

SÉCURITÉ DES RÉSEAUX  
CERTIFICATS X509 ET CLÉS PGP

---

A. Guermouche

Certificats X509

Comment améliorer la confiance?

Clés PGP

Certificats X509

Comment améliorer la confiance?

Clés PGP

## X509

- X509 définit un cadre pour la fourniture de services d'authentification par répertoire X500.
- Le répertoire X500 sert de dépôt de certificats à clés publique.
- Chaque certificat X509 contient la clé publique de l'utilisateur et est signée par la clé privée de l'autorité de certification confiance.
- X509 définit des protocoles alternatifs d'authentification basés sur l'utilisation des certificats à clé publique.
- X509 est basé sur l'utilisation de cryptographie à clé publique et de signature numérique.
  - La norme ne dicte pas l'utilisation d'un algorithme spécifique.
  - Le procédé de signature nécessite l'utilisation d'une fonction de hachage.

# Certificats

Le certificats à clé publique est au coeur de la résolution X509.

- Certificats délivrés par des autorités de certifications (AC) et placés dans le répertoire par l'AC ou l'utilisateur.
- Le répertoire ne fournit qu'un emplacement facilement accessible des utilisateur pour l'obtention des certificats.

version : Différencie les versions successives du format de certificats.

numéro de série : Valeur entière, unique pour l'AC émettrice, associée au certificat.

id. de l'algo. de signature : Type de l'algorithme utilisé pour la signature, ce champs est peu utile vu que le type de l'algorithme de signature est contenu dans la partie signature.

version
Numéro de série
id. de l'algo. de signature
Nom du créateur
Période de validité
Nom du sujet
Infos. clé publique du sujet
id. unique du créateur
id. unique du sujet
Extensions
Signature

# Certificats

Le certificats à clé publique est au coeur de la résolution X509.

- Certificats délivrés par des autorités de certifications (AC) et placés dans le répertoire par l'AC ou l'utilisateur.
- Le répertoire ne fournit qu'un emplacement facilement accessible des utilisateur pour l'obtention des certificats.

nom d'émetteur : Nom X500 de l'AC qui a créé le certificats.

période de validité : Dates de début et de fin de validité du certificats.

nom du sujet : Nom de l'utilisateur auquel le certificats se réfère.

version
Numéro de série
id. de l'algo. de signature
Nom du créateur
Période de validité
Nom du sujet
Infos. clé publique du sujet
id. unique du créateur
id. unique du sujet
Extensions
Signature

# Certificats

Le certificats à clé publique est au coeur de la résolution X509.

- Certificats délivrés par des autorités de certifications (AC) et placés dans le répertoire par l'AC ou l'utilisateur.
- Le répertoire ne fournit qu'un emplacement facilement accessible des utilisateur pour l'obtention des certificats.

info. clé publique du sujet : La clé publique du sujet ainsi que l'identifiant de l'algorithme utilisé.

id. unique de l'émetteur : Une zone facultative de bits employée pour identifier sans ambiguïté l'AC émettrice (utile si un nom X500 a été utilisé pour différentes entités).

id. unique du sujet : Une zone facultative de bits employée pour identifier sans ambiguïté le sujet.

version
Numéro de série
id. de l'algo. de signature
Nom du créateur
Période de validité
Nom du sujet
Infos. clé publique du sujet
id. unique du créateur
id. unique du sujet
Extensions
Signature

# Certificats

Le certificats à clé publique est au coeur de la résolution X509.

- Certificats délivrés par des autorités de certifications (AC) et placés dans le répertoire par l'AC ou l'utilisateur.
- Le répertoire ne fournit qu'un emplacement facilement accessible des utilisateur pour l'obtention des certificats.

extensions : Un ensemble de champs d'extension.

signature : Couvre tous les autres champs du certificat. Elle contient le code de hachage de tous les autres champs chiffré avec la clé privé de l'AC.

version
Numéro de série
id. de l'algo. de signature
Nom du créateur
Période de validité
Nom du sujet
Infos. clé publique du sujet
id. unique du créateur
id. unique du sujet
Extensions
Signature



La norme emploie la notation suivante pour définir un certificat :

$$CA \ll A \gg = CA\{V, SN, AI, CA, TA, A, Ap\}$$

où :

$Y \ll X \gg$  est le certificat de l'utilisateur  $X$  émis par l'autorité de certification  $Y$ .

$Y\{I\}$  est la signature de  $I$  par  $Y$ . Elle se compose de  $I$  complété d'un code de hachage chiffré.

- L'AC signe le certificat avec sa clé privée.
- Si la clé publique correspondante est connue de l'utilisateur alors il peut vérifier la validité du certificat.

# Obtention d'un certificat utilisateur

Les certificats utilisateurs produits par une AC ont les caractéristiques suivantes :

- n'importe quel utilisateur ayant accès à la clé publique de l'AC peut vérifier la clé publique d'un autre utilisateur qui a été certifié.
- aucune partie autre que l'AC ne peut modifier le certificat sans que cela ne soit détecté.

Si tous les utilisateurs souscrivent à la même AC alors :

- il existe une confiance commune en cette AC
- les certificats peuvent être placés dans le répertoire pour être accessibles à tous les utilisateurs
- les utilisateurs peuvent s'échanger les certificats directement (l'utilisation la plus simple des certificats).

# Obtention d'un certificat utilisateur

Les certificats utilisateurs produits par une AC ont les caractéristiques suivantes :

- n'importe quel utilisateur ayant accès à la clé publique de l'AC peut vérifier la clé publique d'un autre utilisateur qui a été certifié.
- aucune partie autre que l'AC ne peut modifier le certificat sans que cela ne soit détecté.

Exemple :

Si B est en possession du certificat de A, B a la certitude que :

- les messages qu'il chiffre avec la clé publique de A seront à l'abri d'une écoute clandestine
- et que les messages signés avec la clé privé de A sont infalsifiables.

# Un cas plus complexe

Si le nombre d'utilisateurs est important, il peut être intéressant d'utiliser plus d'une AC.

Exemple : Supposons que  $A$  ait obtenu son certificat de  $X_1$  et  $B$  son certificat de  $X_2$ .

Problème : si  $A$  ne connaît pas la clé publique de  $X_2$ , le certificat de  $B$  lui est inutile.

Solution :

si les deux AC ont échangé ont échangé de manière sûre leur clé publique, alors lorsqu'il doit parler à  $B$ ,  $A$  doit :

1. Obtenir à partir du répertoire, le certificat  $X_2$  signé par  $X_1$  et retrouver et vérifier à l'aide de la clé publique de  $X_1$  la clé publique de  $X_2$ .
2. Utiliser le répertoire pour obtenir le certificat de  $B$  (qui est signé par  $X_2$ ) et utiliser la clé publique de  $X_2$  pour vérifier et obtenir en toute sécurité la clé publique de  $B$ .

## Un cas plus complexe

Si le nombre d'utilisateurs est important, il peut être intéressant d'utiliser plus d'une AC.

Exemple : Supposons que  $A$  ait obtenu son certificat de  $X_1$  et  $B$  son certificat de  $X_2$ .

Problème : si  $A$  ne connaît pas la clé publique de  $X_2$ , le certificat de  $B$  lui est inutile.

Solution :

$A$  dans cet exemple a utilisé une chaîne de certificats pour obtenir la clé publique de  $B$ . Cette est exprimée en notation X509 par :

$$X_1 \ll X_2 \gg X_2 \ll B \gg$$

# Révocation de certificats

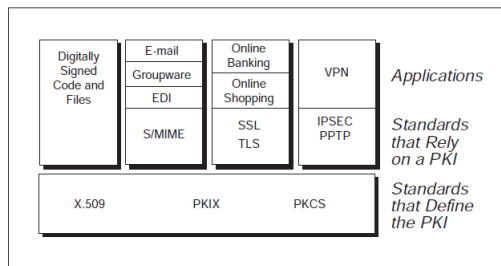
- Chaque certificats a une période de validité.
- Les certificats peuvent être retirés pour diverses raisons.
- Chaque AC maintient une liste se composant de tous les certificats révoqués mais non expirés (CRL) et la signe.
- Un client peut vérifier le statut d'un certificat soit
  - en téléchargeant la liste des certificats révoqués (CRL)
  - ou en utilisant le protocole OCSP qui permet de demander en ligne l'état du certificat.
- Dès qu'un utilisateur reçoit un certificat il est censé vérifier si le certificat a été révoqué ou non.

## PKIX (*PKI for X.509 certificates*)

La spécification de PKIX est basée sur deux standards:

**X509.** qui spécifie le format et la gestion des certificats

**PKCS.** qui est une série de standards couvrant les PKI pour la demande et le renouvellement des certificats, la distribution des CRL, et l'interopérabilité des PKI



## Comment savoir que je parle bien au bon serveur?

- Problème de la confiance accordée aux autorités de certification.
- Que se passe-t-il si une CA se fait voler sa clé privée? (Diginotar par exemple)
- Problème de la vérification de l'identité du propriétaire du certificat côté client.
  - Une erreur très répandue est de considérer que c'est OpenSSL qui va valider le nom du propriétaire dans le certificat du serveur.



## Null Prefix

Qu'arrive-t-il lorsque le nom du propriétaire contient des caractères imprévus? `www.victime.fr\0.pirate.fr`

- le test du nom du propriétaire ne pas donner le résultat qu'il faudrait.
- vérifier tous le champs, rechercher les incohérences et essayer d'être le plus respectueux possible des RFCs (RFC 6125 dans notre cas).

# Quelques problèmes connus

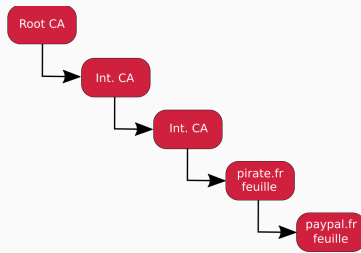
## Basic Constraint

L'extension *BasicConstraints* (BC) spécifie si un certificat est un certificat de CA (et donc qu'il peut en signer d'autres). Qu'est ce qui se passe si un certificat feuille a l'extension BC?

- si le client ne fait pas les bonnes vérifications, le certificat feuille peut être considéré comme un certificat CA intermédiaire.
- un attaquant peut alors générer un certificat frauduleux et le signer avec son certificat feuille

### Bonne pratique :

Lors de la validation d'une chaîne de certificats, assurez vous que *pathLen* est valide et que le champ *cA* est égal à *TRUE* pour chaque certificat non feuille.



Certificats X509

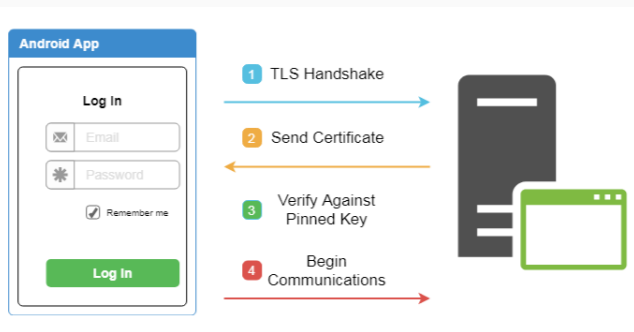
Comment améliorer la confiance?

Clés PGP

# Certificate Pinning (HPKP)

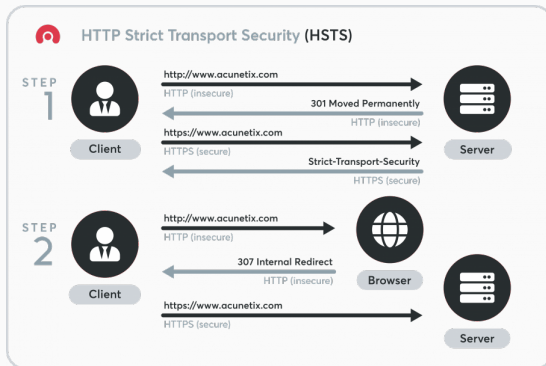
## HTTP Public Key Pinning?

- Le serveur peut demander au client de conserver le certificat pendant un certain temps (TTL définit par le navigateur).
- À chaque interaction avec le serveur, le client compare le certificat reçu avec celui qu'il a enregistré.



## HTTP Strict Transport Security?

Permet de forcer le client pendant un certain temps à utiliser HTTPS après une première connexion HTTP. Pendant l'intervalle de validité, toute connexion au serveur se faire avec HTTPS.



# Certificate Transparency (CT)

## Fonctionnement

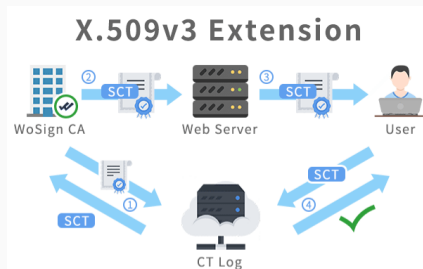
Solution proposée par Google.

1. L'administrateur du serveur achète un certificat à la CA
2. La CA valide l'opérateur du serveur
3. La CA crée un pré-certificat
4. La CA enregistre le pré-certificat avec le *CT-log server*, qui renvoie un horodatage du certificat signé (SCT). Le SCT est une promesse d'ajout du certificat au CT-Log.
5. La CA émet le certificat SSL
6. Le certificat SSL peut comporter le SCT
7. Le navigateur valide le certificat SSL lors de la poignée de main TLS
8. Le navigateur valide le SCT fourni lors de la poignée de main TLS, soit via OCSP, ou une extension TLS, ou encore à partir des informations contenues dans le certificat.
9. La communication peut se dérouler normalement

# Certificate Transparency (CT)

## Validation X509

- Le CT Log est mis à jour par les CA à chaque demande de certificat.
- La CA ajoute le SCT provenant du CT-Log au certificat avant de le signer.
- Dès qu'un client reçoit un certificat, il consulte le CT log pour vérifier que le certificat est valide.
- Avantage : Le comportement du serveur n'est pas modifié.

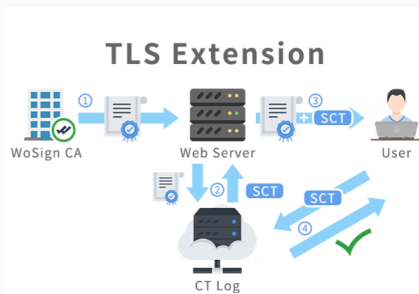


source [https://www.wosign.com/English/News/2016\\_wosign\\_CT.htm](https://www.wosign.com/English/News/2016_wosign_CT.htm)

# Certificate Transparency (CT)

## Extension TLS

- La CA fournit le certificat tel quel au serveur.
- Le serveur enregistre le certificat auprès du CT-log et obtient le SCT.
- Lors de la poignée de main, le serveur fournit au client le SCT via une extension TLS.
- Le client valide le SCT auprès du CT log.
- Avantage : Le comportement de la CA n'est pas modifié.



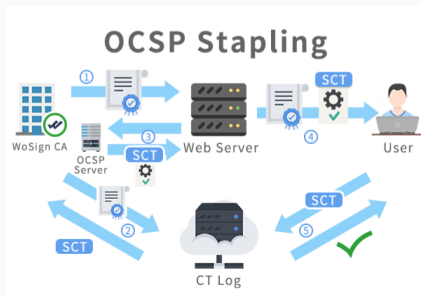
source [https://www.wosign.com/English/News/2016\\_wosign\\_CT.htm](https://www.wosign.com/English/News/2016_wosign_CT.htm)



# Certificate Transparency (CT)

## OCSP Stapling

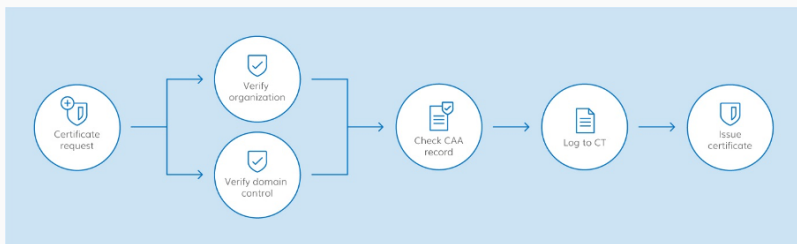
- La CA fournit le certificat au serveur et au CT-log.
- L'administrateur du serveur effectue alors une requête OCSP à la CA.
- La CA répond avec le SCT, que le serveur peut inclure dans une extension OCSP pendant la poignée de main TLS.
- Avantages : 1) La CA prend la responsabilité du SCT. 2) Elle peut de plus recevoir le SCT de manière asynchrone sans retarder l'émission du certificat.



# Certificate Authority Authorization (CAA)

## Certificate Authority Authorization

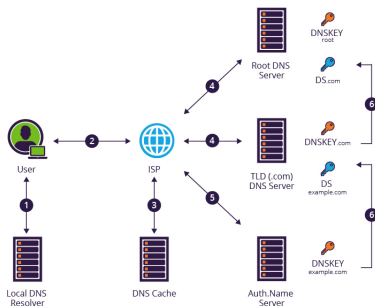
- CAA est une extension DNS qui permet aux propriétaires de domaines de spécifier quelles CA ont le droit d'émettre un certificat pour leur domaine.
- L'application de la CAA repose sur la CA, et non sur le navigateur.
  - Quand une organisation demande un certificat à la CA, celle-ci vérifie si le DNS du domaine contient un enregistrement CAA.
  - Si la CA est spécifiée dans l'enregistrement CAA, le certificat sera accordé.
  - Sinon, l'AC signalera la demande illégitime.



## DNSSEC

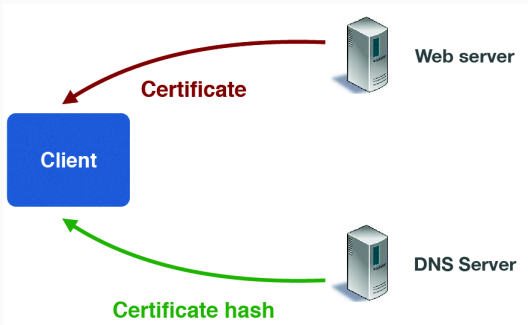
Avec DNSSEC, le contenu de réponse du DNS peut être considéré comme authentique.

- Utilisation de DNSSEC pour diffuser des données authentifiées (Hachages de certificats, ...).



## DNS-based Authentication of Named Entities

Ajouter des hachages de certificat (resp. fingerprint de clés ssh) dans les enregistrements DNS pour améliorer la vérification.



source <https://www.ripe.net/support/training/material/dnssec-training-course/dnssec-slides.pdf>

## DNS-based Authentication of Named Entities

Ajouter des hachages de certificat (resp. fingerprint de clés ssh) dans les enregistrements DNS pour améliorer la vérification.

- DNSSEC ajoute des clés et des signatures aux enregistrements DNS.
- DANE ajoute des enregistrements pour les clés utilisées par les applications
- Peut être considérée comme une PKI:
  - Moins d'intermédiaires de confiance (ICANN root, TLD registry, registrar, et nos propres clés DNSSEC)
  - Se construit au dessus de relations d'authentification existantes
- Problème du déploiement de DNSSEC.

## TACK

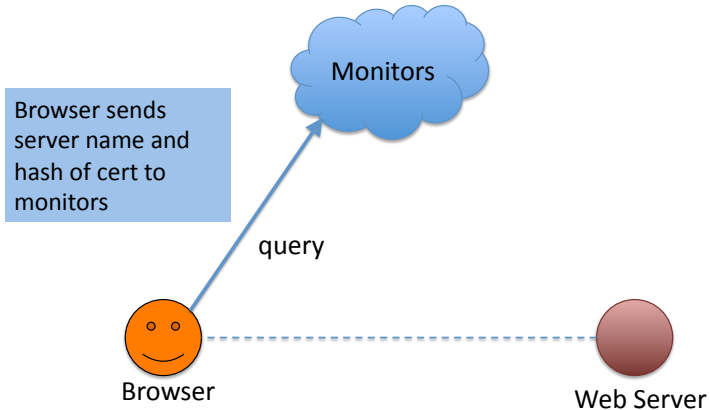
Pinning de clé publique lors d'une communication sécurisée (à la SSH).

## Convergence

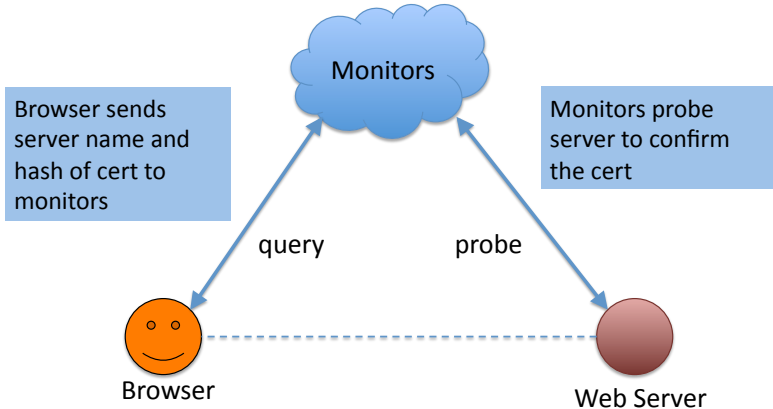
- Introduction de la notion de *trust agility*
- Ajout d'observateurs appelés moniteurs qui vont servir à valider que les informations (certificats, clés) sont bien les bonnes.
- Le client a la liberté de refuser un certificat s'il considère qu'il peut y avoir un problème.

Les transparents suivant sont extraits de <https://crypto.stanford.edu/RealWorldCrypto/slides/perrin.pdf>

# Convergence

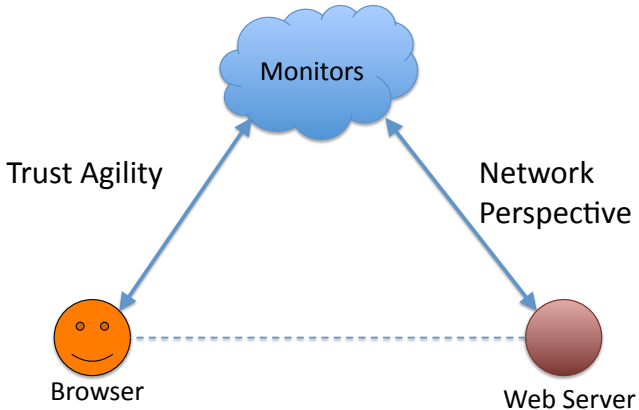


# Convergence

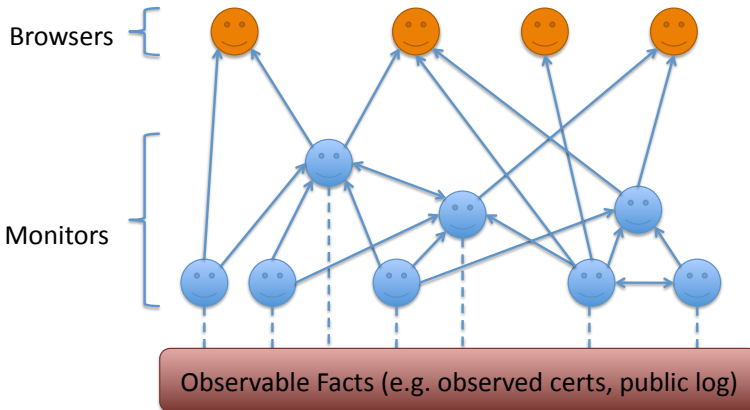




# Convergence



# Trust Agility in action



# Observational Trust Modes

- Net Perspective: “Do you see what I see?”
- Key Continuity: “Is this the same as before?”
- SSH, Convergence, Perspectives, etc.
- Rationale: Internet works for most people most of the time

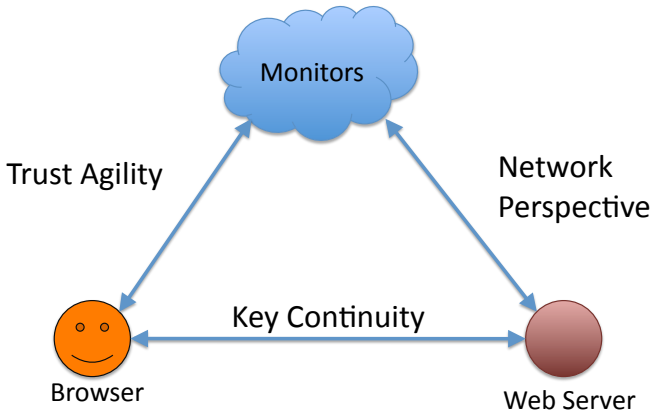
# Convergence Challenges

- Online lookups
  - Performed on first connection or key discontinuity
  - Costly infrastructure
  - Performance and reliability risk

# Observational Trust Challenges

- Key Continuity
  - Doesn't protect initial connection
  - Doesn't handle key changes well
- Network Perspective
  - Handles initial connection and key changes at cost of online lookups
  - Doesn't handle multiple-keys-per-site well

# Observational Trust



Certificats X509

Comment améliorer la confiance?

Clés PGP

# Pourquoi?

- Le courrier électronique est l'application réseau la plus utilisée et la plus répandue.
- Pas de sécurité dans les échanges standards de mail.
- Forte croissance de la demande de services garantissant la confidentialité et l'authenticité des messages.
- Deux modèles sont susceptibles d'être largement utilisés : *Pretty Good Privacy* (PGP) et S/MIME.



- PGP est en grande partie le résultat des recherches de Phil Zimmermann.
- Utilisation dans les applications de courrier électronique et de stockage de fichiers.
- Succès expliqué par :
  - Sa gratuité et sa portabilité.
  - La diversité des algorithmes de chiffrement sur lesquels il est basé.
  - Un large éventail d'applications.
  - Son indépendance (n'est contrôlé par aucune autorité).
- PGP fournit cinq services : authentification, confidentialité, compression, compatibilité entre applications de courrier électronique et segmentation.

# Authentification

L'authentification se déroule en effectuant la séquence d'opérations suivante :

1. L'expéditeur crée un message.
2. On utilise SHA-1 pour générer le code de hachage de 160 bits du message.
3. Le code de hachage est chiffré avec RSA en utilisant la clé privée de l'expéditeur, et le résultat est ajouté au début du message.
4. Le récepteur emploie RSA avec la clé publique de l'expéditeur pour déchiffrer et récupérer le code hachage.
5. Le receveur produit un nouveau code de hachage pour le message et le compare au code de hachage déchiffré. S'il y a correspondance, le message est considéré comme authentique.

La signature est normalement attachée au message mais peut être isolée dans le cas où un message doit être signé par plusieurs personnes (un contrat légal par exemple).

La confidentialité est assurée en chiffrant les messages à transmettre ou à stocker localement comme fichiers.

- Possibilité d'utiliser différents algorithmes de chiffrement (CAST-128, IDEA, TDEA, 3DES ...).
- Génération d'une clé de session pour chaque message.

Lors de l'émission d'un message :

1. L'expéditeur génère un nombre aléatoire de 128 bits à utiliser comme clé de session uniquement pour ce message.
2. Le message est chiffré en utilisant CAST-128 (ou IDEA, ou 3DES, ...) avec la clé de session.
3. La clé de session est chiffrée avec RSA (ou DSS) en utilisant la clé publique du destinataire et est ajoutée au message.
4. Le destinataire utilise RSA avec sa clé privée pour déchiffrer et récupérer la clé de session.
5. La clé de la session est utilisée pour déchiffrer le message.

# Identifiant de clé

Comment faire pour déchiffrer la clé de session si un utilisateur donné a plusieurs paires de clés ?

**solution naïve.** Transmettre la clé publique avec le message.

- Gaspillage d'espace.

**solution "futée".** Associer à chaque clé un identifiant unique calculé à partir de l'identifiant de l'utilisateur et de la clé.

- Les identifiants de clé doivent être attribués et enregistrés pour être permettre à l'émetteur et au récepteur d'établir la correspondance identifiant/clé.

**solution retenue.** Attribuer à chaque clé un identifiant qui a une forte probabilité d'être unique.

- L'identifiant de la clé se compose de ses 64 bits les moins significatifs.

# Format d'un message PGP

horodatage : Date et heure  
auxquelles la signature a été  
produite.

résumé du message : Le résumé  
160 bits SHA-1 chiffré avec la clé  
privée de A.

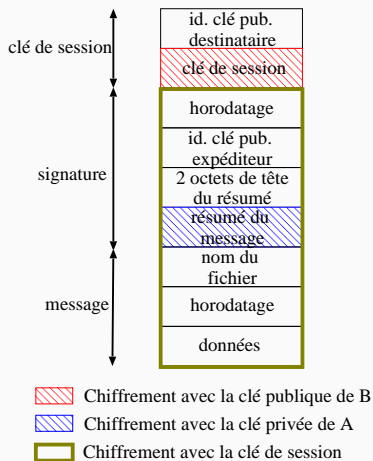


Figure 1: Message PGP de A à B.

# Format d'un message PGP

2 octets de tête du résumé de message : Pour permettre à B de déterminer si la bonne clé publique a été utilisée pour déchiffrer le résumé lors de l'authentification.  
identifiant de clé publique de A : Identifie la clé publique qui doit être utilisée pour déchiffrer le message.

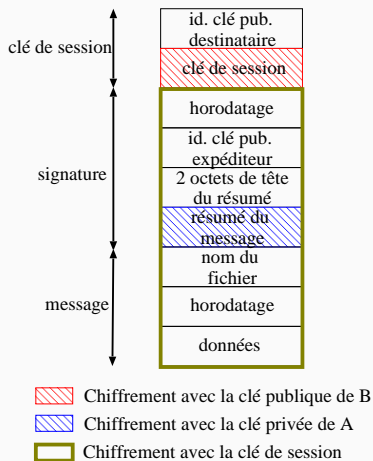


Figure 1: Message PGP de A à B.

# Anneaux de clés

- Les paires de clés publiques/clés privées doivent être organisées et stockées efficacement sur chacun des noeuds.
- PGP utilise deux structures (*anneaux*) de données sur chaque noeud :
  - Une structure pour stocker les paires clés publiques/clés privées du noeud.
  - Une structure pour stocker les clés publiques d'autres utilisateurs connus pas ce noeud.

# Anneaux de clés

Les anneaux peuvent être vus comme des tables.

L'anneau de clés privées :

- Contient pour chaque clé : un horodatage (date de création), un identifiant de la clé, une clé publique, une clé privée chiffrée, et un identifiant utilisateur.
- La clé privée est chiffrée en utilisant CAST-128, IDEA ou TDEA.
  - L'utilisateur choisit une *passphrase* à utiliser pour chiffrer les clés privées.
  - Quand le système génère une nouvelle clé privée, il demande à l'utilisateur sa *passphrase* et génère à partir de la phrase un code de hachage SHA-1 de 160 bits et la détruit.
  - La clé privée est chiffrée avec le code de hachage. Le code de hachage est ensuite détruit.

L'anneau de clés publiques :

- Contient pour chaque clé : un horodatage, un identifiant de la clé, la clé publique, et un identifiant utilisateur.



## L'utilisation de la confiance (1/2)

Est ce que les clés contenues dans l'anneau de clés publiques sont authentiques?

- Utilisation d'intermédiaire de confiance pour récupérer la clé publique de l'utilisateur distant.
- Obtenir la clé publique à partir d'une autorité de certification.
- Chaque entrée de l'anneau de clés publiques est vue comme un certificat ayant un champ de confiance. Plus le niveau de confiance est élevé plus fort sera le lien entre l'identifiant utilisateur et la clé.
- Le certificat peut être signé par zéro ou plusieurs signatures. À chaque signature est associé un degré de confiance.
- Le champ de légitimité de la clé est dérivé de l'ensemble des champs de confiance de signature de l'entrée.

## L'utilisation de la confiance (2/2)

- Lorsque A insère une nouvelle clé publique dans son anneau :
  - Si A est le propriétaire, la clé a un indice de confiance absolu.
  - Sinon, PGP demande à A d'évaluer le degré de confiance qui sera accordé au propriétaire de la clé.
- Quand la nouvelle clé publique est saisie, une ou plusieurs signatures peuvent lui être attachées. Quand une signature est insérée dans l'entrée, PGP vérifie si l'utilisateur est connu (à partir de l'anneau de confiance). Si c'est le cas, il lui affecte le degré de confiance de l'utilisateur.
- La valeur de légitimité de la clé est calculée sur la base des champs de confiance de signature présents dans cette entrée.