rightarrow B : \text{Anything}$

1.4: $B \rightarrow I(S) : \{A, \text{Anything}\}k_{bs}$

2.1: $C \rightarrow I(D) : C$

2.2: $I(D) \rightarrow C : N_b$

2.3: $C \rightarrow I(D) : \{N_b\}_{Kcs}$

3.1: $C \rightarrow I(E) : C$

3.2: $I(E) \rightarrow C : \{C, \{N_b\}_{Kcs}\}$

3.3: $C \rightarrow I(E) : \{C, \{N_b\}_{Kcs}\}_{Kcs}$

4.1: $I(C) \rightarrow B : C$

4.2: $B \rightarrow I(C) : N'_b$

4.3: $I(C) \rightarrow B : \{C, \{N_b\}_{Kcs}\}_{Kcs}$

4.4: $B \rightarrow I(S) : \{C, \{C, \{N_b\}_{Kcs}\}_{Kcs}\}_{Kbs}$

4.5: $I(S) \rightarrow B : \{\{C, \{C, \{N_b\}_{Kcs}\}_{Kcs}\}_{Kbs}\}$

5.4: $I(B) \rightarrow S : \{C, \{Nb\}_{Kcs}\}_{Kbs}$

5.5: $S \rightarrow I(B) : \{Nb\}_{Kcs}$

1.5: $I(S) \rightarrow B : \{Nb\}k_{bs}$

Trace d'exécution

$R1: A \rightarrow I(B) : A$

$R2: I(B) \rightarrow A : X_1$

$R3: A \rightarrow I(B) : \{X_1\}k_{as}$

$R1: I(A) \rightarrow B : A$

$R2: B \rightarrow I(A) : N_b$

$R3: I(A) \rightarrow B : X_2$

$R4: B \rightarrow I(S) : \{A, X_2\}k_{bs}$

$R5: I(S) \rightarrow B : \{Nb\}k_{bs}$

$R4: I(B) \rightarrow S : \{A, \{X_3\}k_{as}\}k_{bs}$

$R5: S \rightarrow I(B) : \{X_3\}k_{bs}$

1.1: $I(A) \rightarrow B : A$

1.2: $B \rightarrow I(A) : N_b$

1.3: $I(A) \rightarrow B : \text{Anything}$

1.4: $B \rightarrow I(S) : \{A, \text{Anything}\}k_{bs}$

2.1: $C \rightarrow I(D) : C$

2.2: $I(D) \rightarrow C : N_b$

2.3: $C \rightarrow I(D) : \{N_b\}k_{cs}$

3.1 $C \rightarrow I(E) : C$

3.2: $I(E) \rightarrow C : \{C, \{N_b\}k_{cs}\}$

3.3: $C \rightarrow I(E) : \{C, \{N_b\}k_{cs}\}k_{cs}$

4.1: $I(C) \rightarrow B : C$

4.2: $B \rightarrow I(C) : N'_b$

4.3: $I(C) \rightarrow B : \{C, \{N_b\}k_{cs}\}k_{cs}$

4.4: $B \rightarrow I(S) : \{C, \{C, \{N_b\}k_{cs}\}k_{cs}\}k_{bs}$

4.5: $I(S) \rightarrow B : \{\{C, \{C, \{N_b\}k_{cs}\}k_{cs}\}k_{bs}\}$

5.4: $I(B) \rightarrow S : \{C, \{Nb\}k_{cs}\}k_{bs}$

5.5: $S \rightarrow I(B) : \{Nb\}k_{cs}$

1.5: $I(S) \rightarrow B : \{Nb\}k_{bs}$

Trace d'exécution

$R1: A \rightarrow I(B) : A$

$R2: I(B) \rightarrow A : X_1$

$R3: A \rightarrow I(B) : \{X_1\}k_{as}$

$R1: I(A) \rightarrow B : A$

$R2: B \rightarrow I(A) : N_b$

$R3: I(A) \rightarrow B : X_2$

$R4: B \rightarrow I(S) : \{A, X_2\}k_{bs}$

$R5: I(S) \rightarrow B : \{Nb\}k_{bs}$

$R4: I(B) \rightarrow S : \{A, \{X_3\}k_{as}\}k_{bs}$

$R5: S \rightarrow I(B) : \{X_3\}k_{bs}$

1.1: $I(A) \rightarrow B : A$

1.2: $B \rightarrow I(A) : N_b$

1.3: $I(A) \rightarrow B : \text{Anything}$

1.4: $B \rightarrow I(S) : \{A, \text{Anything}\}k_{bs}$

2.1: $C \rightarrow I(D) : C$

2.2: $I(D) \rightarrow C : N_b$

2.3: $C \rightarrow I(D) : \{N_b\}_{Kcs}$

3.1 $C \rightarrow I(E) : C$

3.2: $I(E) \rightarrow C : \{C, \{N_b\}_{Kcs}\}_{Kcs}$

3.3: $C \rightarrow I(E) : \{C, \{N_b\}_{Kcs}\}_{Kcs}$

4.1: $I(C) \rightarrow B : C$

4.2: $B \rightarrow I(C) : N'_b$

4.3: $I(C) \rightarrow B : \{C, \{N_b\}_{Kcs}\}_{Kcs}$

4.4: $B \rightarrow I(S) : \{C, \{C, \{N_b\}_{Kcs}\}_{Kcs}\}_{Kbs}$

4.5: $I(S) \rightarrow B : \{\{C, \{C, \{N_b\}_{Kcs}\}_{Kcs}\}_{Kbs}\}_{Kbs}$

5.4: $I(B) \rightarrow S : \{C, \{Nb\}_{Kcs}\}_{Kbs}$

5.5: $S \rightarrow I(B) : \{Nb\}_{Kcs}$

1.5: $I(S) \rightarrow B : \{Nb\}k_{bs}$

Trace d'exécution

R1: $A \rightarrow I(B) : A$

R2: $I(B) \rightarrow A : X_1$

R3: $A \rightarrow I(B) : \{X_1\}k_{as}$

R1: $I(A) \rightarrow B : A$

R2: $B \rightarrow I(A) : N_b$

R3: $I(A) \rightarrow B : X_2$

R4: $B \rightarrow I(S) : \{A, X_2\}k_{bs}$

R5: $I(S) \rightarrow B : \{Nb\}k_{bs}$

R4: $I(B) \rightarrow S : \{A, \{X_3\}k_{as}\}k_{bs}$

R5: $S \rightarrow I(B) : \{X_3\}k_{bs}$

1.1: $I(A) \rightarrow B : A$

1.2: $B \rightarrow I(A) : N_b$

1.3: $I(A) \rightarrow B : \text{Anything}$

1.4: $B \rightarrow I(S) : \{A, \text{Anything}\}k_{bs}$

2.1: $C \rightarrow I(D) : C$

2.2: $I(D) \rightarrow C : N_b$

2.3: $C \rightarrow I(D) : \{N_b\}_{Kcs}$

3.1: $C \rightarrow I(E) : C$

3.2: $I(E) \rightarrow C : \{C, \{N_b\}_{Kcs}\}$

3.3: $C \rightarrow I(E) : \{C, \{N_b\}_{Kcs}\}_{Kcs}$

4.1: $I(C) \rightarrow B : C$

4.2: $B \rightarrow I(C) : N'_b$

4.3: $I(C) \rightarrow B : \{C, \{N_b\}_{Kcs}\}_{Kcs}$

4.4: $B \rightarrow I(S) : \{C, \{C, \{N_b\}_{Kcs}\}_{Kcs}\}_{Kbs}$

4.5: $I(S) \rightarrow B : \{\{C, \{C, \{N_b\}_{Kcs}\}_{Kcs}\}_{Kbs}\}$

5.4: $I(B) \rightarrow S : \{C, \{Nb\}_{Kcs}\}_{Kbs}$

5.5: $S \rightarrow I(B) : \{Nb\}_{Kcs}$

1.5: $I(S) \rightarrow B : \{Nb\}k_{bs}$

Trace d'exécution

R1: $A \rightarrow I(B) : A$

R2: $I(B) \rightarrow A : X_1$

R3: $A \rightarrow I(B) : \{X_1\}k_{as}$

R1: $I(A) \rightarrow B : A$

R2: $B \rightarrow I(A) : N_b$

R3: $I(A) \rightarrow B : X_2$

R4: $B \rightarrow I(S) : \{A, X_2\}k_{bs}$

R5: $I(S) \rightarrow B : \{Nb\}k_{bs}$

R4: $I(B) \rightarrow S : \{A, \{X_3\}k_{as}\}k_{bs}$

R5: $S \rightarrow I(B) : \{X_3\}k_{bs}$

1.1: $I(A) \rightarrow B : A$

1.2: $B \rightarrow I(A) : N_b$

1.3: $I(A) \rightarrow B : \text{Anything}$

1.4: $B \rightarrow I(S) : \{A, \text{Anything}\}k_{bs}$

2.1: $C \rightarrow I(D) : C$

2.2: $I(D) \rightarrow C : N_b$

2.3: $C \rightarrow I(D) : \{N_b\}_{Kcs}$

3.1: $C \rightarrow I(E) : C$

3.2: $I(E) \rightarrow C : \{C, \{N_b\}_{Kcs}\}$

3.3: $C \rightarrow I(E) : \{C, \{N_b\}_{Kcs}\}_{Kcs}$

4.1: $I(C) \rightarrow B : C$

4.2: $B \rightarrow I(C) : N'_b$

4.3: $I(C) \rightarrow B : \{C, \{N_b\}_{Kcs}\}_{Kcs}$

4.4: $B \rightarrow I(S) : \{C, \{C, \{N_b\}_{Kcs}\}_{Kcs}\}_{Kbs}$

4.5: $I(S) \rightarrow B : \{\{C, \{C, \{N_b\}_{Kcs}\}_{Kcs}\}_{Kbs}\}$

5.4: $I(B) \rightarrow S : \{C, \{Nb\}_{Kcs}\}_{Kbs}$

5.5: $S \rightarrow I(B) : \{Nb\}_{Kcs}$

1.5: $I(S) \rightarrow B : \{Nb\}k_{bs}$

Modèle d'intrus DY

Le modèle de Dolev-Yao est un modèle d'intrus (très puissant) qui est couramment utilisé dans le domaine de la sécurité informatique pour analyser la sécurité des protocoles de communications. Il stipule qu'un intrus:

peut être un agent régulier (possède ses propres clés);

A une forte capacité de calcul;

A un accès aux messages (intercepter, modifier, insérer ou supprimer);

A une capacité de mémorisation infinie;

Analyse complète et logique de déduction;

On dit qu'un protocole contient une faille s'il ne remplit pas les exigences de sécurité pour lesquelles il a été conçu.

Exemples

Un protocole d'authentification contient une faille si un agent *A* arrive à prouver à un agent *B* qu'il est un autre agent *C*.

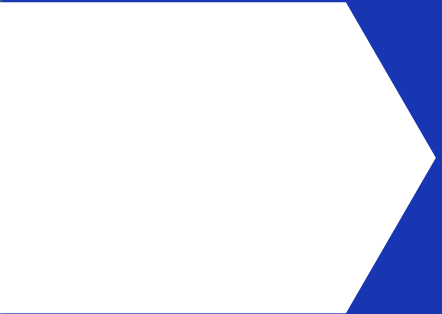
Un protocole qui permet d'acheminer des informations confidentielles est dit défaillant si un intrus est capable de connaître une des ces informations secrètes.

La preuve de l'existence d'une faille dans un protocole cryptographique est généralement une trace valide de ce protocole montrant qu'un des objectifs visé (une propriété de sécurité) n'est pas atteint.


Classes de failles

Les principales classes des failles des protocoles cryptographiques sont:

- faille de fraîcheur;
- failles d'oracle;
- failles d'association;
- failles de types;
- failles d'implantation.



Verification des protocoles–Méthode formelle (Logique BAN)

At the bottom right of the slide, there are two overlapping geometric shapes: a purple triangle pointing upwards and a light blue trapezoid pointing to the right.

Logique BAN

- La logique BAN baptisée selon les noms de ses inventeurs Michael Burrows, Martin Abadi et Roger Needham.
- C'est la première et la plus célèbre logique dédiée pour l'analyse des protocoles cryptographiques.
- C'est une logique de croyance : modélise les croyances des principaux impliqués dans un protocole et l'évolution de ces croyances après échange de messages durant l'exécution du protocole.

Rend les protocoles plus efficaces en répondant aux questions suivantes :

- Le protocole fait-t-il des actions en plus, c'est-à-dire des actions qu'on peut supprimer sans affaiblir la sécurité assurée par le protocole.
- Le protocole crypte-t-il des messages qu'on peut envoyer en texte clair sans l'affaiblir la sécurité assurée par le protocole.

Syntaxe de la logique BAN

$P \models X$: (P croit X) (P croit en la véracité de X)

$P \triangleleft X$: (P voit X) Quelqu'un a envoyé le message X à P

$P \sim X$: (P dit X) P a envoyé un message contenant X. (On ne sait pas si le message a été envoyé dans l'exécution courante, du protocole ou bien dans une exécution antérieure. Dans les deux cas, P croyait X ($P \models X$) lorsqu'il a envoyé le message.)

Syntaxe de la logique BAN

$P \models X$ (P a une juridiction sur X) : Dans plusieurs protocoles, les principaux se fient à un serveur pour générer et gérer les clés. Ceci peut être exprimé par l'hypothèse qui dit que les principaux croient que le serveur a une juridiction sur clés.

$\#(X)$ (La formule X est fraîche): La formule X n'a été envoyée dans aucun message des exécutions précédentes du protocole.

Syntaxe de la logique BAN

$A \xleftrightarrow{K} B$ (K est la clé partagée entre A et B)

$\{X\}_K$ X est crypté par la clé K

$\overset{K}{\rightarrow} B$ (K est la publique de B) (K^{-1} est sa clé privée)

$\{X\}_{K^{-1}}$ X est signé par la clé K^{-1}

Postulats de la logique BAN

$$\frac{P \models Q \overset{K}{\leftrightarrow} P, \quad P \triangleleft \{X\}_K}{P \models Q \sim X} \mathbf{R1}$$

$$\frac{P \models Q \Rightarrow X, \quad P \models Q \models X}{P \models X} \mathbf{R2}$$

$$\frac{P \models \overset{K}{\mapsto} Q, \quad P \triangleleft \{X\}_{K^{-1}}}{P \models Q \sim X} \mathbf{R3}$$

$$\frac{P \models \#(X), \quad P \models Q \sim X}{P \models Q \models X} \mathbf{R4}$$

$$\frac{P \models X, \quad P \models Y}{P \models (X, Y)}$$

$$\frac{P \models (X, Y)}{P \models X}$$

$$\frac{P \models Q \models (X, Y)}{P \models Q \models X} \mathbf{R5, R6, R7}$$

Postulats de la logique BAN

$$\frac{P \models Q \sim (X, Y)}{P \models Q \sim X} \text{R8}$$

$$\frac{P \models \overset{K}{\mapsto} Q, \quad P \triangleleft \{X\}_{K^{-1}}}{P \triangleleft X} \text{R9}$$

$$\frac{P \models \overset{K}{\mapsto} P, \quad P \triangleleft \{X\}_K}{P \triangleleft X} \text{R10}$$

$$\frac{P \models \sharp(X)}{P \models \sharp(X, Y)} \text{R11}$$

$$\frac{P \models R \overset{K}{\leftrightarrow} R'}{P \models R' \overset{K}{\leftrightarrow} R} \text{R12} \quad \frac{P \models Q \models R \overset{K}{\leftrightarrow} R'}{P \models Q \models R' \overset{K}{\leftrightarrow} R} \text{R13}$$

Processus de démonstration BAN

Première phase:

Idéalisation (I1, I2, etc.)

Hypothèses (H1, H2, etc.)

Fixer les objectifs (O1, O2, etc.)

Seconde phase: (La preuve)

Application des postulats sur les I, les H, les résultats intermédiaires pour atteindre les O.

Exemple protocole Kerberos

$A \rightarrow S: A, B$

$S \rightarrow A: \{T_s, K_{ab}, B, \{T_s, K_{ab}, A\}_{K_{bs}}\}_{K_{as}}$

$A \rightarrow B: \{T_s, K_{ab}, A\}_{K_{bs}}, \{A, T_a\}_{K_{ab}}$

$B \rightarrow A: \{T_a + 1\}_{K_{ab}}$

Le but est de montrer que A et B croient en la clé K_{ab} .

Exemple protocole Kerberos

$A \rightarrow S: A, B$

$S \rightarrow A: \{T_s, K_{ab}, B, \{T_s, K_{ab}, A\}_{K_{bs}}\}_{K_{as}}$

$A \rightarrow B: \{T_s, K_{ab}, A\}_{K_{bs}}, \{A, T_a\}_{K_{ab}}$

$B \rightarrow A: \{T_a + 1\}_{K_{ab}}$

Le but est de montrer que A et B croient en la clé K_{ab} .

BAN (Idéalisation)

Idéalisation: Enlever le contenu en clair, enlever les fonctions de calcul et les identifiants, transformer chaque message en deux formules BAN.

Exemples:

Transformer ces messages en formule BAN (I1 jusqu'à I6)

$S \rightarrow A: \{T_s, K_{ab}, \{T_s, K_{ab}\}_{K_{bs}}\}_{K_{as}}$

$A \rightarrow B: \{T_s, K_{ab}\}_{K_{bs}}, \{A, T_a\}_{K_{ab}}$

$B \rightarrow A: \{T_a + 1\}_{K_{ab}}$

Hypothèses

Hypothèses: c'est une formalisation de ce qui nous semble évident.

$$A \mid \equiv A \xleftrightarrow{K_{as}} S \quad (H1)$$

$$S \mid \equiv A \xleftrightarrow{K_{as}} S \quad (H2)$$

$$S \mid \equiv A \xleftrightarrow{K_{as}} B \quad (H3)$$

$$A \mid \equiv (S \rightrightarrows A \xleftrightarrow{K_{ab}} B) \quad (H4)$$

$$A \mid \equiv \#(T_s) \quad (H5)$$

$$B \mid \equiv B \xleftrightarrow{K_{as}} S \quad (H6)$$

$$S \mid \equiv B \xleftrightarrow{K_{as}} S \quad (H7)$$

$$B \mid \equiv S \rightrightarrows A \xleftrightarrow{K_{ab}} B \quad (H8)$$

$$B \mid \equiv \#(T_s) \quad (H9)$$

$$B \mid \equiv \#(T_a) \quad (H10)$$

Les objectifs

Les objectifs de sécurité: Les services de sécurité garantis (La logique BAN vérifie l'authentification et l'intégrité et la fraîcheur)

$$O1: A \mid \equiv A \overset{K_{ab}}{\longleftrightarrow} B$$

$$O2: B \mid \equiv A \overset{K_{ab}}{\longleftrightarrow} B$$

NB: parfois on a besoin de prouver une double croyance.

La preuve

Donner une suite de règle et laisser les étudiants les appliquer



Verification des protocoles

Decorative geometric shapes at the bottom of the slide, including a purple triangle and a light blue trapezoid.

Les analyseurs AVISPA

Scyther : Un outil open source écrit en Python, Scyther est utilisé pour l'analyse de protocoles cryptographiques.

CASL (Cryptographic Analysis Server Language) : Un framework spécialisé dans la spécification et l'analyse de protocoles cryptographiques.

Avispa : Avispa se distingue par son interface utilisateur graphique conviviale, ce qui en fait un outil attrayant pour la vérification de protocoles de sécurité.

ProVerif : Cet outil de vérification automatique des protocoles cryptographiques repose sur une analyse formelle rigoureuse.



**MERCI POUR
VOTRE ATTENTION**

