Gestion des clés

Génération des clés

- Espaces de clés réduits
  - Codage restreint, caractères choisis, clés faibles, …
- Mauvais choix de clés
  - Lettre, mnémotechnique, …
  - → attaque par dictionnaire
- Clefs aléatoires
  - Générateurs, broyage de clé, acronyme, …
- Phrases mots de passe
  - Broyage de clé

Transfert de clé

- Physiquement
  - Rencontre, canal de transmission protégé, …
  - → rarement possible
- un tiers choisit et fournit la clé
- employer une clé précédente pour chiffrer une nouvelle clé
- si A et B ont des communications sures avec un tiers C, C peut relayer la clé entre A et B

Vérification de clés

- Origine
  - Rencontre physique
  - Annuaire
  - Tiers
- Moyens
  - Fonction de hachage
  - certificat

Stockage des clés

- Fichiers
- Support extérieur
  - Bande magnétique
  - Token, carte ROM
  - Carte à puce
- Ajout de code supplémentaire
  - PIN
  - Surchiffrement
  - …

Remarques

- Utilisation de distribution de clés décentralisée (KDC)
- Les hiérarchies de KDC sont exigées pour de grands réseaux, mais doivent se faire confiance entre elles
- La durée de vie des clés de session devrait être limitée pour une plus grande sécurité
- Contrôle de buts d'utilisation des clés

Distribution des clés

Clés symétriques

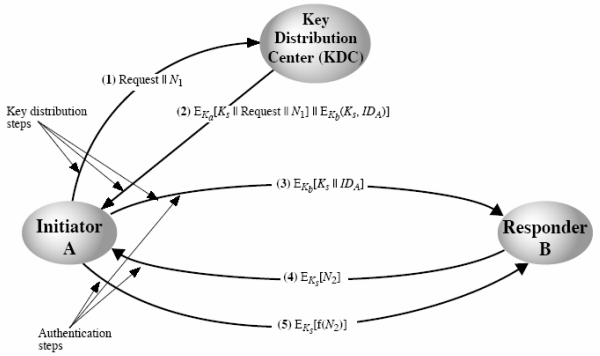
Clés symétriques

- Nécessité pour les deux usagers de partager une clé secrète commune
- Comment distribuer sûrement cette clé ?
- L'échec d'un système sur est souvent dû à une rupture dans le schéma de distribution des clés

Transfert de clé symétrique

- Physiquement
  - Rencontre, canal de transmission protégé, …
  - → rarement possible
- Un tiers de confiance :
  - Le tiers choisit et fournit la clé
- Une ancienne clé :
  - utilisée pour chiffrer une nouvelle clé
- Distribution automatique de clés
  - à la demande des utilisateurs
  - possible, mais on doit faire confiance au système

Scénario de distribution de clés symétriques



Distributions des clés

Clés publiques

Clés publiques

- Le chiffrement par clé publique permet de résoudre les problèmes de distribution de clés
- On peut employer soit :
  - Annonce publique
  - Annuaire publiquement disponible
  - Autorité de clé publique
  - Certificats de clé publique

Annonce Publique

- Distribution des clés publiques directement aux destinataires ou par broadcast à la communauté dans son ensemble
  - Exple : apposer les clefs de PGP aux emails ou les poster dans des newsgroups ou mailing-lists
- Risque : la contrefaçon
  - n'importe qui peut créer une clef en prétendant être quelqu'un d'autre et la publier
  - la mascarade peut continuer tant que la contrefaçon n'est pas découverte

Annuaire public

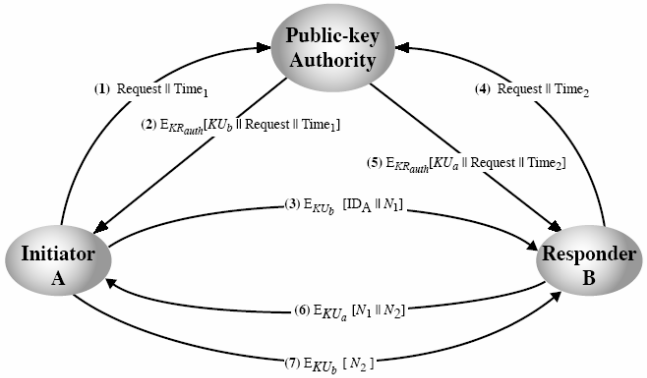
- Enregistrement des clés dans un annuaire public
- Nécessité de faire confiance à cet annuaire
- Propriétés :
  - Doit contenir les entrées {nom, publiс clef}
  - Possibilité de s'inscrire de manière sécurisée dans l'annuaire
  - Possibilité de remplacer la clef à tout moment
  - Publié périodiquement
  - Possibilité de consultation électronique
- Encore vulnérable au trifouillage et contrefaçon

Cet schéma est clairement plus sûr que les annonces publiques individuelles mais a toujours des vulnérabilités. Si un adversaire réussit à obtenir ou à calculer la clef privée de l'autorité d'annuaire, l'adversaire pourrait avec autorité fournir des clefs publiques contrefaites et plus tard personnifier n'importe quel participant et écouter clandestinement des messages envoyés à n'importe quel participant. Une autre manière d'atteindre le même but est pour que l'adversaire trifouille les enregistrements gardés par l'autorité.

Autorité de clé publique

- Renforcement du contrôle de la distribution des clefs à partir de l'annuaire
- Dispose des mêmes propriétés qu'un annuaire
- Chaque participant doit disposer d'une paire de clé
  - Publication de la clé publique dans l'annuaire
- Interaction avec l'autorité pour obtenir la clé publique du correspondant
  - exige l'accès en temps réel à l'annuaire quand les clefs sont nécessaires

Scénario de distribution de clés publiques



Certificat de clés publiques

- Les certificats permettent l'échange de clé sans accès en temps réel à l'autorité de clé publique
- Il lie une identité à une clef publique
  - habituellement avec d'autres informations telles que la période de validité, les droits d'utilisation etc..
- Son contenu est signé par la clé publique d'une entité de confiance (ou autorité de certification (CA))
- Il peut être vérifié par toute personne connaissant la clé publique de l'autorité de certification

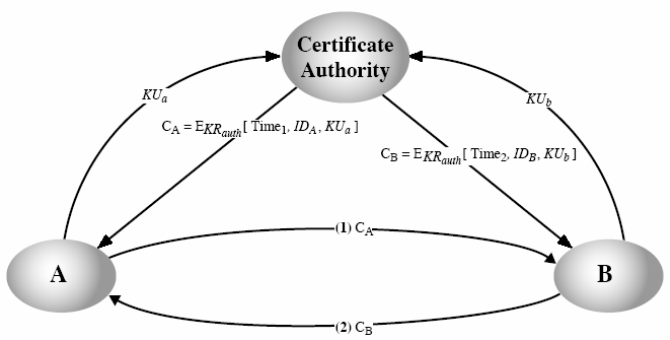
Une approche alternative (1978) est d'employer les certificats qui peuvent être employés par des participants pour échanger des clefs sans entrer en contact avec une autorité de public-clef, d'une façon fiable comme si les clefs avaient été obtenues directement à partir d'une autorité de public-clef.

Chaque certificat contient une clef publique et d'autres informations, est créé par un Certificate Authority, et est donné au participant disposant de la clef privée assortie. Un participant fournit l'information sur sa clé à un autre en transmettant son certificat. D'autres participants peuvent vérifier que le certificat a été créé par l'autorité.

Nous pouvons placer les conditions suivantes sur ce schéma : (1979)

1. N'importe quel participant peut lire un certificat pour déterminer le nom et la clé publique du propriétaire du certificat.
2. N'importe quel participant peut vérifier que le certificat provient du Certificate Authority et n'est pas contrefait.
3. Seul le Certificate Authority peut créer et mettre à jour des certificats.
4. Tous les participants peuvent vérifier l'actualité des certicates (Denning – 1983)

Échange de certificats de clés publiques



Distribution des clés

Certificats

19

Le problème des certificats numériques est à l'opposé de celui de la signature électronique : si vous commandez des Cds sur Internet, comment être sûr que vous envoyez bien votre numéro de carte bleue au commerçant, et non à un pirate qui aurait usurpé son identité et donné sa propre clé publique. Cette fois, c'est donc du Destinataire que l'on veut être sûr, et non de l'Expéditeur.

Comme dans la vie courante, on a recours à des certificats. Pour passer un examen, il vous faut prouver votre identité, ie fournir une carte d'identité, passeport ou permis de conduire. Un organisme supérieur (l'Etat) a signé ces certificats, s'assurant auparavant (par un acte de naissance,...) qu'il s'agit bien de vous.

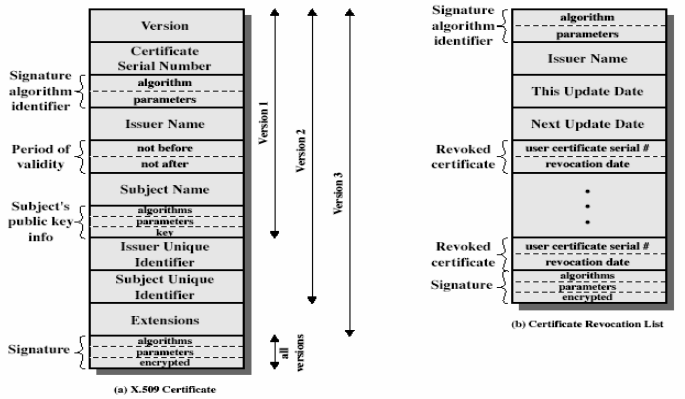
Les certificats numériques fonctionnent sur le même principe. Alice veut certifier que sa clé publique lui appartient. Elle envoie sa clé à un organisme de certification, ainsi que différentes informations la concernant (nom, email, etc...). Cet organisme vérifie les informations fournies par Alice, et ajoute au certificat son propre nom, une date limite de validité, et surtout une signature numérique. Cette signature est calculée de la façon suivante : à partir des informations du certificat, l'organisme calcule un résumé en appliquant une fonction de hachage connue, comme MD5. Puis il signe ce résumé en lui appliquant sa clé secrète.

Lorsque Bob veut envoyer son message à Alice, il télécharge le certificat de celle-ci sur un serveur de certificat (on parle de PKI, *Public Key Infrastructure*). Il calcule le résumé du certificat, puis applique la clé publique de l'organisme auteur du certificat à la signature électronique. Si cette quantité est égale au résumé, il est sûr qu'il a bien affaire à Alice.

Service d'authentification X.509

- Une partie de la norme de service d'annuaire X.500
  - serveurs distribués maintenant une base de données d'information
- Définit le cadre pour des services d'authentification
- L'annuaire peut stocker des certificats de clés publiques et les clés publiques des utilisateurs correspondant
- Signé par une autorité de certification
- Utilisé fréquemment dans les protocoles d'authentification
- Utilise la crypto à clé publique et les signatures digitales
  - algorithmes non imposés, mais RSA recommandé

X.509 Certificates



Obtention d'un certificat

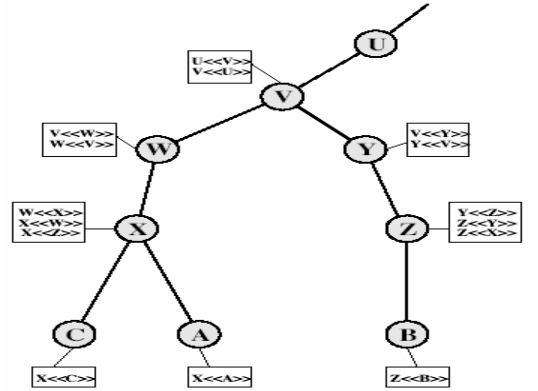
- N'importe quel utilisateur ayant accès au CA peut obtenir un certificat de celui-ci
- Seul le CA peut modifier un certificat
- Comme il est difficile de forger un certificat, on peut sans trop de risque le placer dans un annuaire public


```
Certificate:
    Data:
        Version: v3 (0x2)
        Serial Number: 3 (0x3)
        Signature Algorithm: PKCS #1 MD5 With RSA Encryption
        Issuer: OU=Ace Certificate Authority, O=Ace Industry, C=US
        Validity:
            Not Before: Fri Oct 17 18:36:25 1997
            Not After: Sun Oct 17 18:36:25 1999
        Subject: CN=Jane Doe, OU=Finance, O=Ace Industry, C=US
        Subject Public Key Info:
            Algorithm: PKCS #1 RSA Encryption
            Public Key:
                Modulus:
                    00:ca:fa:79:98:8f:19:f8:d7:de:e4:49:80:48:e6:2a:2a:86:
                    ed:27:40:4d:86:b3:05:c0:01:bb:50:15:c9:de:dc:85:19:22:
                    43:7d:45:6d:71:4e:17:3d:f0:36:4b:5b:7f:a8:51:a3:a1:00:
                    98:ce:7f:47:50:2c:93:36:7c:01:6e:cb:89:06:41:72:b5:e9:
                    73:49:38:76:ef:b6:8f:ac:49:bb:63:0f:9b:ff:16:2a:e3:0e:
                    9d:3b:af:ce:9a:3e:48:65:de:96:61:d5:0a:11:2a:a2:80:b0:
                    7d:d8:99:cb:0c:99:34:c9:ab:25:06:a8:31:ad:8c:4b:aa:54:
                    91:f4:15
                Public Exponent: 65537 (0x10001)
        Extensions:
            Identifier: Certificate Type
                Critical: no
                Certified Usage:
                    SSL Client
            Identifier: Authority Key Identifier
                Critical: no
                Key Identifier:
                    f2:f2:06:59:90:18:47:51:f5:89:33:5a:31:7a:e6:5c:fb:36:
                    26:c9
    Signature:
        Algorithm: PKCS #1 MD5 With RSA Encryption
        Signature:
            6d:23:af:f3:d3:b6:7a:df:90:df:cd:7e:18:6c:01:69:8e:54:65:fc:06:
            30:43:34:d1:63:1f:06:7d:c3:40:a8:2a:82:c1:a4:83:2a:fb:2e:8f:fb:
            f0:6d:ff:75:a3:78:f7:52:47:46:62:97:1d:d9:c6:11:0a:02:a2:e0:cc:
            2a:75:6c:8b:b6:9b:87:00:7d:7c:84:76:79:ba:f8:b4:d2:62:58:c3:c5:
            b6:c1:43:ac:63:44:42:fd:af:c8:0f:2f:38:85:6d:d6:59:e8:41:42:a5:
            4a:e5:26:38:ff:32:78:a1:38:f1:ed:dc:0d:31:d1:b0:6d:67:e9:46:a8:
            dd:c4
```

Hiérarchie de CA

- Hypothèse :
  - si les deux utilisateurs partagent un CA commun alors on suppose qu'ils connaissent sa clef publique
  - chaque CA a des certificats pour les clients (vers l'avant) et le parent (vers l'arrière)
  - chaque client fait confiance aux certificats parents
- Solution : hiérarchie de CA
  - employer les certificats liant les membres de la hiérarchie pour valider d'autre CA
- Objectif : permettre la vérification de n'importe quel certificat d'un CA par des utilisateurs de tout autre CA dans la hiérarchie

CA Hierarchy Use



Révocation de certificat

- Les certificats ont une période de validité
- On doit pouvoir le retirer avant l'échéance car, par exemple :
  - la clef privée de l'utilisateur est compromise
  - l'utilisateur n'est plus certifié par ce CA
  - le certificat du CA est compromis
- Les CA maintiennent la liste de certificats retirés
  - la liste de révocation de certificat (CRL)
- Les utilisateurs devraient pouvoir vérifier les certificats avec les CRLs

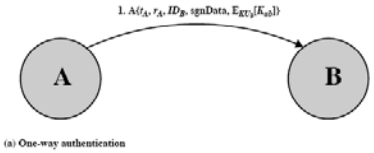
Authentication Procedures

- X.509 inclut trois procédures alternatives d'authentification :
  - Authentification à sens unique (one way)
  - Authentification à 2 passages (two way)
  - Authentification à 3 passages (three way)
- Tous emploient des signatures à clés publiques (voir plus loin)

The X.509 standard specifies the authentication protocols that can be used when obtaining and using certificates. 1-way for unidirectional messages (like email), 2-way for interactive sessions when timestamps are used, 3-way for interactive sessions with no need for timestamps (and hence synchronised clocks).

One-Way Authentication

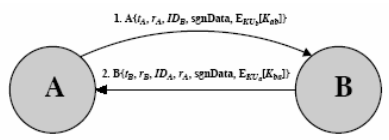
- 1 message (A->B) est utilisé pour établir
  - L'identité de A et l'origine du message
  - Le destinataire du message
  - L'intégrité du message
- le message doit inclure l'horodateur (t_A), le nonce (r_A), l'identité de B (ID_B) et est signé par A (sgnData)



(a) One-way authentication

Two-Way Authentication

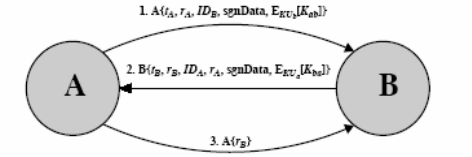
- 2 messages (A->b, B->a) – même principe +
  - l'identité de B et que cette réponse est de B
  - Le destinataire
  - intégrité et l'origine de réponse
- la réponse inclut le nonce de A, l'horodateur et le nonce de B



Permet donc aux deux parties de vérifier l'identité de l'autre

Three-Way Authentication

- 3 messages (A->b, B->a, A->b) qui permettent l'authentification sans horloges synchronisées
- la réponse de A à B contient la copie signée du nonce de B
- signifie que des horodateurs n'ont pas besoin d'être vérifiés ou comptés



X.509 Version 3

- Nécessité d'information additionnelle
  - email/URL, détails de politique, contraintes d'utilisation
- Définition d'une méthode générale d'extension plutôt que création de nouveaux champs statiques
- Composition des extensions
  - marqueur
  - indicateur d'importance
  - valeur

Distributions de clés

Clés de session

Distribution des clés de session

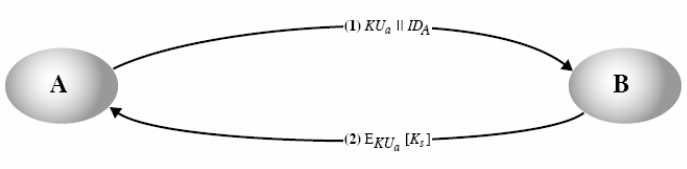
- Distribution de clé publique
  - Simple
  - Permet la confidentialité et/ou l'authentification
  - Lent
- Objectif : protection du contenu d'un message
- Solution :
  - Système hybride + Clé de session
- Souhait :
  - disposer de plusieurs solutions alternatives pour négocier une session (voir SSL)

Distribution simple de clés de session

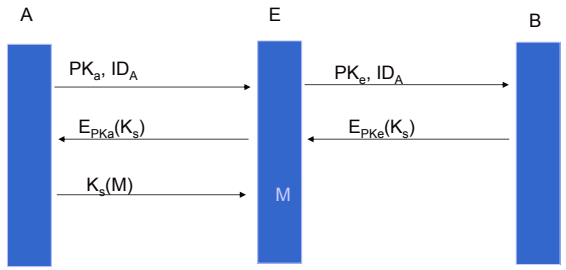
- Solution proposée par Merkle en 1979
  1. A produit d'une nouvelle paire de clé publique provisoire
  2. A envoie à B sa clef publique (et son identité)
  3. B produit d'une clef K de session et l'envoie à A (la clé est chiffrée au moyen de la clef publique fournie par A)
  4. A déchiffre la clef de session et tous les deux l'emploient
- Problème :
  - un adversaire peut arrêter et personnifier les deux moitiés de protocole (attaque *man in the middle*)

Distribution simple de clés de session

- S'ils ont au préalable échangé des clés publiques de manière sûre :



Attaque « Man in the middle » (MITM)

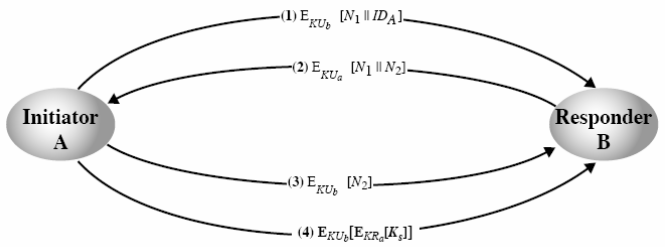


Ce protocole est vulnérable à une attaque active. Si un adversaire, E, a la commande de la voie de transmission intervenante, alors E peut compromettre la communication de la façon suivante sans être détecté :

1. A produit d'une paire de clé public/privée {KU_a, KR_a} et transmet un message destiné à B se composant de KU_a et d'une marque de A, ID_A.
2. E arrête le message, crée sa propre paire de clé public/private {KU_e, KR_e} et transmet {KU_e||ID_A} à B.
3. B produit d'une clef secrète, K_s, et transmet $E_{KUe}[K_s]$.
4. E arrête le message, et apprend K_s par $D_{KRe}[E_{KUe}[K_s$ de calcul]]
5. E transmet $E_{KUa}[K_s]$ à A.

Le résultat est qu'A et B connaissent K_s et ignorent que K_s a été également indiqué à E. A et B peuvent maintenant échanger des messages en utilisant K_s. E n'interfère plus activement la voie de transmissions mais écoute clandestinement simplement. Connaissant K, E peut déchiffrer tous les messages, et A et B sont ignorants du problème.

Distribution de clés secrètes avec confidentialité et authentification



Échange des clés

Diffie-Hellman

Échange de Clef Diffie-Hellman

- Premier schéma de clé publique proposé
- Diffie et Hellman en 1976 avec l'exposition des concepts de clés publiques
  - note : on sait maintenant que James Ellis (UK CESG) a secrètement proposé le concept en 1970
- Méthode pratique pour l'échange public d'une clef secrète (ou de session)
- Utilisé dans un certain nombre de produits commercial (SSL, …)

Échange De Clef Diffie-Hellman

- Ne peut pas être employé pour échanger un message arbitraire
- Permet d'établir une clef commune connue seulement des deux participants
- la valeur de la clef dépend des participants (et de l'information sur leur clés privée et publique)

Échange De Clef Diffie-Hellman

- Ne peut pas être employé pour échanger un message arbitraire
- La valeur de la clef dépend des participants (et de l'information sur leur clés privée et publique)
- Basé sur l'élévation à une puissance dans un champ fini $\rightarrow$ facile
- La sécurité se fonde sur la difficulté de calculer des logarithmes discrets $\rightarrow$ difficile

Principe de l'algorithme

- Soient A et B, les deux parties de la communication.
- A génère un nombre premier *p* et un primitif *a* (*p* et *a* sont publics).

  Déf: Soient *p* premier et a < *p*.
  a est un **primitif** de *p* si $\forall\ b \in [1, p-1], \exists\ g$ tel que $a^g \equiv b \bmod p$
- A et B génèrent chacun un nombre aléatoire
  - x_A (< *p*) et x_B (< *p*) (secrets).

Principe de l'algorithme

A	B
1. X_A	X_B
2. $y_A = a^{x_A} \pmod p$	$y_B = a^{x_B} \pmod p$

y_A et y_B sont rendues publiques

3. $K_{AB} = y_B^{\ x_A} \bmod p$ $\qquad$ $K_{BA} = y_A^{\ x_B} \bmod p$

Primitif:

Tout nombre b peut être exprimé par $a^g \pmod p$

Ex:

Pour p =11, 2 est un primitif:

$2^{10} \equiv 1 \pmod{11}$

$2^{1} \equiv 2 \pmod{11}$

$2^{8} \equiv 3 \pmod{11}$

$2^{2} \equiv 4 \pmod{11}$

$2^{4} \equiv 5 \pmod{11}$

$2^{9} \equiv 6 \pmod{11}$

$2^{7} \equiv 7 \pmod{11}$

$2^{3} \equiv 8 \pmod{11}$

$2^{6} \equiv 9 \pmod{11}$

$2^{5} \equiv 10 \pmod{11}$

Inversibilité

- A et B partagent ainsi la même clé
- Vu que $K_{AB} = K_{BA}$
- En effet :

$$K_{AB} = y_B^{x_A} \bmod p = (a^{x_B})^{x_A} (\bmod p)$$
$$= a^{x_A x_B} (\bmod p) = K_{BA}$$

Exemple

- Alice et Bob souhaitent échanger une clé:
- Ils se mettent d'accord sur p = 353 et α = 3
- Ils sélectionnent une clé privée aléatoire :
  - A choisit Xa = 97, B choisit Xb = 233
- Ils calculent la clé publique :
  - $Y_A = 3^{97} \bmod 353 = 40$ (Alice)
  - $Y_B = 3^{233} \bmod 353 = 248$ (Bob)
- Et calculent finalement la clé de session :
  - $K_{AB} = (Y_B)^{Xa} \bmod 353 = 24897 = 160$ (Alice)
  - $K_{AB} = (Y_A)^{Xb} \bmod 353 = 40233 = 160$ (Bob)

Sécurité

- Un imposteur pourrait essayer de trouver K_{AB} à partir de y_A , y_B de *a* et de *p* en calculant:

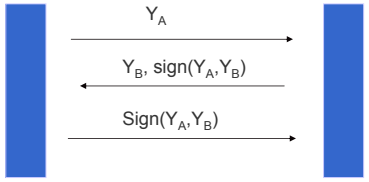
$$K_{AB} = y_B^{\log_a y_A} (\mathrm{mod}\ p)$$

- La sécurité de cet algorithme repose sur le fait qu'il est impossible de calculer $\log_a y$ (mod *p*) en algèbre modulaire.
- NB: L'authentification de A et B n'est pas assurée par cet algorithme.

Protocole « station à station » (STS)

- Diffie-Hellman sensible à l'attaque MITM.
- Solution : insérer de l'information relative
  - Soit à l'identité de l'expéditeur
  - Soit au message échangé
- Moyen :
  - Signatures digitales
  - MAC
  - …

Protocole « station à station » (STS)



Sign est calculé au moyen de la clé définie par DH

Références

- http://home.ecn.ab.ca/~jsavard/crypto/pk0503.htm
- http://home.ecn.ab.ca/~jsavard/crypto/mi0607.htm
- http://developer.netscape.com/docs/manuals/security/pkin/contents.htm
- www.itl.nist.gov/
- [schneier] – ch3, 7, 8, 22
- [stallings] – ch7, 10, 14.2

Questions

- Problématique de la gestion des clés
- Distributions de clé
  - Symétrique
  - Publique
  - De session
- Certificats
- Diffie-Hellman