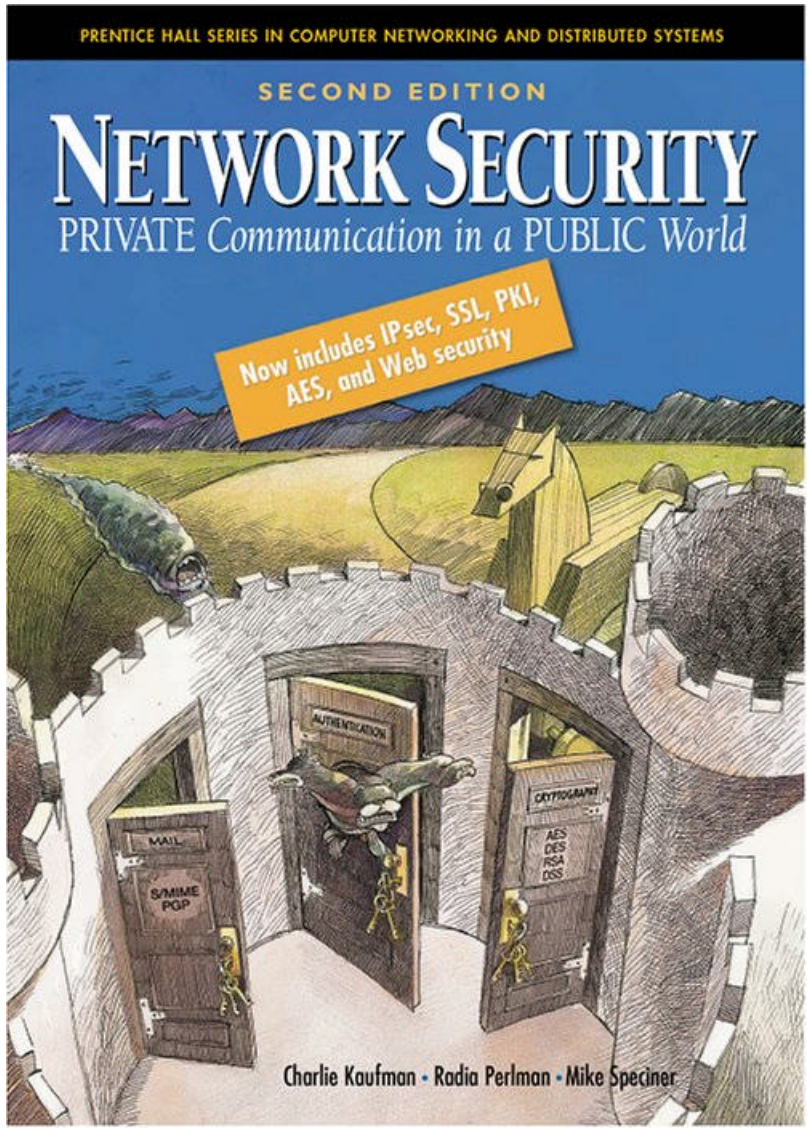
Sécurité des Réseaux, Master CSI 2

J.Bétréma, LaBRI, Université Bordeaux 1

Protocoles d'authentification

1. Authentification simple
2. Authentification mutuelle
3. Clé de session
4. KDC

Source


PRENTICE HALL SERIES IN COMPUTER NETWORKING AND DISTRIBUTED SYSTEMS
SECOND EDITION
NETWORK SECURITY
PRIVATE Communication in a PUBLIC World
Now includes IPsec, SSL, PKI, AES, and Web security
AUTHENTICATION
MAIL
S/MIME PGP
CRYPTOGRAPHY
AES DES RSA DSS
Charlie Kaufman • Radia Perlman • Mike Speciner

1. Authentification simple

1.1 Clé secrète partagée

Protocol 11-1. Bob authenticates Alice based on a shared secret $K_{Alice\text{-}Bob}$

- La clé secrète (partagée) est utilisée pour chiffrer R : notation *f* (K, R) = K{R}

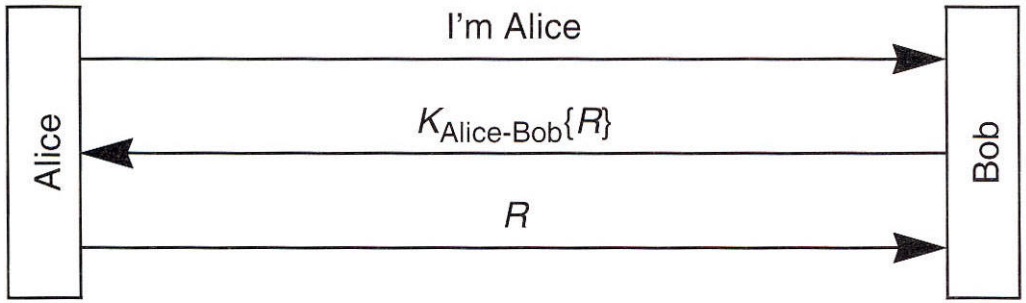
ou bien:

- *f* est une fonction de hachage.

Vulnérabilités :

- Clés stockées dans une base de données sur le serveur (Bob), et sur tous les serveurs pour lesquels le client (Alice) utilise la même clé…
- Clair connu (R) ⇒ Attaque de dictionnaire possible si clé faible (par ex. dérivée d'un mot de passe faible).

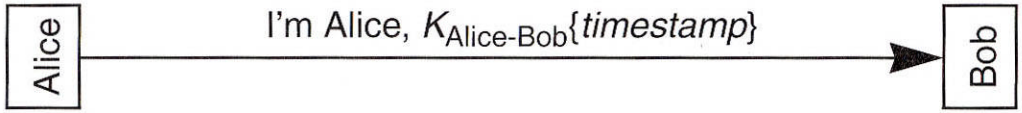
Variante



Protocol 11-2. Bob authenticates Alice based on a shared secret key $K_{Alice\text{-}Bob}$

Alice prouve sa capacité de déchiffrer.

Estampille



Protocol 11-3. Bob authenticates Alice based on synchronized clocks and a shared secret $K_{Alice\text{-}Bob}$

- Bob déchiffre, et vérifie que l'estampille appartient à un intervalle de temps acceptable
- Un seul message, attaque de dictionnaire difficile
- Demande des horloges assez bien synchronisées

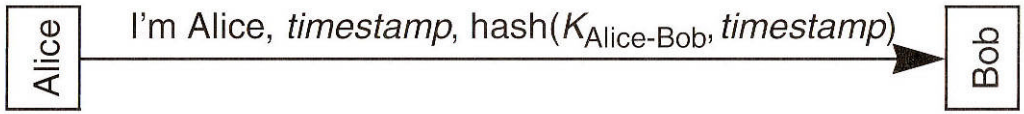
Vulnérabilités et remèdes :

- Replay ⇒ Bob doit mémoriser les estampilles utilisées par Alice, jusqu'à leur expiration.
- Replay vers un autre serveur (pour lequel le client (Alice) utilise la même clé…)

  ⇒ Ajouter (concaténer) l'identité du serveur à l'estampille.

Estampille (2)

Pour utiliser une fonction de hachage (au lieu d'un chiffrement), il faut aussi transmettre l'estampille en clair :

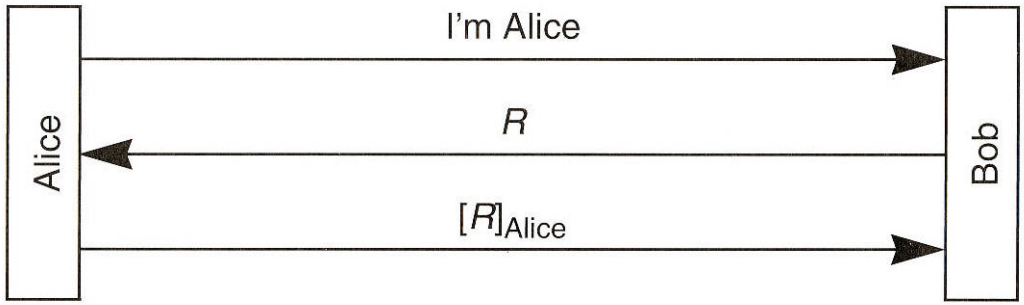


Protocol 11-4. Bob authenticates Alice based on hashing a high-resolution time and a shared secret $K_{Alice-Bob}$

Attaque de dictionnaire à nouveau possible.

Authentification simple

1.2 Clé publique



Protocol 11-5. Bob authenticates Alice based on her public key signature

- Alice utilise sa clé secrète pour chiffrer R (signature).
- Bob utilise la clé publique d'Alice pour déchiffrer.

La base de données du serveur (Bob) ne contient que des clés publiques: protégée seulement en *écriture.*

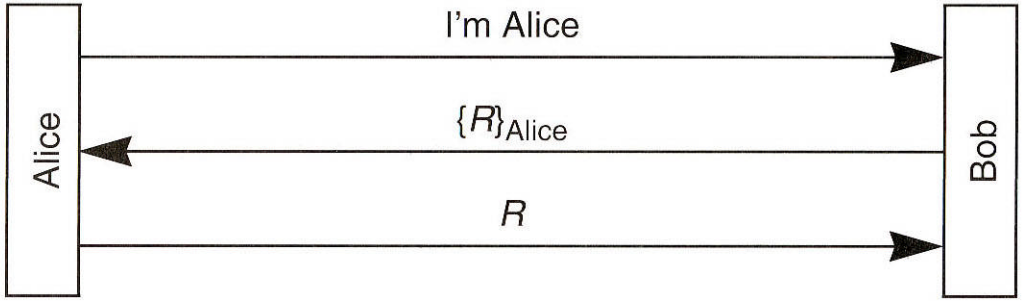
Vulnérabilité :

Un intrus, qui usurpe l'identité de Bob, obtient un message de son choix (R) *signé* par Alice, utilisable dans un autre contexte.

Remède :

Typer R (en-tête; cf. **PKCS**) ; vérifier type et taille avant de signer.

Variante



Alice prouve sa capacité de déchiffrer.

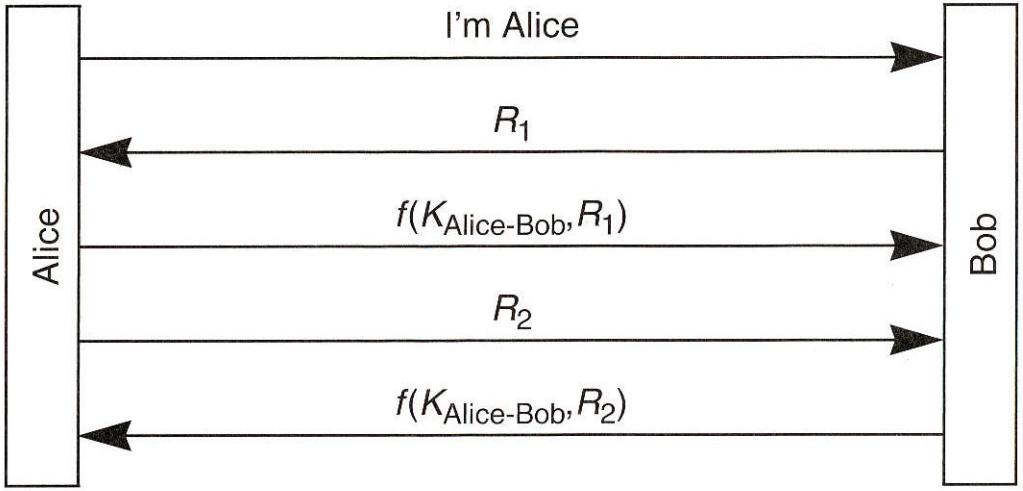
Protocol 11-6. Bob authenticates Alice if she can decrypt a message encrypted with her public key

Vulnérabilité :

Un intrus, qui usurpe l'identité de Bob, fait *déchiffrer* par Alice un message de son choix (peut-être intercepté dans un autre contexte).

2. Authentification mutuelle

2.1 Clé secrète partagée

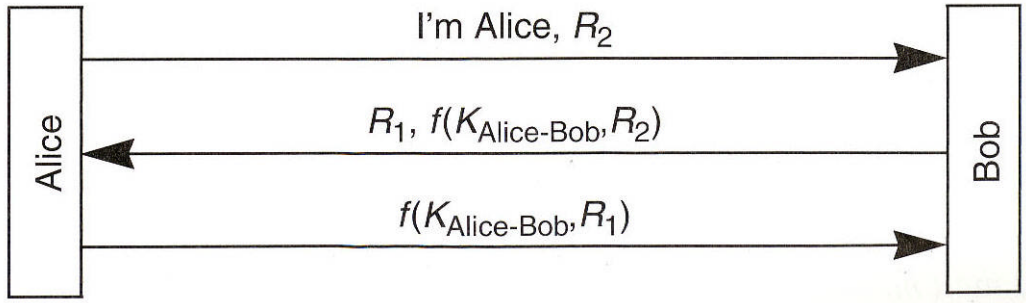


Protocol 11-7. Mutual authentication based on a shared secret $K_{\text{Alice-Bob}}$

Protocole 1 exécuté dans les deux sens. Celui qui prend l'initiative s'authentifie en premier.

> *…if you only spoke when you were spoken to, and the other person always waited for* you *to begin, you see nobody would ever say anything…*
>
> —Alice (in *Through the Looking Glass*)

Attaque par réflexion



Protocol 11-8. Optimized mutual authentication based on a shared secret $K_{\text{Alice-Bob}}$

Ce protocole, qui semble être une simple optimisation du précédent, présente une vulnérabilité fondamentale (absente du protocole 7)

Attaque par réflexion (2)

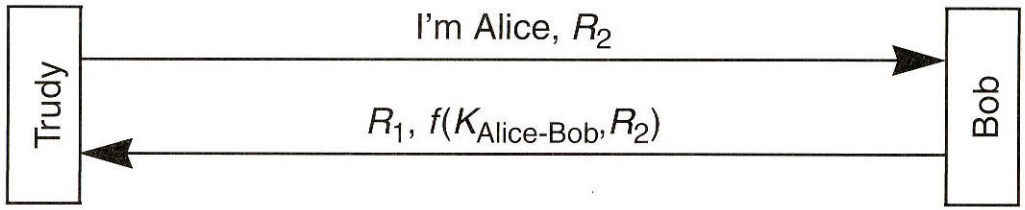


Figure 11-9. Beginning of reflection attack

"I can't explain myself, *I'm afraid sir," said Alice, "because I'm not myself, you see."*

Alice in Wonderland

Trudy ouvre une seconde session, pour obtenir de Bob l'authentification qu'elle ne peut fabriquer !

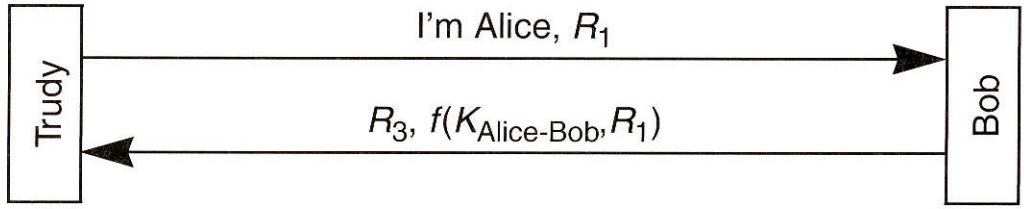


Figure 11-10. Second session in reflection attack

Trudy ferme la seconde session, et peut maintenant reprendre la première session et s'authentifier !

Attaque par réflexion (3)

Contre-mesures

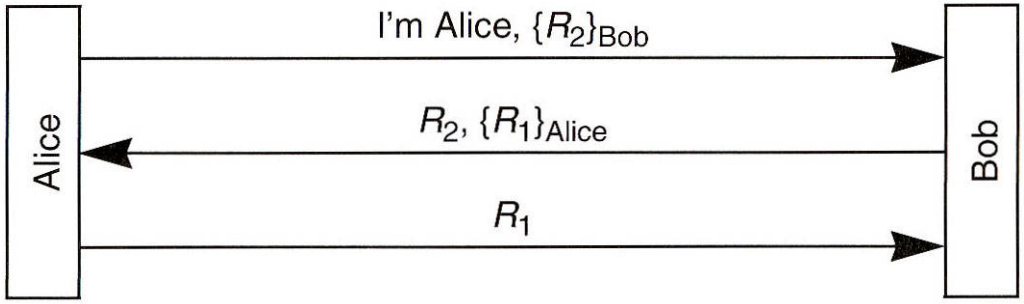
- Alice et Bob utilisent des clés différentes (dérivées de K), par exemple K et K+1

- Alice et Bob utilisent des défis de types différents, par exemple Alice calcule *f* (K, **Alice**|R), tandis que Bob calcule *f* (K, **Bob**|R)

Mais ce protocole reste particulièrement vulnérable à une attaque de dictionnaire, puisque Bob effectue le calcul demandé par Alice, *avant* authentification d'Alice.

Authentification mutuelle

2.2 Clé publique



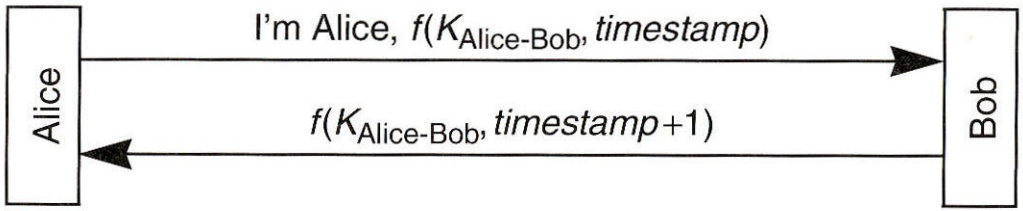
Protocol 11-12. Mutual authentication with public keys

Dissymétrie : pas d'attaque par réflexion

- Variante : Alice envoie R_2 et Bob retourne R_2 signé (idem pour R_1).
- PKI (Public Key Infrastructure) : comment stocker la clé publique de Bob et la clé privée d'Alice de façon sûre.
- Faiblesse : Bob déchiffre un message choisi par Alice (voir protocoles 5 et 6).

Authentification mutuelle

2.3 Estampilles



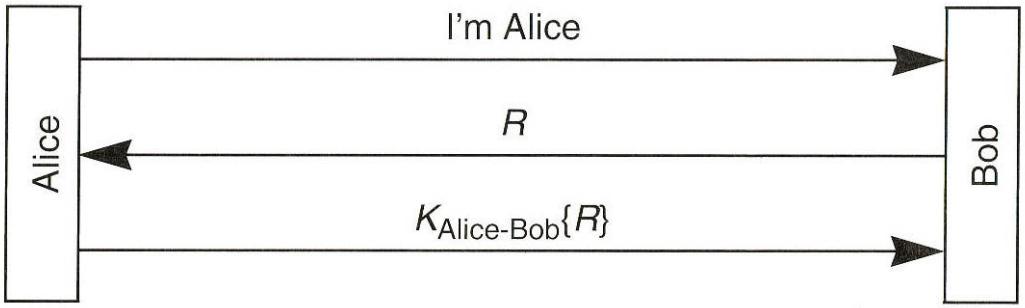
Protocol 11-13. Mutual authentication based on synchronized clocks and a shared secret $K_{Alice-Bob}$

- Kerberos V4 ! *f* désigne la fonction de chiffrement.
- Bob doit mémoriser les valeurs de *timestamp* et *timestamp* + 1 pour éviter les attaques par **replay**
- Variantes comme dans le protocole 3 (concaténer nom du destinataire et estampille)

3. Clé de session

- Indispensable pour éviter un **détournement** (**hijacking**) de session après authentification.
- Clé temporaire.

3.1 Après authentification par clé secrète partagée



Protocol 11-14. Authentication with shared secret

Clés de session possibles :

- (K+1){R} : OK
- K{R+1} : faible
- etc.

Clé de session calculée à partir de K (secret) et R (différent pour chaque session)

Clé de session

3.2 Après authentification par clé publique

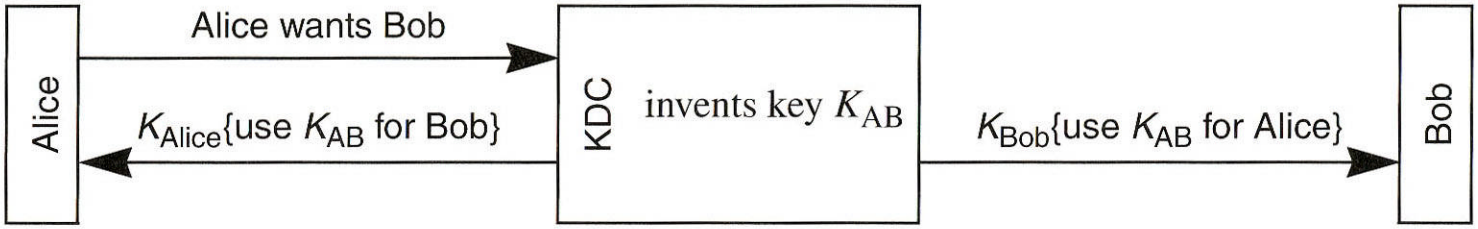
- Alice choisit une clé S (typée), la chiffre avec la clé publique de Bob, et signe le résultat avec sa clef privée.

- Un attaquant qui enregistre ce message, et la session cryptée, peut tout décrypter plus tard s'il se rend maître (**overruns**) de Bob.

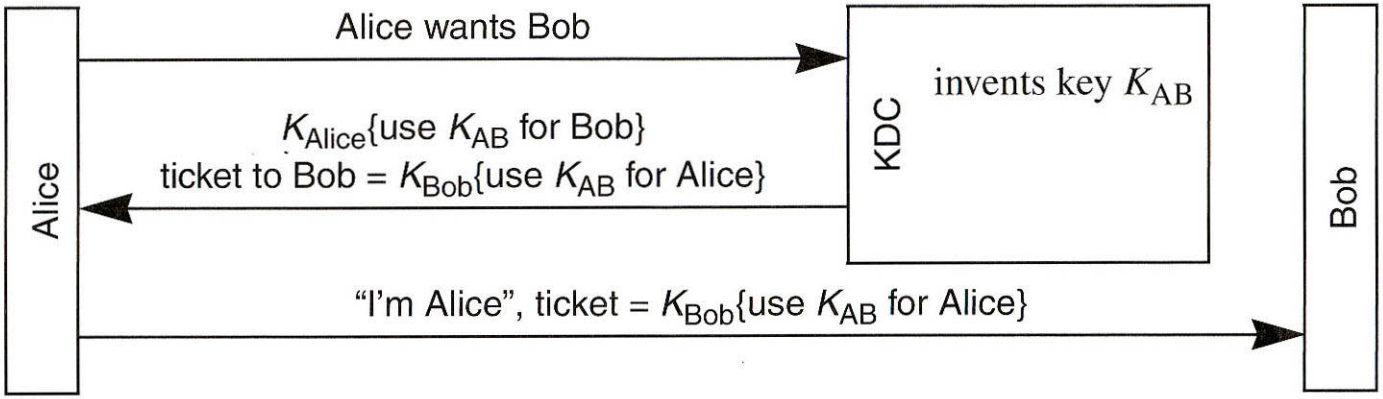
- Contre-mesure : Alice émet $\{S_1\}_{Bob}$ et Bob émet $\{S_2\}_{Alice}$; la clé $S_1 \oplus S_2$ ne peut être retrouvée qu'après « invasion » d'Alice et Bob.

- Diffie-Hellmann avec des messages signés : **PFS** (Perfect Forward Secrecy)

4. KDC (Key Distribution Center)

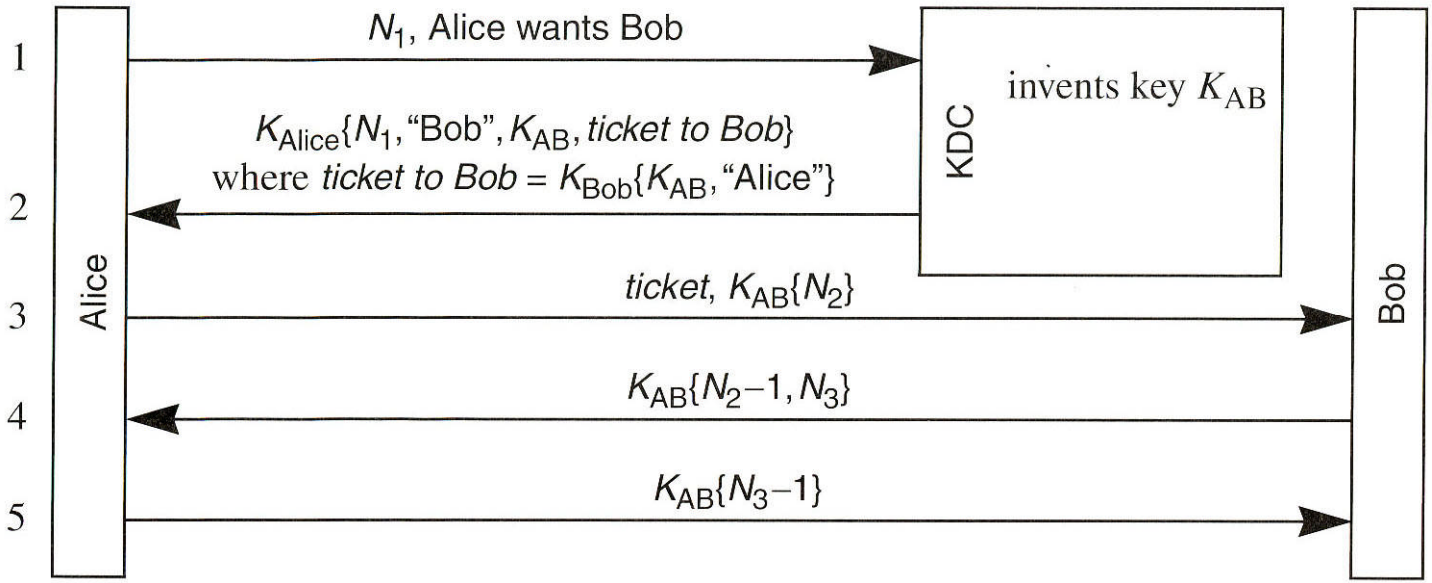


Protocol 11-16. KDC operation (in principle)



Protocol 11-17. KDC operation (in practice)

Needham-Schroeder

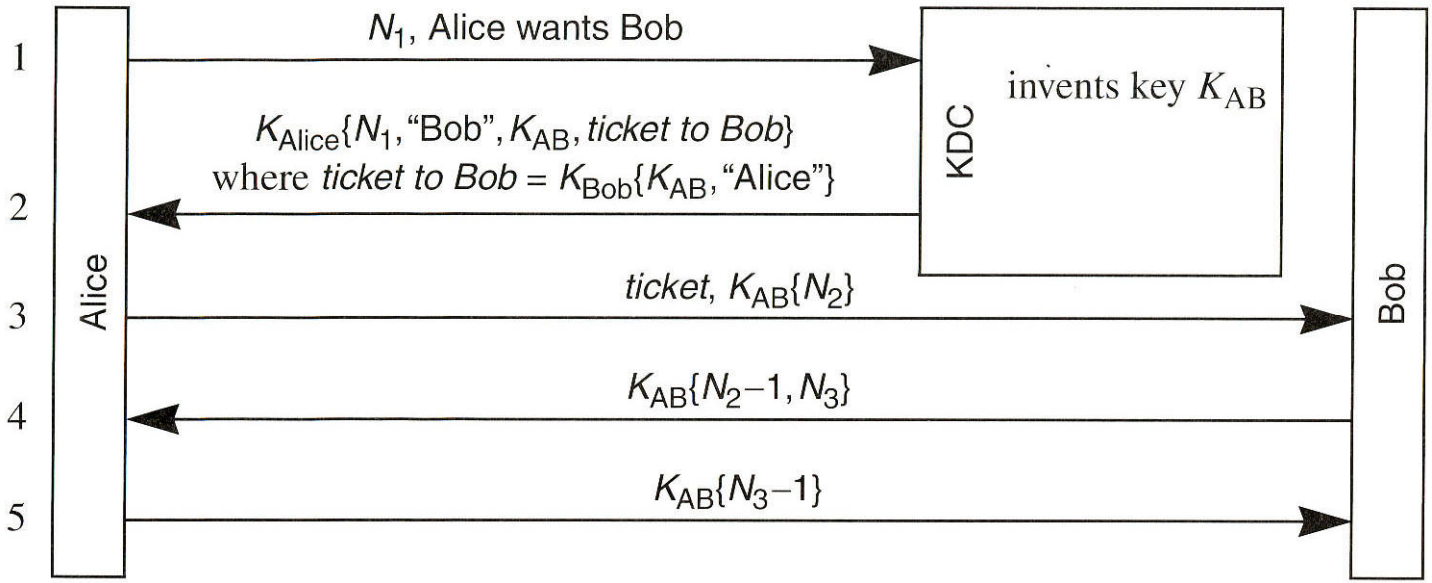


Protocol 11-18. Needham-Schroeder

1978 → bug découvert en 1981.

Si Trudy enregistre le message 2, et découvre *plus tard* la clé d'Alice, elle peut se faire passer pour Alice auprès de Bob (messages 3 et 5), même si Alice a changé de clé entre-temps !

Needham-Schroeder (2)

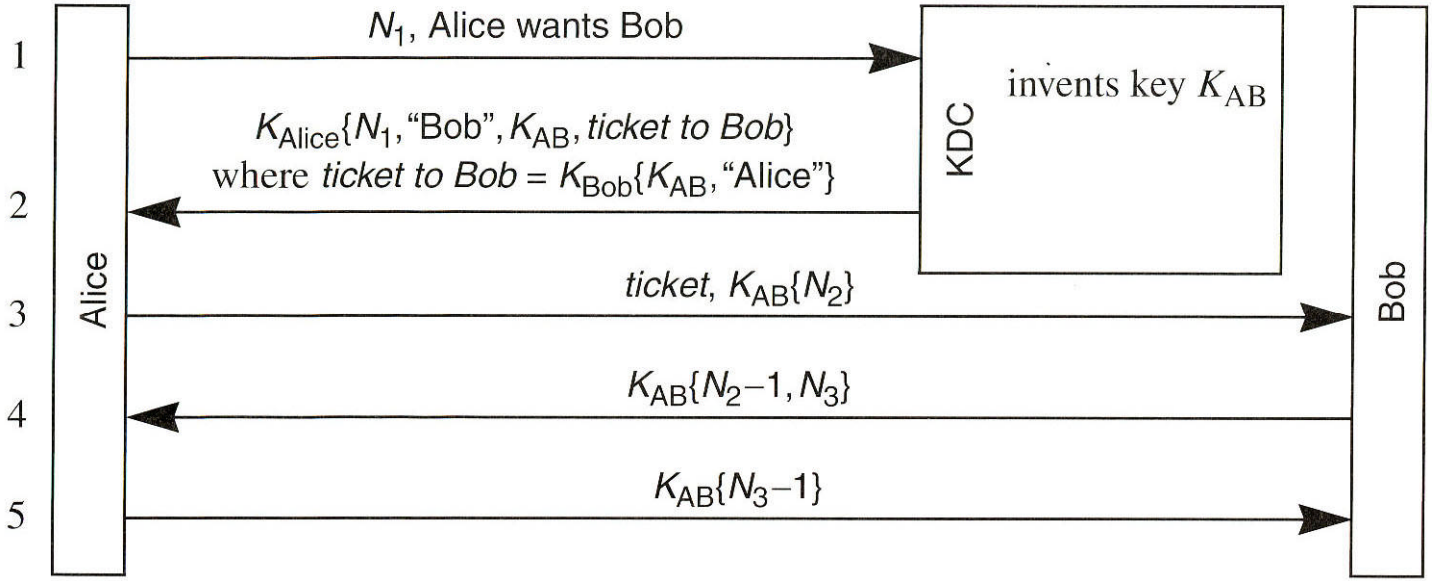


Protocol 11-18. Needham-Schroeder

Rôle de N_1 (**Nonce** = Number Once) : sinon Trudy

- conserve le message 2, jusqu'à découverte de la clé de Bob
- retourne ce message à Alice, à la place du KDC (?)
- prend la place de Bob (?) (messages 3 à 5), même si Bob a changé de clé entre-temps

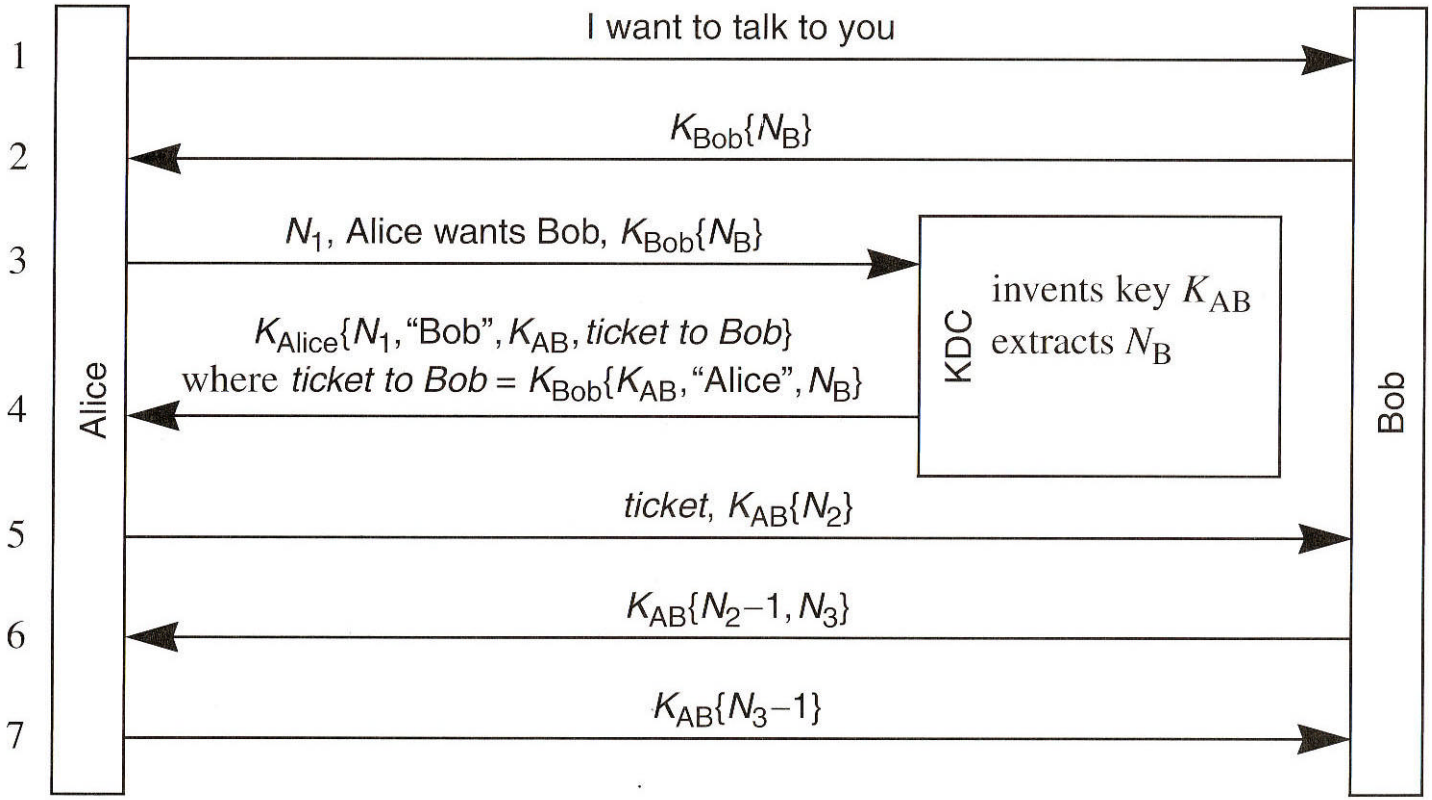
Needham-Schroeder (3)



Protocol 11-18. Needham-Schroeder

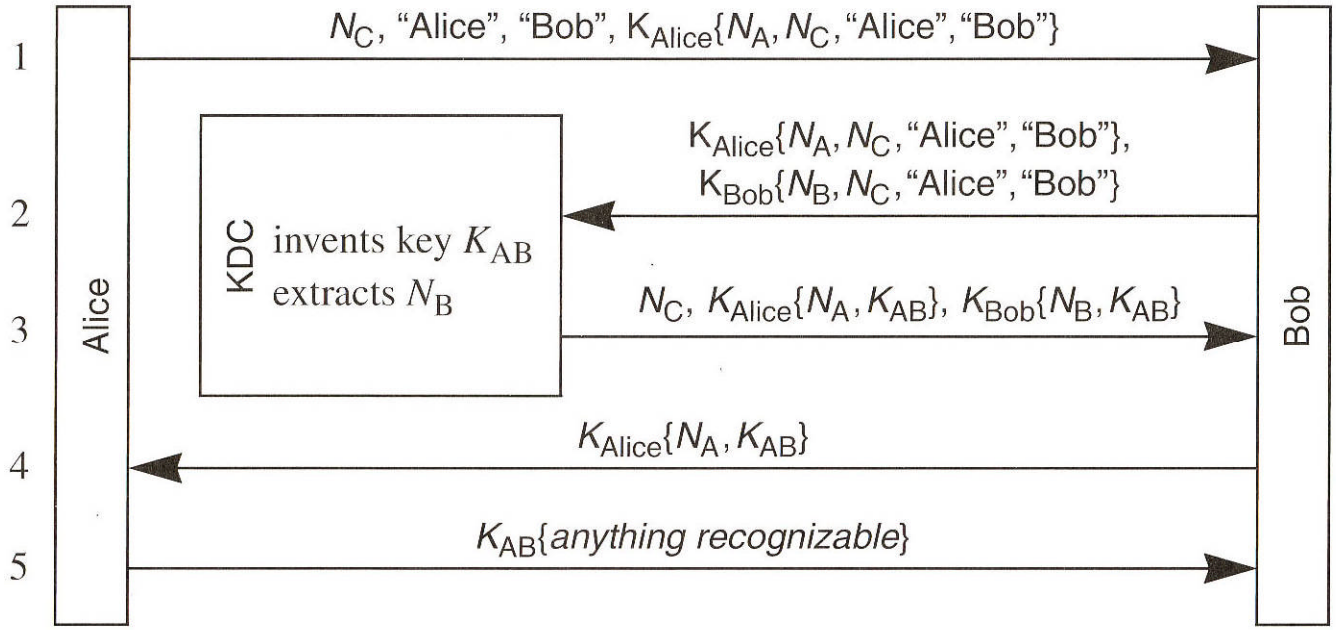
Attention : attaque par réflexion si K{x,y} = K{x}, K{y}

Expanded Needham-Schroeder



Protocol 11-19. Expanded Needham-Schroeder

Otway-Rees



Protocol 11-20. Otway-Rees

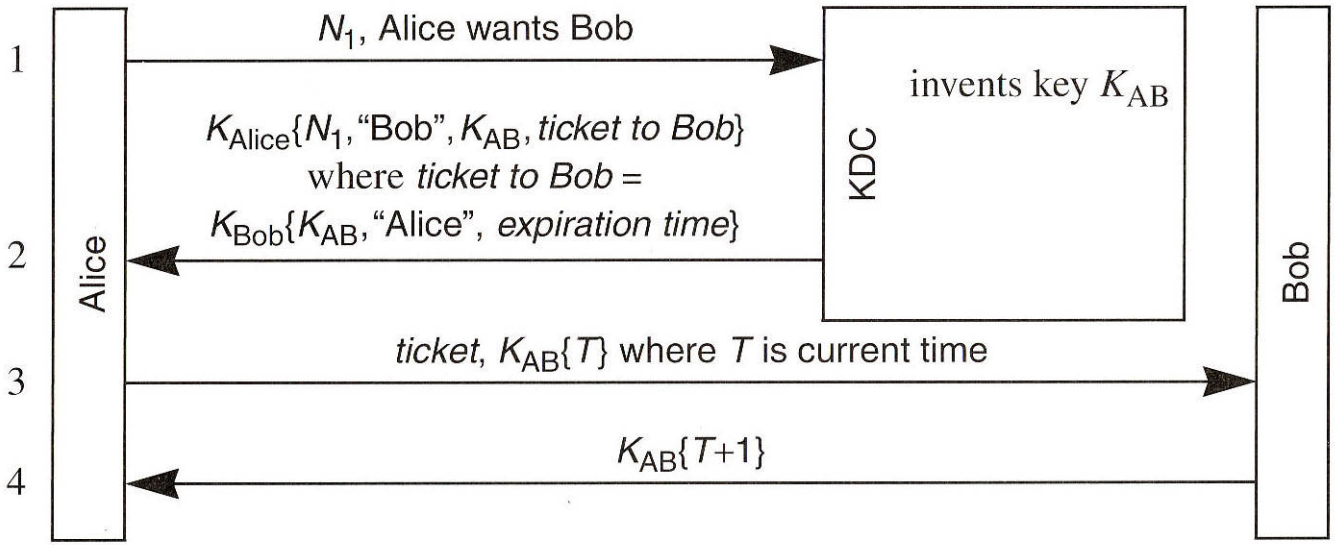
1987

Otway-Rees (2)

- Le KDC vérifie que N_C est le même dans les deux parties du message 2, ce qui authentifie Bob
- Le message 4 assure à Alice que le KDC est légitime (il a déchiffré N_A)
- Le message 4 assure à Alice que Bob est légitime (vérifié par le KDC)
- Le message 3 assure à Bob que le KDC est légitime (il a déchiffré N_B)

Exercice : si N_C est prévisible, Trudy peut se faire passer pour Bob.

Kerberos (principe)



Protocol 11-21. Kerberos