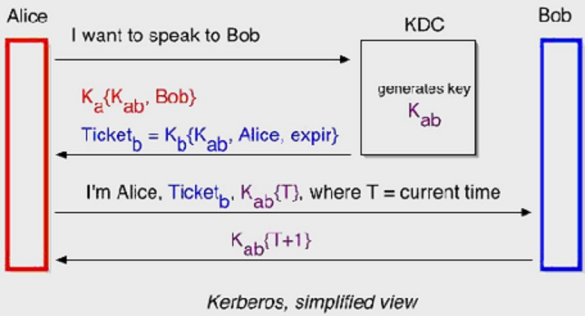
ierie des Protocoles et Logiciels Sécurisés

Conception de protocoles cryptographiques

Version 1



Kerberos, simplified view

Yacine Challal

18/12/2014

Table des matières

Conception de protocoles cryptographiques

I

A. Protocoles cryptographiques : généralités

Définition : Protocole cryptographique

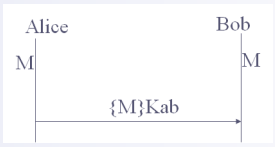
Un protocole cryptographique est une séquence d'étapes spécifiant les actions respectives devant être réalisée par deux entités (ou plus) pour remplir un objectif de sécurité donné (exemple: Diffie Hellman pour l'échange de clé).

Remarque : Protocole vs. algorithme cryptographique

Un protocole cryptographique utilise des algorithmes de cryptographie.

Attention

Les algorithmes de cryptographie ne suffisent pas pour garantir le secret ou l'authenticité d'un message. Considérons l'exemple illustré par la figure suivante :



L'usage d'algorithme cryptographique n'est pas synonyme de sécurité

Pour que M reste confidentiel, il faut que Kab reste secret entre Alice et Bob.

Comment garantir cette hypothèse ?

C'est le rôle d'un protocole cryptographique.

Dans cet exemple, on voit bien que l'usage d'un algorithme de chiffrement symétrique ne signifie pas que le service de confidentialité est acquis.

Typologie des protocoles cryptographiques

Les protocoles cryptographiques peuvent être classés en plusieurs catégories :

- Protocoles d'échange de clefs (key exchange, key agreement)
  - Création d'un secret partagé (exemple: protocole de Diffie Hellman)

- Protocoles d'authentification
  - Authentification de l'origine des données
  - Authentification de l'entité homologue (ou identification)
- Protocoles combinant authentification et échange de clés
- Autres protocoles
  - Vote électronique
  - Partage de clé de groupe (group key agreement)
  - Horodatage et estampillage
  - ...

Syntaxe : Notation

Nous allons adopter la notation suivante pour la spécification de protocoles cryptographiques :

- Kab désignera une clé secrète (utilisée dans un algorithme symétrique) partagée entre a et b
- PKa désignera une clé publique de a(utilisée dans un algorithme asymétrique)
- SKa désignera une clé privée de a (utilisée dans un algorithme asymétrique)
- Si m désigne un message
  - {m}Kab désignera m chiffré avec Kab
  - {m}PKa désignera m chiffré avec PKa
  - {m}SKa désignera m chiffré avec SKa
  - H(m) désignera un condensé calculé sur m avec la fonction de hachage h
  - H_k(m) désignera un MAC calculé sur m avec la fonction de hachage H paramétrée avec la clé k, H_k.
- Protocole := Message ... Message
- Message := M_n, Entité -> Entité : Data
- Entité := A|B|S|C/A|C/B|C/S (A: Alice, B: Bob, C: Attaquant, S: Serveur)

  Data :=

  A|B|S

  K tel que K dans {Ksa,Ksb,Kab,PKa,PKb,PKs,SKa,SKb,SKs}

  ra|rb (Nonces générés respectivement par a et b)

  ta|tb (estampilles générées respectivement par a et b)

  Data.Data (concaténation)

  {Data}k donnée chiffrée avec k

  Data* (donnée optionnelle)
- Nonce (oNly ONCE) (ou aléas) : nombre unique et imprévisible
- Estampille : marqueur de temps, sert à calculer la fraîcheur

Hypothèses sur l'attaquant

- C peut écouter les messages échangés
- C peut bloquer les messages
- C peut rediriger les messages
- C peut enregistrer les messages
- C peut rejouer les messages
- C ne sait pas déchiffrer (sans avoir la clé) dans le temps d'une session

B. Protocoles d'échange de clé

Echange de clé en utilisant un système asymétrique

M1: b->a : PKb

M2: a->b : {Kab}PKb

Attention : Attaque man in the middle

Ne garantie pas l'authentification: attaque « man in the middle »

M1: b->c/a: PKb

M2: c/a->b: {Kcb}PKb

Protocoles d'échange de clé : Otway-Rees

M1: a->b: m.a.b.{na.m.a.b}Ksa (m identifiant de transaction)

M2: b->s: m.a.b.{na.m.a.b}Ksa.{nb.m.a.b}Ksb

M3: s->b: m.{na.Kab}Ksa.{nb.Kab}Ksb

M4: b->a: m.{na.Kab}Ksa

Attention : Attaque par confusion de type

Supposons que m fait 32 bits, a et b 16 bits, et Kab 64 bits. Il suffit que C rejoue la partie chiffrée de M1

M1: a->c/b: m.a.b.{na.m.a.b}Ksa

M4': c/b->a: m.{na.m.a.b}Ksa

Conclusion a accepte m.a.b comme nouvelle clé (envoyée en claire !!!)

C. Protocoles d'authentification de l'origine

Objectif

b doit être assuré que le message a été créé par a. On peut assurer l'authentification de l'origie de deux manières :

- Utilisation d'un algorithme asymétrique de signature

  M1: a->b : a.{m}SKa
- Utilisation d'un MAC

  M1: a->b : a.m.H_k(m)

Attention : Attaque de rejeu

Ce protocole ne protège pas contre le rejeu. Pour cela, on peut ajouter un paramètre de temps pour assurer la fraîcheur : estampille, numéro de séquence, nombre aléatoire envoyé dans un message précédent.

D. Protocoles d'authentification d'entité

Objectif

A doit être authentifié auprès de b à la fin du déroulement du protocole

Types d'identification

- Authentification faible : mots de passe fixes, mots de passe à usage unique
- Authentification forte : protocoles défi-réponse basés sur la cryptographie symétrique ou asymétrique

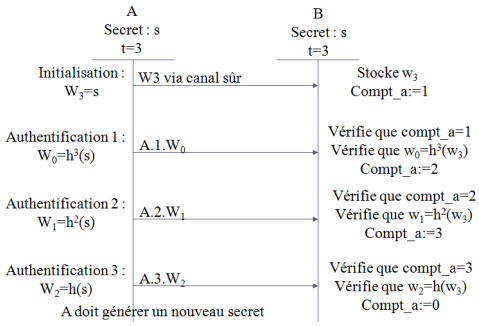
Attention : Vulnérabilité d'authentification d'entité faible par mot de passe fixe

- Possibilité d'intercepter le mot de passe
- Mots de passes stockés dans un fichier protégé en lecture et écriture, ou
- Utilisation de fonction à sens unique avant le stockage (exemple: login sous unix)
- Attaque par dictionnaire

Méthode : Mots de passe à usage unique : vers l'authentification forte

- Protège contre les interceptions de message
- Basée sur une liste pré-partagée
- Basée sur un incrément
- Basée sur une fonction à sens unique : exemple protocole de Lamport
  - A veut s'identifier à B
  - A possède un secret w,
  - h fonction à sens unique connue par A et B
  - t : constante définissant le nombre d'authentifications autorisées (exemple si t=100, après 100 authentification, A génère un nouveau secret w)
  - On définit par récurrence : h0(s)=s, $h^i(s)=h^{i-1}(h(s))$ pour 1<=i<=t
  - On note $w_i=h^{t-i}(s)$, i-ième mot de passe
  - A transfère à B par un canal sûr $w_0=h^t(s)$,
  - B initialise son compteur compt_a à 1
  - À la ième identification : A calcule w_i, et envoie à B M1 : A->B : a.i.w_i, B vérifie que compt_a=i et que $h(w_i)=w_{i-1}$

La figure suivante illustre le fonctionnement de ce protocole :



One Time Password

E. Authentification forte par défi réponse

Principe

Une entité (le prouveur) prouve son identité à une autre entité (le vérificateur) en démontrant au vérificateur qu'il possède un secret sans révéler ce secret grâce à

une réponse à un challenge variant dans le temps

Définition : Challenge

Utilisation de nonce garantissant une notion de fraîcheur / unicité

- Nombre pseudo-aléatoire (noté r dans la suite)
  - Nombre non prévisible par un attaquant
- Numéro de séquence (noté n dans la suite)
  - Nécessite de mémoriser les numéros de séquence déjà utilisés
- Estampille (horodateur, time stamp) + horloges synchronisées (noté t)
  - Les horloges doivent être sécurisées
- Combinaison de nonces : nombre aléatoire concaténé à un numéro de séquence ou à une estampille
  - Permet de garantir qu'un nombre aléatoire n'a pas été dupliqué

Remarque : Critères de classification

- Systèmes cryptographiques à clés publiques ou symétriques
- Nombre de messages
- Authentification unilatérale ou mutuelle

Remarque : Types de protocoles par défi-réponse

- Protocoles utilisant une clé secrète (chiffrement symétrique ou MAC)
- Protocoles basés sur un chiffrement avec clé publique
- Protocoles basés sur la vérification d'une signature

1. Authentification forte par défi-réponse basée sur une clé partagée

Variante 1

- M1: a->b : I'm A
- M2: b->a : nb
- M3: a->b : {nb}Kab

L'authentification est non mutuelle

Variante 2

- M1: a->b : I'm A
- M2: b->a : {nb}Kab
- M3: a->b : nb

Variante 3 : une passe

- M1: a->b : a.{ta}Kab

Nécessite que A et B aient des horloges synchronisées. Bob déchiffre le message et s'assure que ta est dans un intervalle de temps raisonnable.

Remarque : Inconvénient d'usage de clés partagées

Si la base de donnée du côté serveur (B) est corrompue, Alice peut être usurpée par un intrus.

Solution: usage des clés publiques

2. Authentification forte par défi-réponse à base de clés publiques

Variante 1

- M1: a->b : I'm A
- M2: b->a : nb
- M3: a->b : {nb}SKa (A signe le nonce nb avec sa clé privée)

Variante 2

- M1: a->b : I'm A
- M2: b->a : {nb}PKa (b chiffre le nonce nb avec la clé publique de A)
- M3: a->b : nb

Attention : Limitation

L'authentification est unilatérale => un intrus peut faire signer à A un message, ou intercepter un message qui lui est destiné et le lui faire déchiffrer !!!

3. Authentification forte par défi réponse : Authentification mutuelle à base de clé partagée

Version 1

- M1: a->b : I'm A
- M2: b->a : nb
- M3: a->b : {nb}Kab
- M4: a->b : na
- M5: b->a : {na}Kab

Version 2 : réduire le nombre de messages

- M1: a->b : I'm A . na
- M2: b->a : nb.{na}Kab
- M3: a->b : {nb}Kab

Attention : Vulnérabilité : Attaque réflexive avec entrelacement de sessions

Session 1	Session 2
M1: c/a->b : I'm A . na M2: b->c/a : nb.{na}Kab	
	M1: c/a->b : I'm A . nb M2: b->c/a : nb'.{nb}Kab
M3: c/a->b : {nb}Kab	

Entrelacement de sessions

Solution:

- M1: a->b : I'm A . na
- M2: b->a : nb.{na.**b**}Kab
- M3: a->b : {nb.**a**}Kab

Version 3 : Réduire le nombre de messages en utilisant des estampilles à la place de nonces

Nécessite la synchronisation des horloges de A et B

- M1: a->b : I'm A . {a.ta}Kab
- M2 : b->a : {b.ta}Kab

4. Authentification forte par défi réponse : Authentification mutuelle à base de clé publique

Version 1

- M1: a->b : I'm A . {na}PKb
- M2: b->a : na.{nb}PKa
- M3: a->b : nb

Version 2

- M1: a->b : I'm A . na
- M2: b->a : nb . {na}SKb
- M3: a->b : {nb}Ska

Remarque

Comment A connaît la clé publique de B ? Ces protocoles sont vulnérables aux attaques "man in the middle". Pour résoudre le problème, il faut introduire la certification des clés publiques par une autorité de confiance.

F. KERBEROS : Etude de cas

Définition : Kerberos

Développé au MIT, Kerberos est un service basé sur la cryptographie symétrique pour assurer l'authentification dans les réseaux ouverts. L'authentification est basée sur une tierce partie de confiance, le KDC: Key Distributor Center.

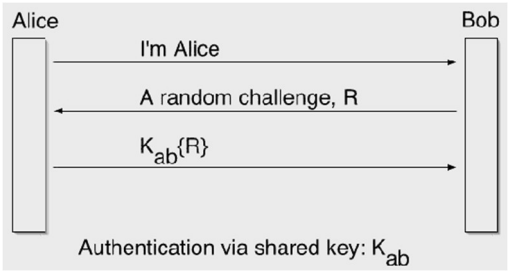
Avantages de Kerberos

- Un système d'authentification standardisé
- Largement adopté par les systèmes d'exploitation
- Pas de transmission de mots de passe dans le réseau
- Un seul mot de passe à se rappeler

Protocole d'authentification simple

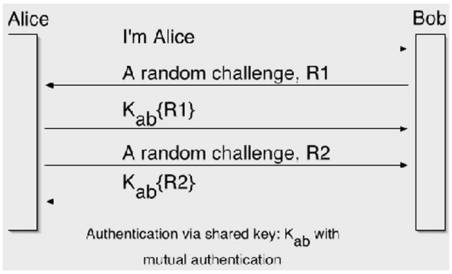
- Alice: Client initiateur de la communication
- Bob: Serveur
- Alice veut accéder au service de Bob

La figure suivante illustre un protocole d'authentification simple :



Protocole d'authentification simple

La protocole suivant illustre une variante d'un protocole simple d'authentification mutuelle :



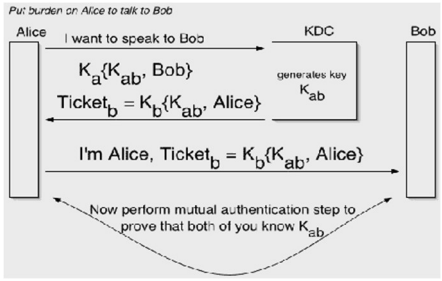
Protocole simple d'authentification mutuelle

Attention : Problèmes avec ce schéma d'authentification

- Ne s'adapte pas au facteur d'échelle: Avec n clients et m services, il va falloir distribuer a priori n*m clés symétriques

Complément : Amélioration possible: Partager une clé entre chaque service et client avec une tierce partie de confiance

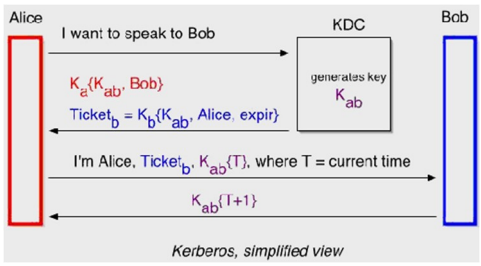
La tierce partie de confiance joue le rôle d'intermédiaire dans le processus d'authentification ; le KDC: Key Distribution Center. Chaque client et serveur partage une clé secrète avec le KDC. Le KDC génère une clé de session et la distribue aux parties communicantes confidentiellement. Les parties communicantes prouvent qu'elles connaissent la clé de session. Ce qui donne le schamé d'authentification suivant :



Version simplifiée de Kerberos

Remarque : Kerberos utilise des Timestamps

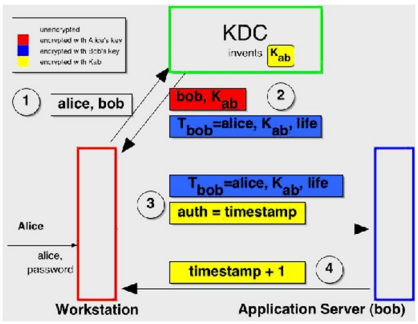
Kerberos utilise des timestamps comme nonces dans la phase d'authentification mutuelle. Donc, Kerberos nécessite une synchronisation raisonnable des horloges des parties communicantes. ce qui donne le schéma suivant :



Kerberos simplifié avec timestamps

Kerberos détaillé

Chaque client et service partage une clé avec le KDC. Tout le monde fait confiance au KDC. La clé du client est dérivée d'un mot de passe à qui on applique une fonction de hashage. La clé du service est un grand nombre aléatoire. La figure suivante illustre les principales étapes du protocole Kerberos :

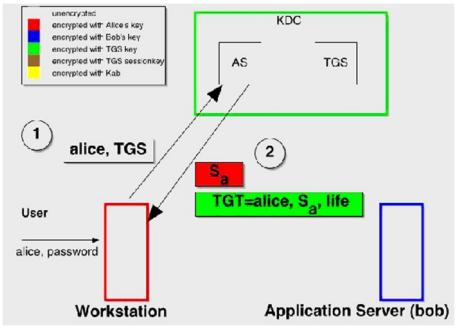


Kerberos avec authentification de l'utilisateur par mot de passe

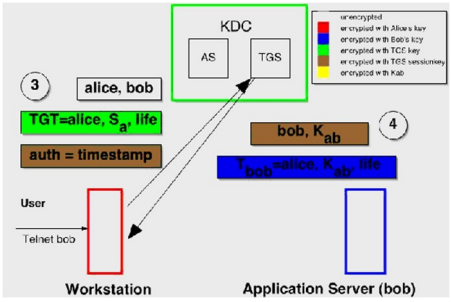
Kerberos avec TGS

- Ticket Granting Service (TGS) : Permet à des utilisateurs d'obtenir des tickets pour des services. TGS est localisé avec le KDC
- Ticket Granting Ticket (TGT) : Un ticket pour accéder au TGS afin d'obtenir des tickets de service

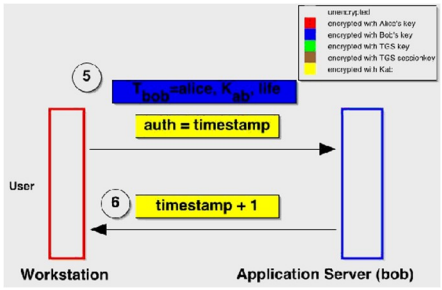
Ainsi la version détaillée de Kerberos se déroule selon les étapes suivantes :



Kerberos détaillé

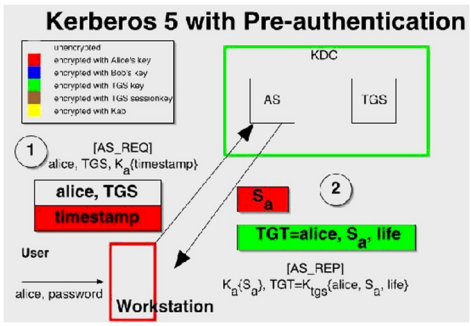


Kerberos détaillé



Kerberso détaillé

Kerberos v5 avec pré_authentification



Kerberos v5

Résumé

- Méthode d'authentification:
  - L'utilisateur introduit son mot de passe sur la machine locale
  - Authentification par le KDC une fois dans journée
  - Aucun mot de passe ne traverse le réseau
- Single sign-on (grasse à TGS)
  - KDC vous donne un TGT valable généralement pour la journée
  - Ce ticket peut être utilisé pour obtenir d'autres tickets de service qui seront utilisés pour accéder à ces services, une fois présentés avec un authentificateur (timestamp chiffré avec la clé de session)

Série d'exercices sur la conception de protocoles cryptographiques

II

A. Objectif de sécurité

Soit le protocole cryptographique suivant
M1 : B ==> A : B.PKb
M2 : A ==> B : {m}PKb
L'objectif de ce protocole est :

- ☐ Authentification de A auprès de B
- ☐ Confidentialité de m
- ☐ Authentification de B auprès de A

B. Man in the Middle

Soit le protocole cryptographique suivant
M1 : B ==> A : B.PKb
M2 : A ==> B : {m}PKb
Ce protocole est vulnérable à une attaque de type « man in the middle ». Un intrus

"I" peut attaquer ce protocole comme suit :

☐	I intercepte M1 I bloque M1 I remplace M1 par : M1 : I ==> B : A.PKi I intercepte M2 I bloque M2 I remplace M2 par : M2 : I ==> A : {m}PKa
☐	I intercepte M1 I bloque M1 I remplace M1 par : M1 : I ==> A : B.PKi I intercepte M2 I bloque M2 I remplace M2 par : M2 : I ==> B : {m}PKb
☐	I intercepte M1 I bloque M1 I remplace M1 par : M1 : I ==> A : B.SKb I intercepte M2 I bloque M2 I remplace M2 par : M2 : I ==> B : {m}PKb

C. Certification

Soit le protocole cryptographique suivant
M1 : B ==> A : B.PKb
M2 : A ==> B : {m}PKb
En supposant que S est une autorité de certification, ce protocole peut atteindre son objectif de sécurité en remplaçant M1 par :

☐	B==>A : {B.PKb}PKs
☐	B==>A : {B.SKb}SKs
☐	B==>A : {B.SKb}PKs
☐	B==>A : {B.PKb}SKs

D. Objectif de sécurité

Supposons que Kab est un secret partagé entre A et B.
Soit le protocole suivant :
M1 : A==> B : Hello.A
M2 : B==> A : Nb
M3 : A==> B : {Nb}Kab
Quel est l'objectif de sécurité de ce protocole ?

☐	confidentialité de Nb
☐	confidentialité de Kab
☐	authentification mutuelle
☐	authentification de A auprès de B
☐	authentification de B auprès de A

E. Equivalence

Supposons que Kab est un secret partagé entre A et B.
Soit le protocole suivant :
M1 : A==> B : Hello.A
M2 : B==> A : Nb
M3 : A==> B : {Nb}Kab
Quels sont les protocoles équivalents, au protocole précédent, dans l'objectif de sécurité ?

☐	M1 : A==> B : Hello.A M2 : B==> A : {Nb}Kab M3 : A==> B : Nb
☐	M1 : A==> B : Hello.A M2 : B==> A : {Nb}PKb M3 : A==> B : Nb
☐	M1 : A==> B : Hello.A M2 : B==> A : Nb M3 : A==> B : {Nb}SKa

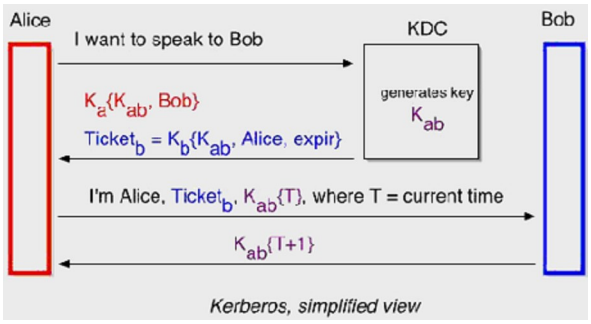
F. Kerberos

Le protocole Kerberos :

☐	a pour objectif d'assurer une authentification mutuelle
☐	est basé sur une tierce partie de confiance
☐	nécessite une synchronisation des horloges du client et serveur.

G. Fonctionnement de Kerberos

Soit le schéma simplifié de Kerberos de la figure suivante :



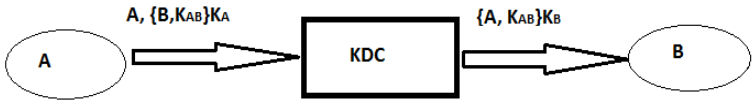
Kerberos simplifié

☐	Kb est connue par Alice
☐	Kb est une clé symétrique partagée entre le KDC est Bob uniquement
☐	Kb est égale à Kab-Ka
☐	Kb est utilisée par Alice pour déchiffrer Kab à partir du Ticket_b
☐	le message qui permet d'authentifier Alice auprès de Bob est M1
☐	le message qui permet d'authentifier Alice auprès de Bob est M3

H. Protocole d'authentification reposant sur une tierce partie

Antoine et al. 2010

La figure suivante représente un protocole d'authentification utilisant un centre de distribution de clefs (KDC). A l'issu de l'échange de la clef K_{AB}, A peut envoyer des messages chiffrés avec K_{AB} à B.



Protocole d'authentification via KDC

Question 1

Expliquer pourquoi un pirate ne peut pas se faire passer pour A auprès de KDC.

Question 2

Expliquer pourquoi B est certain que le message provient du KDC.

Question 3

A quelle attaque ce protocole ne résiste-t-il pas ?

Indice :

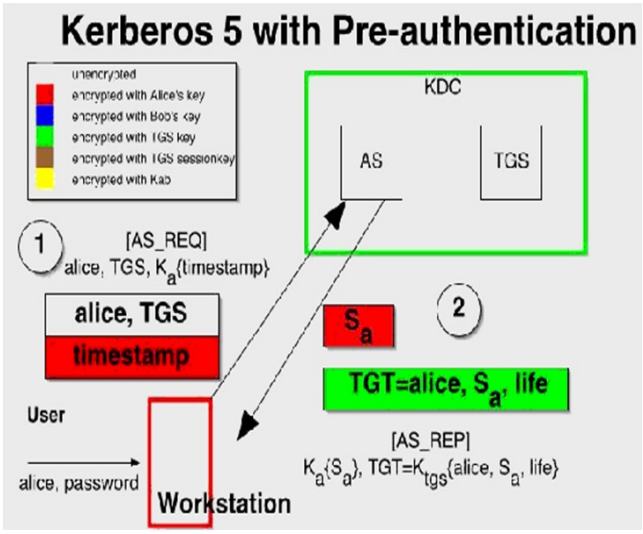
Imaginer qu'un pirate I ait effectué un travail pour A. Après avoir échangé une clef de session via le KDC, A envoie un message à son banquier B pour lui demander de verser la rétribution sur le compte de I. Que faire à la place de I pour augmenter ses gains ?

Question 4

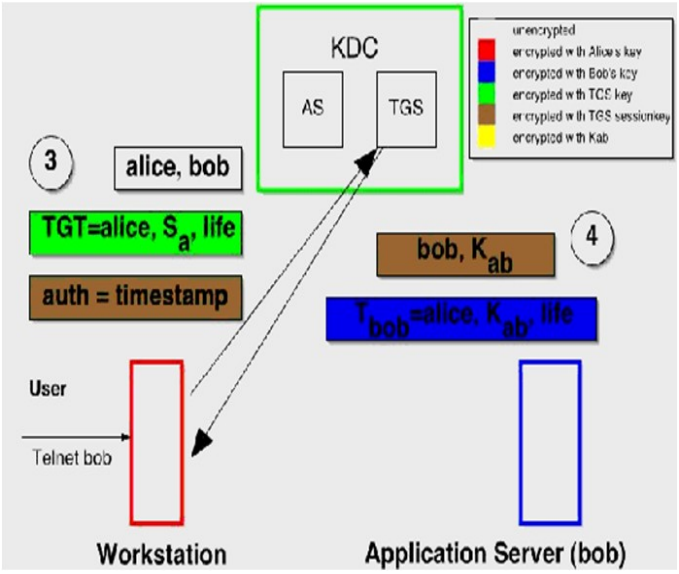
Comment améliorer le protocole sans augmenter le nombre d'échanges pour déjouer ce type d'attaque ?

I. Analyse du Système d'Authentification Kerberos

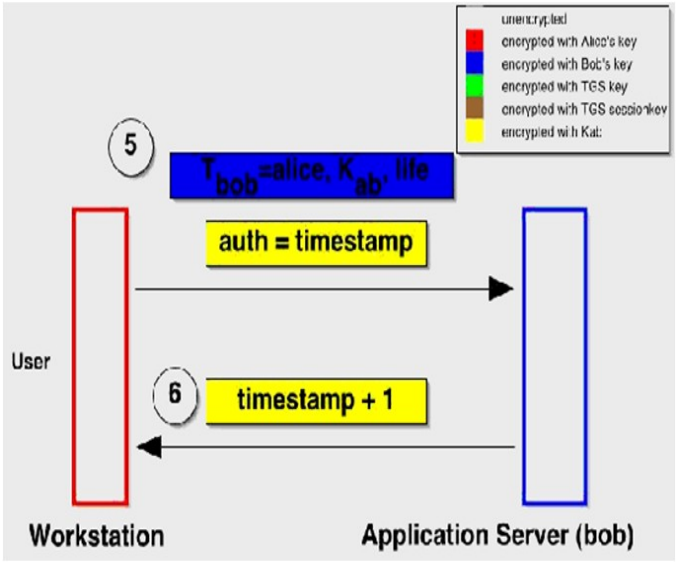
Rappelons le fonctionnement de Kerberos :



Kerberos Steps 1-2



Kerberos Steps 3-4



Kerberos Steps 5-6

Antoine et al. 2010

On s'intéresse à la possibilité d'usurper le ticket Kerberos v5 d'un client. On considère un pirate qui écoute le réseau et voit passer le ticket que le TGS envoie à un client. Le pirate connaît aussi l'identité du client à qui est destiné le ticket.

Question 1

Qu'est ce qui empêche le pirate d'utiliser le ticket pour obtenir un service à la place du client légitime ?

L'une des caractéristqiues de Kerberos est qu'un utilisateur n'a pas besoin de s'authentifier auprès du KDC chaque fois qu'il désire accéder à un service.

Question 2

Pourquoi ? Donner un avantage et un inconvénient de cette caractéristique (en ce qui concerne la sécurité) et les justifier.

Dans le procédé d'authentification Kerberos, la clef symétrique utilisée par le client et le serveur d'authentification est le haché du mot de passe du client. le serveur possède donc une liste de hachés des mots de passe de tous les clients.

Question 3

Si on arrive à voler cette liste, peut-on s'authentifier à la place d'un client ? Si non, pourquoi ?

Si oui, que pourrait-on faire pour éliminer ce problème ?

Lorsque'un client présente un ticket à un serveur, il doit y joindre un authentificateur pour prouver qu'il en est le détenteur légitime.

Question 4

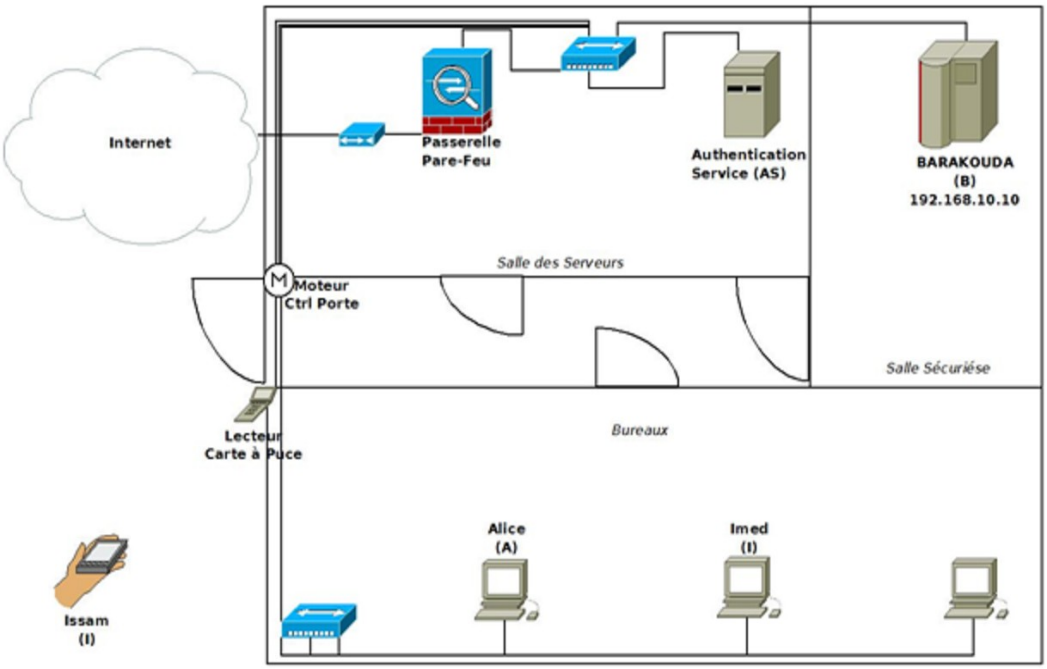
Comment peut-on vérifier que l'authentificateur a bien été créé par le détenteur légitime du ticket ?

Question 5

Qu'est ce qui empêche un pirate de copier un ticket et un authentificateur et de les réutiliser à son avantage ?

J. Sécurité du réseau de l'organisme d'études criminalistiques

L'organisme d'études criminalistiques possède un ordinateur BARAKOUDA (B) stockant des informations sensibles comme des empreintes digitales, des identités liées à des infractions et/ou crimes. Cet ordinateur est alors installé dans une pièce et bâtiment sécurisés comme illustré sur la figure suivante :



Architecture du réseau de l'organisme d'études criminalistiques

L'accès aux données de cet ordinateur nécessite les contrôles suivants :

1. Authentification par carte à puce pour l'accès au bâtiment
2. Authentification auprès d'un service d'authentification (AS) pour autoriser la connexion à BARAKOUDA

Typiquement, une employée Amina (A) dispose d'une carte à puce dans laquelle est stockée sa clé privée chiffrée avec une clé issue d'un code PIN que seule Amina connaît. Pour accéder au bâtiment, Amina insère sa carte à puce dans le lecteur à côté de l'entrée. Le lecteur lui demande son code PIN pour pouvoir lire et déchiffrer la clé privée. Un processus d'authentification « challenge/response » s'enclenche alors entre l'AS et le lecteur de carte. Si l'authentification réussit, l'AS envoie un signal pour l'ouverture de la porte.

Question 1

Compléter le protocole challenge/response suivant permettant à l'AS d'authentifier Amina via la carte à puce et d'ouvrir la porte :

```
Lecteur Carte-->AS : Hello, A
.....................................................
.
Lecteur de Carte-->AS : {A, Na}PK_AS
AS : Ouvrir la porte
```

Une fois installée sur un poste de travail, Amina peut se connecter à BARAKOUDA après une authentification auprès de l'AS comme suit :

1. A-->AS : message authentifiant A auprès de AS | B
2. AS-->A : {Validité,B,ListeServices}$K_{B,AS}$ | {A,$K_{A,B}$,Validité}$K_{B,AS}$ | {$K_{A,B}$}$K_{A,AS}$
3. A-->B :{Validité,B,ListeServices}$K_{B,AS}$ | {A,$K_{A,B}$,Validité}$K_{B,AS}$ | {A, Service}$K_{A,B}$

$K_{B,AS}$ et $K_{A,AS}$ sont pré-partagées entre AS et B et A respectivement

Question 2

Donner le message envoyé dans l'échange (1) du protocole, en précisant l'hypothèse nécessaire pour que ce seul message suffise pour authentifier A auprès de AS.

Question 3

Expliquer comment BARAKOUDA peut vérifier qu'Alice est bien autorisée à accéder au service demandé ?

On s'intéresse à un employé Imed (I) qui a accès au bâtiment mais qui ne dispose pas des droits d'accès nécessaires pour accéder au service des empreintes digitales stockées sur BARAKOUDA.

Question 4

Expliquer comment Imed peut exploiter le protocole d'authentification décrit ci-dessus pour réussir à se connecter à BARAKOUDA et avoir accès au service d'empreintes digitales.

Question 5

Comment corriger le protocole pour éviter cette attaque.

Il est également possible de se connecter à BARAKOUDA de l'extérieur par SSH via la passerelle du réseau de l'organisme. Pour cela, l'administrateur réseau configure le pare-feu sans mémoire pour n'autoriser que les connexions SSH sur le port 22 de BARAKOUDA. SSH est configuré pour authentifier les clients par challenge/response en utilisant une paire de clés privée/publique du client identique à la paire de clés stockée sur la carte à puce de l'employé.

Question 6

Ecrire les règles de filtrage du pare-feu qui permet cette configuration.

Source		Destination		Flags	Action
Adresse	Port	Adresse	Port	SYN=1 & ACK=0	
.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....

Table de filtrage

Issam (I) est un intrus qui n'est pas employé de l'organisme et qui souhaite se connecter à BARAKADOU de l'extérieur via un tunnel SSH. La nuit, il réussit à remplacer le lecteur de carte du portail du bâtiment par un faux lecteur relié à son smart-phone via une carte SIM 3G.

Question 7

Que peut récupérer Issam avec ce faux lecteur ?

Question 8

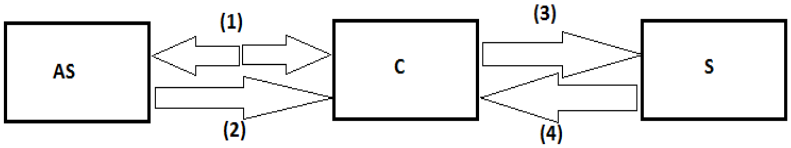
Est-ce que l'information récupérée par Issam dans la question précédente est suffisante pour ouvrir le tunnel SSH ? Si oui expliquer comment. Si non, expliquer pourquoi et proposer une solution à Issam.

K. Analyse d'un système d'authentification chez la Sarl. Amane

Antoine et al. 2010

La Sarl. Amane possède un réseau informatique interne qui permet de gérer la ligne de production. L'accès depuis les stations de travail aux serveurs intégrés sur les machines de la ligne de production nécessite une authentification sur un serveur centralisé.

Quand un employé,appelé dans la suite Client et noté C, veut accéder au serveur (noté S) contrôlant une machine, il doit passer le contrôle d'authentification exigé par le serveur centralisé (AS). Si cette authentification réussit, AS envoie des données à C, en particulier la liste des serveurs auxquels il est autorisé à se connecter. La figure suivante représente les échanges entre AS, C et S. On suppose qu'un pirate est capable d'écouter la communication.



Système d'authentification de Amane

1. C et AS effectuant un protocole d'authentification (que vous devrez concevoir dans la suite). A l'issu de ce protocole, on suppose qu'il se sont mis d'accord sur une clef de session $K_{C,AS}$.
2. Quand C est authentifié, AS choisit une clef aléatoire $K_{C,S}$ et envoie :
   $\{V,L\}K_{S,AS}||\{C,K_{C,S},V\}K_{S,AS}||\{K_{C,S}\}K_{C,AS}$
   à C, où V est une période de validité (suffisamment longue pour que le client ne doive effectuer l'authentification qu'une seule fois par jour) et L est une liste des serveurs auxquels le client est autorisé d'accéder.
3. Ensuite C envoie le message suivant à S :
   $\{V,L\}K_{S,AS}||\{C,K_{C,S},V\}K_{S,AS}||\{C,\text{requête}\}K_{C,S}$
4. A partir de $\{V,L\}K_{S,AS}$, le serveur S vérifie que le client est autorisé à accéder au service demandé dans la requête. Si c'est le cas, il exécute la requête et renvoie si nécessaire le résultat au client.

Question 1

Concevoir un protocole d'authentification qui pourraît être entre C et AS. Ce protocole ne doit pas nécessiter l'envoi de mots de passe ou de hachés de mots de passe en clair sur le canal.

Question 2

On considère dans la suite un pirate qui a pu être correctement authentifié par le serveur AS (par exemple, le pirate est un employé de la société Amane) mais qui n'a pas les droits d'accès pour un certain serveur S'.

Proposer une attaque telle que S' accepte la requête forgée par le pirate.

Question 3

On considère maintenant un pirate qui n'a pas pu être authentifié par le serveur AS. Expliquer quelle type d'attaque le pirate peut tenter et pourquoi une telle attaque est possible.

Question 4

Comment peut-on modifier le protocole pour éviter les attaques des deux questions précédentes ?

Question 5

Nous remarquons que C doit contacter AS à chaque fois qu'il veut interroger un nouveau serveur S. Pourquoi ? Comment peut-on modifier le protocole pour éviter ce problème, sans ajouter d'entité dans l'architecture.