Sécurité informatique
Cryptographie et IGC
ENSEEIHT 3A IMA

Pierre-Yves Bonnetain-Nesterenko
py.bonnetain@ba-consultants.fr

B&A Consultants – BP 70024 – 31330 Grenade-sur-Garonne

Année 2018-2019

Plan

B&A CONSULTANTS SECURITE INFORMATIQUE

Cryptographie et confidentialité

Segmente dans le temps les données en les rendant inaccessibles quand elles ne sont pas utilisées.

- Inutile sur données en cours d'utilisation (en clair [1]).
- Inefficace si mauvaise sécurité applicative (idem).
- Inintéressant si pas compris comment bien utiliser la cryptographie.

Principes de Kerckhoffs

Stipulent que la seule « bonne » cryptographie est celle dans laquelle tout (y compris les algorithmes) peut être connu de l'adversaire sans affaiblir la sécurité, tant que les clés restent protégées.

1. Sauf chiffrement homomorphe, encore domaine recherche

Un outil très tranchant

Dans la famille « connaître les inconvénients de nos outils », je demande « la cryptographie ».

- Chiffrer ses données les protège contre tout accès non autorisé, mais...
- Que se passe-t-il si vous perdez/oubliez le mot de passe/la clé ?
- Comment donner à des tiers *légitimes* l'accès parallèle à ces mêmes données ?
- Comment bien chiffrer des données pour qu'aucun exemplaire en clair n'existe ?

B&A CONSULTANTS SECURITE INFORMATIQUE

Caractéristiques émergentes

- Authentification (réciproque ou non) : une extrémité peut s'assurer de l'identité de l'autre extrémité
- Non-répudiation : une action authentifiée ne peut être niée par son auteur
- Intégrité : une altération accidentelle ou malveillante d'un message peut être détectée

Avec un bémol

Tout cela (y compris la confidentialité) suppose comme point de départ une bonne utilisation de la cryptographie et une bonne gestion des clés.

Un outil à multiple usages

- Confidentialité : longue durée de vie. Révocation des clés pénible : déchiffrer les données avec l'ancienne clé, re-chiffrer avec la nouvelle.
- Authentification : dans l'instant. Révocation des clés « facile » : réémettre de nouvelles clés.

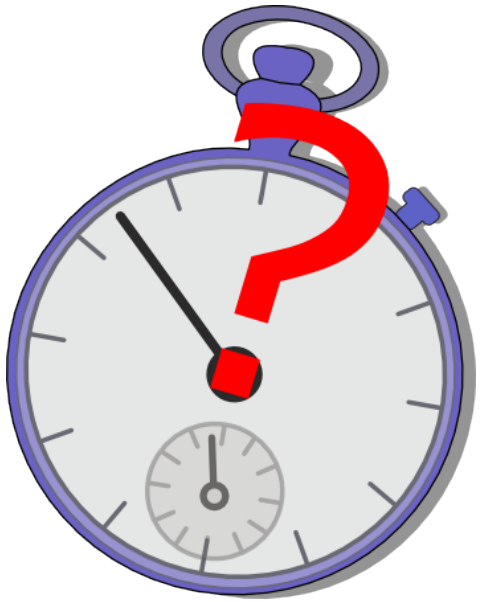
Pas de mélange de genre

Clés de chiffrement, clés d'authentification **doivent être distinctes**. Longueurs de clés pas forcément identiques.

Une erreur courante

- `fichier.sensible` est chiffré avec GPG, clé de 4096 bits, gnagnagna...
- Quand j'ai besoin des données,
  1. je déchiffre le fichier,
  2. je travaille sur le clair,
  3. je re-chiffre le fichier,
  4. j'efface le clair.
- Je suis vraiment très fort.

Où est l'erreur ?



Une erreur courante

- `fichier.sensible` est chiffré avec GPG, clé de 4096 bits, gnagnagna...
- Quand j'ai besoin des données,
  1. je déchiffre le fichier,
  2. je travaille sur le clair,
  3. je re-chiffre le fichier,
  4. j'efface le clair.
- Je suis vraiment très fort.

Où est l'erreur ?

Quelques rappels

Message tout échange d'informations entre deux entités.

Cryptographie Méthodes plus ou moins évoluées afin de rendre un message illisible pour qui ne dispose pas des outils appropriés. Couvre tant les méthodes de chiffrement que celles de déchiffrement.

Cryptanalyse Analyse de messages chiffrés, dont on ne dispose pas des éléments « légitimes » de déchiffrage, pour tenter de les déchiffrer.

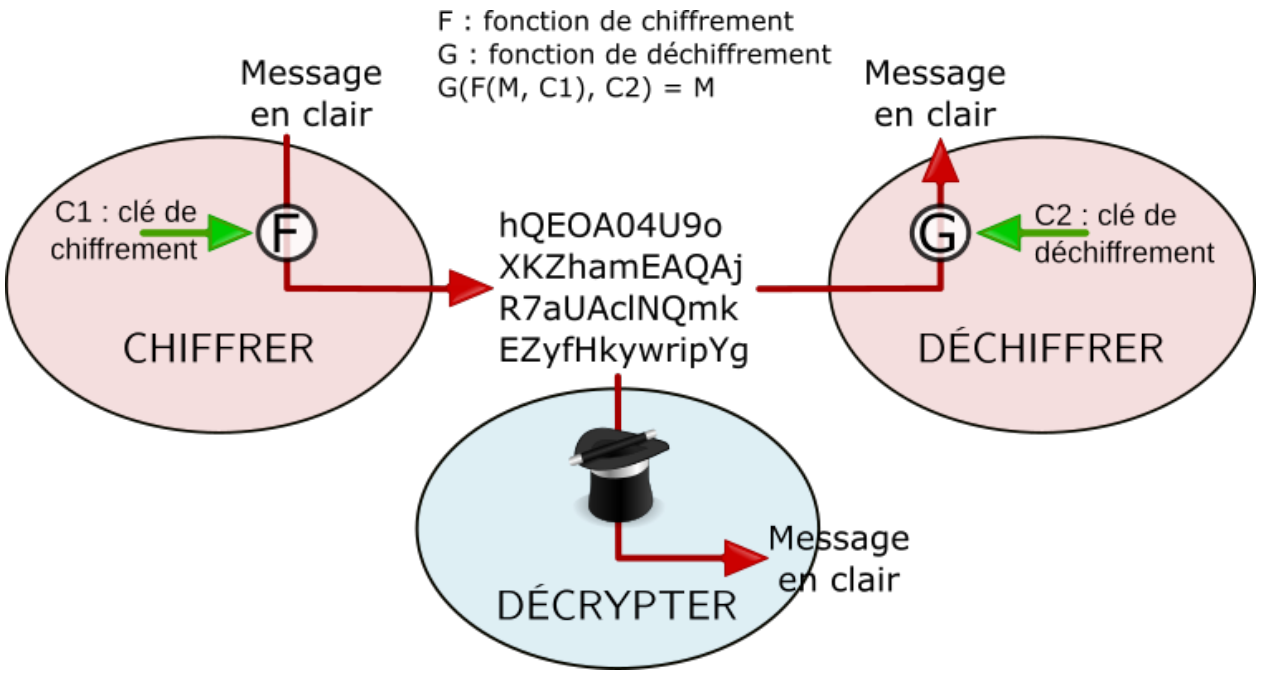
Stéganographie Méthodes visant à cacher un message dans un autre, à le rendre invisible pour qui ne sait pas « où » regarder.

Note

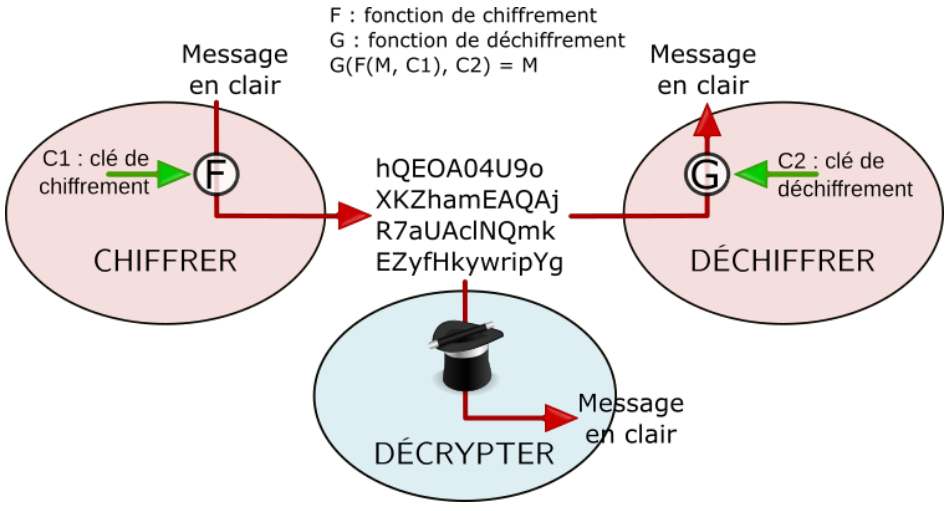
La cryptographie est pratiquement aussi vieille que l'écriture.

Principe général

Principe ancien d'une fonction et de son inverse. Injection secrets (« clés ») dans chiffrement et déchiffrement : $g(f(M, C1), C2) = M$



Symétrique ou asymétrique



- Cryptographie symétrique : clés de chiffrement et de déchiffrement identiques ⇒ secret partagé.
- Cryptographie asymétrique : clés différentes ⇒ plus de secret partagé **mais**
  1. On ne peut pas déchiffrer ce que l'on a chiffré
  2. On ne peut pas re-chiffrer ce qu'on a déchiffré

  (avec les mêmes clés, en tout cas)

Principes de Kerckhoffs résumés

Auguste Kerckhoffs, janvier-février 1883

La cryptographie militaire, Journal des Sciences militaires

- Un bon cryptosystème doit rester sûr même si tout, à l'exception des clés, est connu de l'adversaire.
- Les éléments secrets doivent être les plus faciles à changer (en cas de compromission).

- Difficile de prouver qu'un cryptosystème est sûr
- Des systèmes analysés depuis des années sans trouver de faille s'en approchent probablement
- Modulo nos connaissances et moyens actuels

B&A CONSULTANTS SECURITE INFORMATIQUE

Corrolaires de Kerckhoffs – 1

Un système de chiffrement dont l'algorithme est tenu secret est « mauvais » :

- Il n'a pas été analysé de façon indépendante, voire il n'a pas été analysé
- Chacun est capable d'inventer un algorithme cryptographique qu'il ne sait pas briser (« Loi de Schneier », `https://www.schneier.com/crypto-gram/archives/1998/1015.html#cipherdesign`, `https://www.schneier.com/crypto-gram/archives/1999/0215.html#snakeoil`, `https://www.schneier.com/blog/archives/2015/05/amateurs_produc.html`)

B&A CONSULTANTS SECURITE INFORMATIQUE

Corrolaires de Kerckhoffs – 2

Un système dont la logistique de changement de clés ne peut croître selon le nombre de communiquants est « mauvais ».

- Complexifie révocation/remplacement de clés
- Donc amène à rester « potentiellement compromis »
- Surtout si changement de clés concerne nombreuses personnes indépendantes

Chiffrement symétrique

Principe

Une information (secrète) est partagée entre l'émetteur et le récepteur du message.

- La « clé secrète » peut être un alphabet de substitution (code monoalphabétique), une substitution multiple (Vigenère) ou évolutive (Enigma), un code à usage unique (OTP)
- Les algorithmes modernes (Camélia, AES, IDEA...) reposent sur des transformations importantes du message source, dans l'optique de complexifier toute analyse statistique.
- Problèmes :
  - Transmettre de manière sûre la clé entre les deux correspondants : pas de canaux fiables à 100%.
  - Une clé par couple de correspondants : combinatoire élevée.

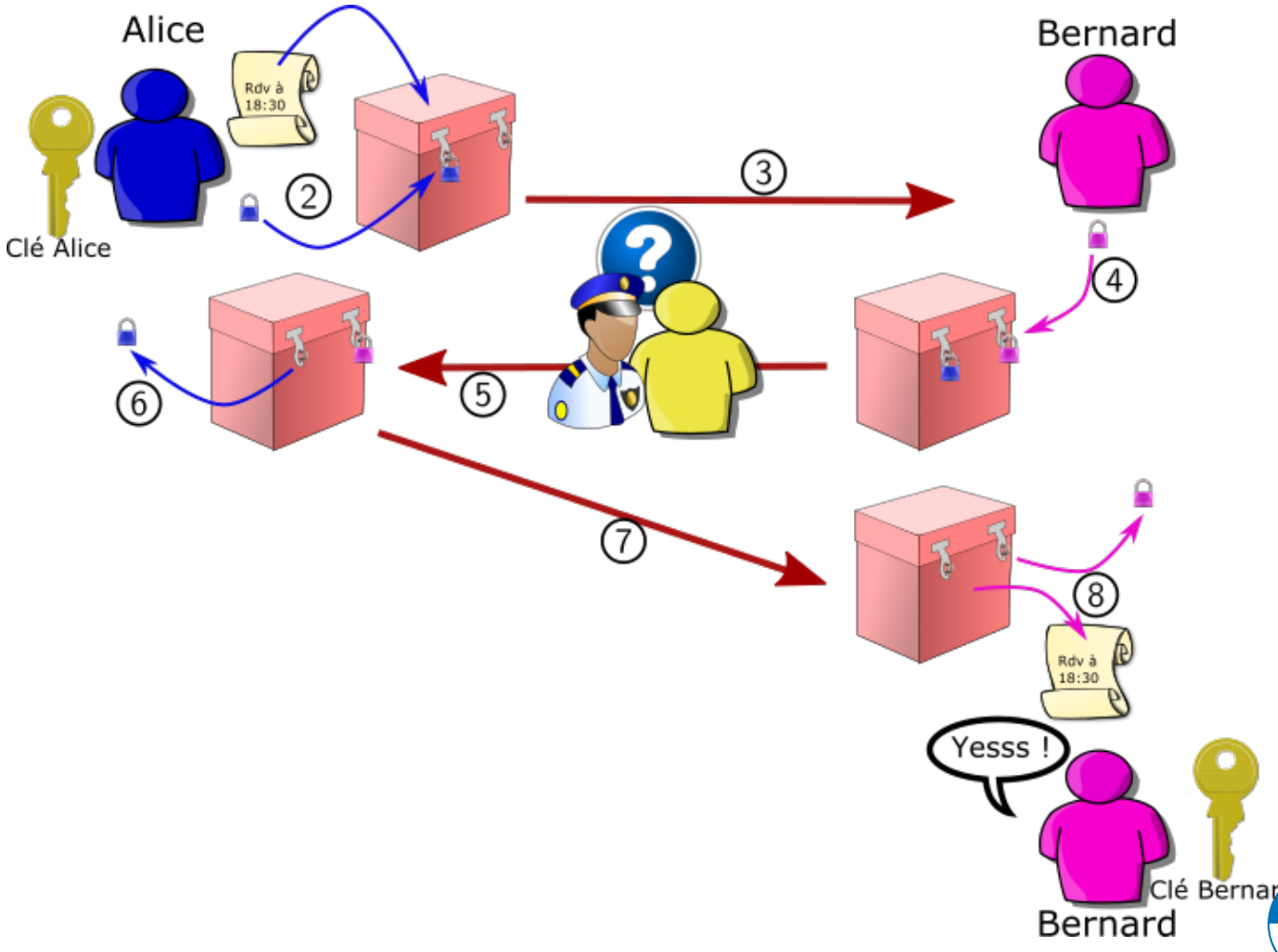
Le point de départ. . .

Contexte : Alice veut envoyer un message secret à Bernard

1. Elle achète un coffret avec 2 anneaux de fermeture
2. Elle met son message dans le coffret et pose son cadenas sur un anneau
3. Elle envoie le coffret à Bernard. . .
4. . . . qui pose son cadenas sur l'anneau libre. . .
5. . . . et renvoie le coffret à Alice. . .
6. . . . qui ôte son cadenas. . .
7. . . . puis renvoie le coffret à Bernard. . .
8. . . . qui retire son cadenas, ouvre le coffret et lit le message.

B&A CONSULTANTS SECURITE INFORMATIQUE

Échange confidentiel sans partage de secret



Dans ce scénario

- Aucun échange préalable entre Alice et Bernard.
- Ils connaissent seulement le protocole à suivre.
- Chacun a gardé la clé (privée) de son cadenas (public).
- Résultat : un échange confidentiel sans synchronisation préalable entre les parties.

Notez bien que

La confidentialité de l'échange est assurée, mais rien de plus et notamment pas l'authentification des parties.
La procédure est peu efficace (trois envois).
L'existence de l'échange n'est pas secrète.

Chiffrement asymétrique

- Règle le problème de transmission des clés.
- Règle le problème de combinatoire.
- On ne sait pas briser les algorithmes associés (problème *difficile*).

Principe de la cryptographie asymétrique

- La clé (bi-clé) est un couple (C1, C2).
- Propriétés spécifiques entre C1 et C2 :
  - Tout ce qui est chiffré par C1 peut être déchiffré par C2 et uniquement par C2, et vice-versa.
  - C2 ne peut déchiffrer que ce qui a été chiffré par C1 et déchiffre tout ce qui a été chiffré par C1 (et vice-versa).
  - On ne peut pas déduire C1 de C2, ni C2 de C1.
- C1 est diffusée librement (publique), C2 est privée (privée).

Avertissements utiles

Quand on utilise de la cryptographie asymétrique

- Il ne faut pas confondre le rôle des clés privées et publiques
- Il ne faut pas s'imaginer que tous les problèmes sont réglés
- Il faut **très bien** protéger sa clé privée.

Sans HSM, la clé privée n'est rien de plus qu'un fichier sur un support de stockage.

Protection de la clé privée

- Mot de passe solide
- Droits d'accès très réduits au fichier
- Sauvegardes multiples, *toutes* protégées par un mot de passe solide.

Hardware Security Module

- Élément matériel contenant les clés privées (et d'autres informations moins sensibles)
- Clés privées ne peuvent être manipulées que par le HSM
- Export des clés privées « impossible »
- Matériel inviolable (destruction des clés)

Conclusions

Un HSM permet une très bonne protection (physique et logique) des clés privées. Surtout utilisé pour les clés privées de « haut niveau » (AC).

Attention

Il **faut** organiser la sauvegarde des clés privées (export « N sur M »), au risque de les perdre en cas de défaillance matérielle.

Exemples de HSM

Gemalto HSM SafeNet (ex-Luna), format 19 pouces rackable

Exemples de HSM

HSM Nitrokey et CardContact, format clé USB

Performances et chiffrement

- Si le chiffrement asymétrique permet de résoudre la problématique de diffusion des clés, il est particulièrement peu performant.
- Le chiffrement symétrique est toujours d'actualité aujourd'hui.
- En fait, on ne fait QUE du chiffrement symétrique, surtout dans des flux de données.

La quadrature du cercle

Echange chiffré/calcul indépendant de la clé de chiffrement symétrique (clé partagée entre émetteur et destinataire, propre à la « session ») par un algorithme asymétrique ⇒ canal sécurisé de transmission de la clé (chiffrement asymétrique) et performances (chiffrement symétrique).

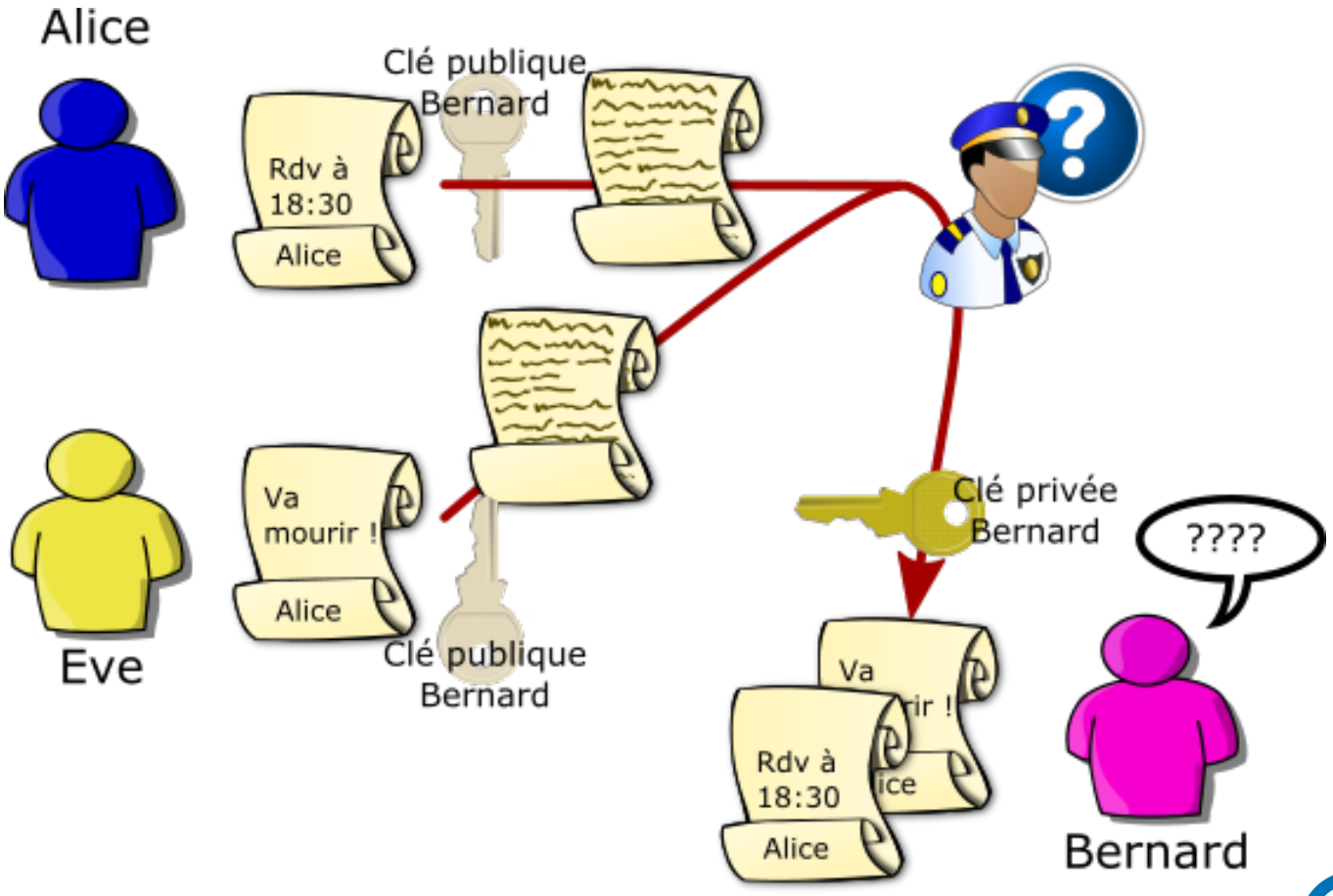
Confidentialité des échanges

Utilisation la plus courante de la cryptographie.

- Alice veut écrire à Bernard.
- Alice récupère la clé publique de Bernard : $pub_{Bernard}$.
- Alice chiffre le message avec la clé publique de Bernard.
- **Seul le propriétaire de la clé privée** $PRIV_{Bernard}$ peut déchiffrer le message.
- Donc, seul Bernard peut déchiffrer le message.
- Mais **n'importe qui** peut l'avoir écrit.

Schéma sur le transparent suivant

Confidentialité des échanges



Authentification d'un message

Utilisation très importante de la cryptographie.

Symétrique Le message a été chiffré par quelqu'un qui connaît la clé de chiffrement

Asymétrique Le message a été chiffré par quelqu'un qui connaît la clé associée à la demi-clé qui déchiffre le message.

Explication

- Message déchiffré avec clé privée $PRIV_{XYZ}$ $\Rightarrow$ A été chiffré avec clé publique pub_{XYZ}. Qui a la clé publique pub_{XYZ} ? Tout le monde $\Rightarrow$ N'identifie rien.
- Message déchiffré avec clé publique pub_{XYZ} $\Rightarrow$ A été chiffré avec clé privée $PRIV_{XYZ}$. Qui a la clé privée $PRIV_{XYZ}$? Uniquement **XYZ** $\Rightarrow$ Identification de l'émetteur.

Authentification d'un message

Dans les faits :

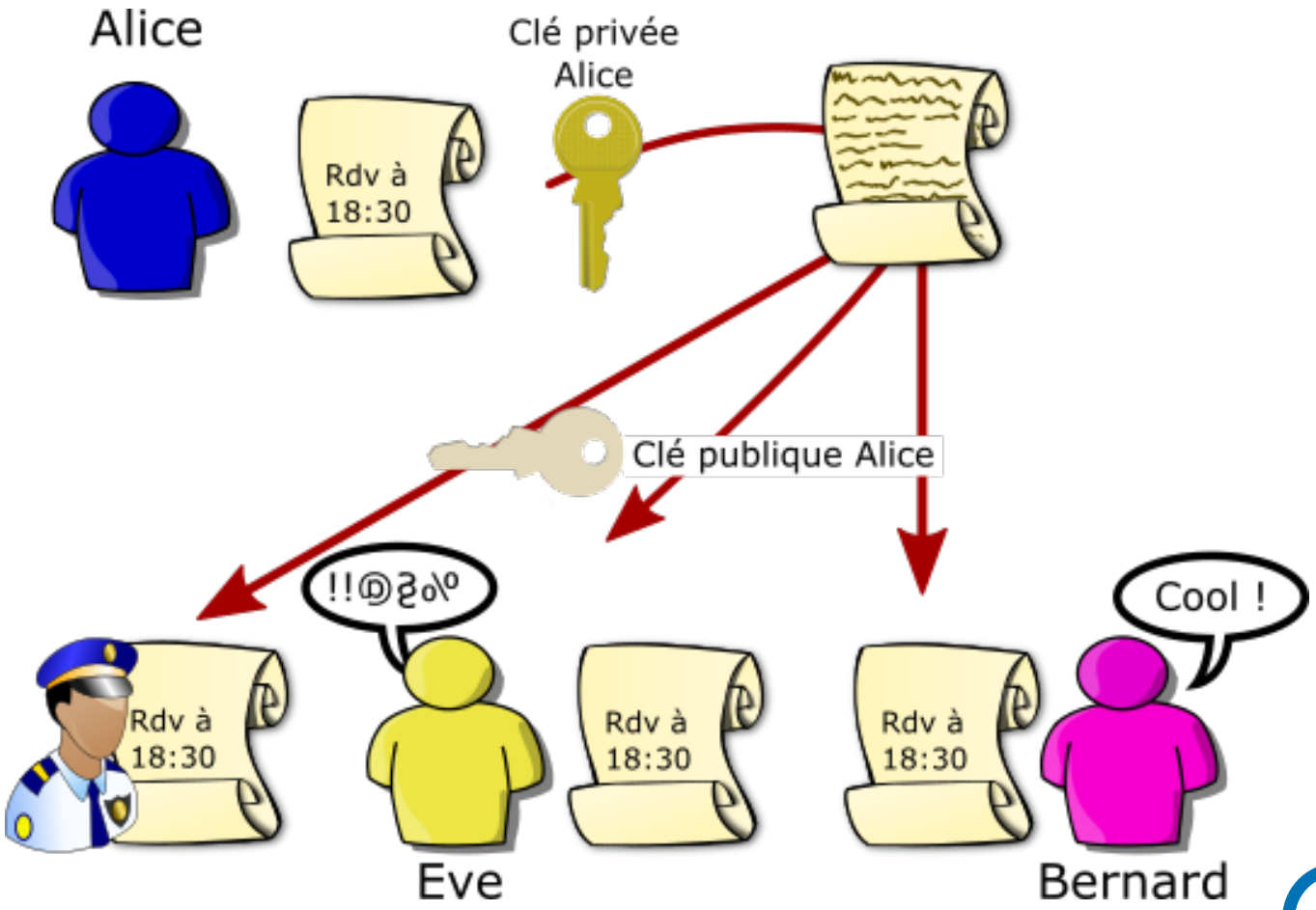
- Alice chiffre un message avec sa clé privée. **Elle est la seule** à pouvoir le faire car elle est la seule à posséder la clé privée.
- **N'importe qui** peut déchiffrer le message, avec la clé publique de Alice.

Notez que

Si le chiffrement utilisé ainsi n'assure aucunement la confidentialité, il garantit malgré tout l'intégrité du message.

Schéma sur le transparent suivant

Authentification d'un message



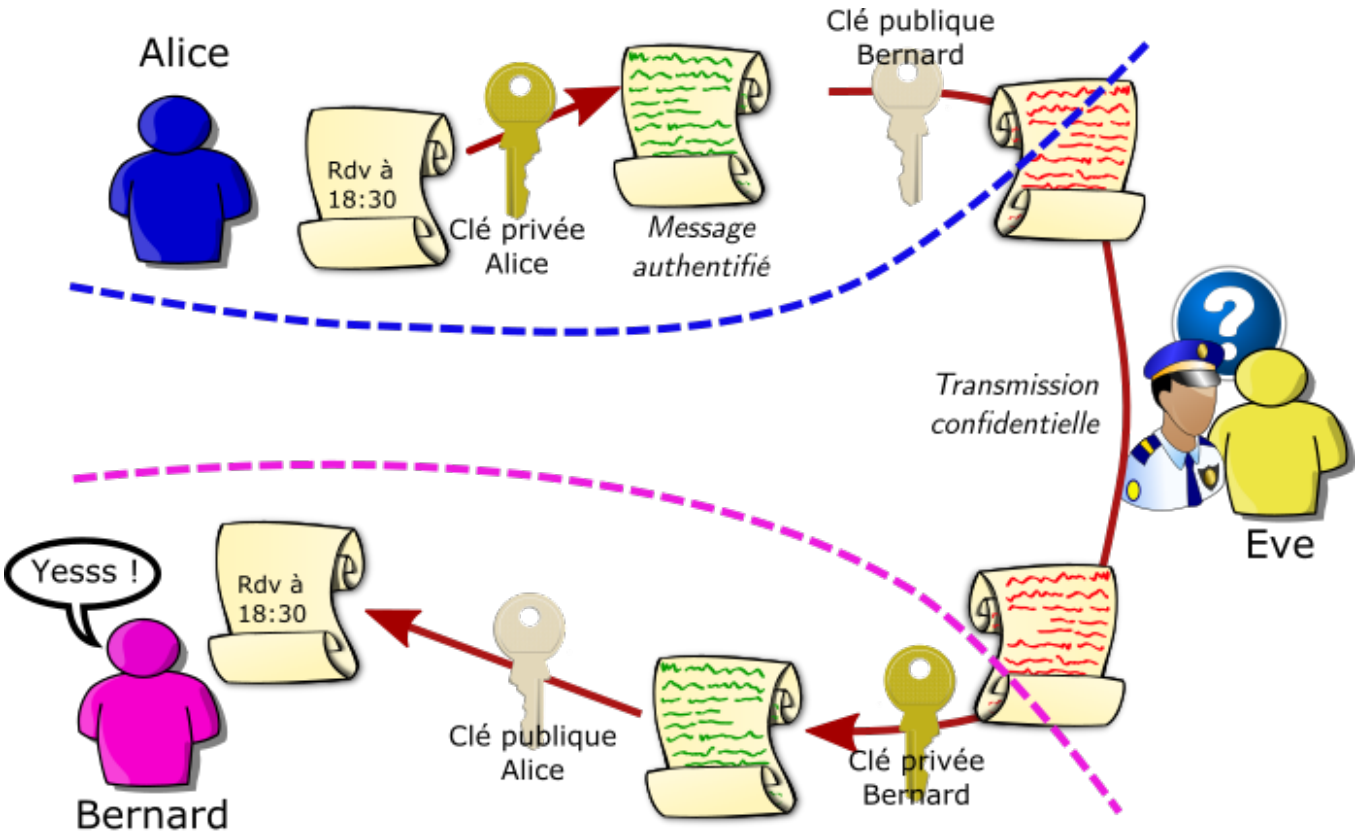
Confidentialité et authentification

S'il est nécessaire d'assurer autant la **confidentialité** que l'**authentification** du message, il suffit d'utiliser les deux techniques précédentes (dans le bon ordre) :

1. Alice chiffre son message avec sa clé privée.
2. Puis elle sur-chiffre le résultat avec la clé publique de Bernard.

- Seul Bernard peut déchiffrer la couche externe ⇒ confidentialité de l'échange.
- Avec la clé publique d'Alice, il déchiffre ensuite le message interne ⇒ authentification de l'émetteur.

Confidentialité et authentification d'un message



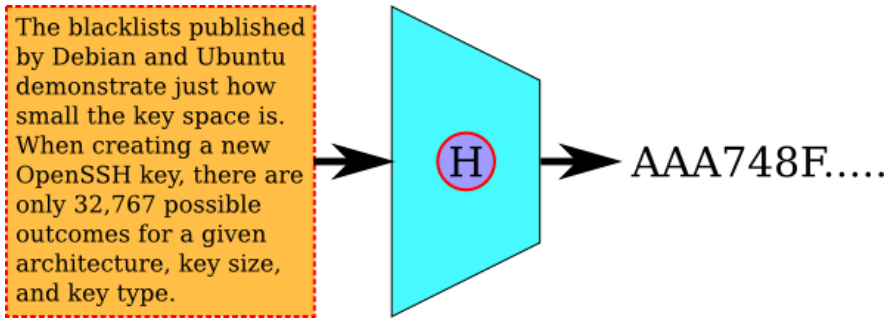
Notez que

Deux opérations de chiffrement/déchiffrement sont nécessaires.

Fonctions de hachage

Fonctions de hachage, condensats, empreintes numériques ou hashcodes. . .

- Moulinette : message de longueur quelconque réduit à une suite de bits de taille fixée (~~MD5~~ : 128 bits ; ~~SHA1~~ : 160 bits ; SHA2 – SHA-224, SHA-256, SHA-384, SHA-512 ; SHA3)
- Transformation non inversible $\Rightarrow$ pas utilisable pour transmettre une information.
- Permet stockage secret (ex : mot de passe) ou preuve connaissance sans transmission du secret.



Fonctions de hachage

Réduction espace infini de messages à espace fini ⇒ collisions inévitables.
Une bonne fonction de hachage doit être :

- Résistante aux collisions
- Résistante aux attaques sur la préimage
- Résistant aux attaques sur la seconde préimage

Principale utilisation des condensats

Stocker secret sans craindre vol (non inversibilité). Typique stockage mots de passe (hachage seul totalement insuffisant !)

Corollaire

Si un outil (« site web ») vous propose de vous renvoyer votre secret ⇒ stocké en clair ⇒ risque majeur de sécurité.

Résistance aux collisions

- Difficile (« impossible ») de trouver deux messages A et B produisant le même condensat.
- Permet d'assurer la « qualité de discrimination » offerte par la fonction de hachage.
- Fonction de hachage peut être utilisée comme empreinte caractéristique du document.

Résistance sur la préimage

- Si l'on connaît un condensat, il est difficile (« impossible ») de trouver un message produisant ce condensat.
- Permet d'assurer « je connais le secret » seulement en transmettant son condensat
- Intercepteur ne dispose que du condensat, ne peut pas en déduire le secret.

Attention !

Transmettre le condensat d'un secret pour preuve de connaissance est une mauvaise idée (*pass the hash*).

Résistance sur la seconde préimage

- Si l'on connaît condensat **et** message associé, il est difficile (« impossible ») de calculer un autre message produisant le même condensat.
- Garantit l'intégrité des informations et autorise l'utilisation du condensat pour les signatures numériques.
- Si on valide un couple (M, H(M)), personne ne peut substituer à M un autre M'.

Signature numérique

Chiffrer un message avec un algorithme asymétrique est coûteux.
Si authentification auteur seulement, gaspillage de ressources.

Optimisation

Plutôt que de chiffrer l'intégralité du message avec sa clé privée, on se contente de le signer.

- Message reste en clair.
- Calcul condensat message (taille fixe, 128 à 512 bits en général).
- Chiffrement condensat avec clé privée émetteur (faible taille, donc rapide) $\Rightarrow$ « signature » du message.
- Condensat chiffré ajouté au message ou transmis à part.

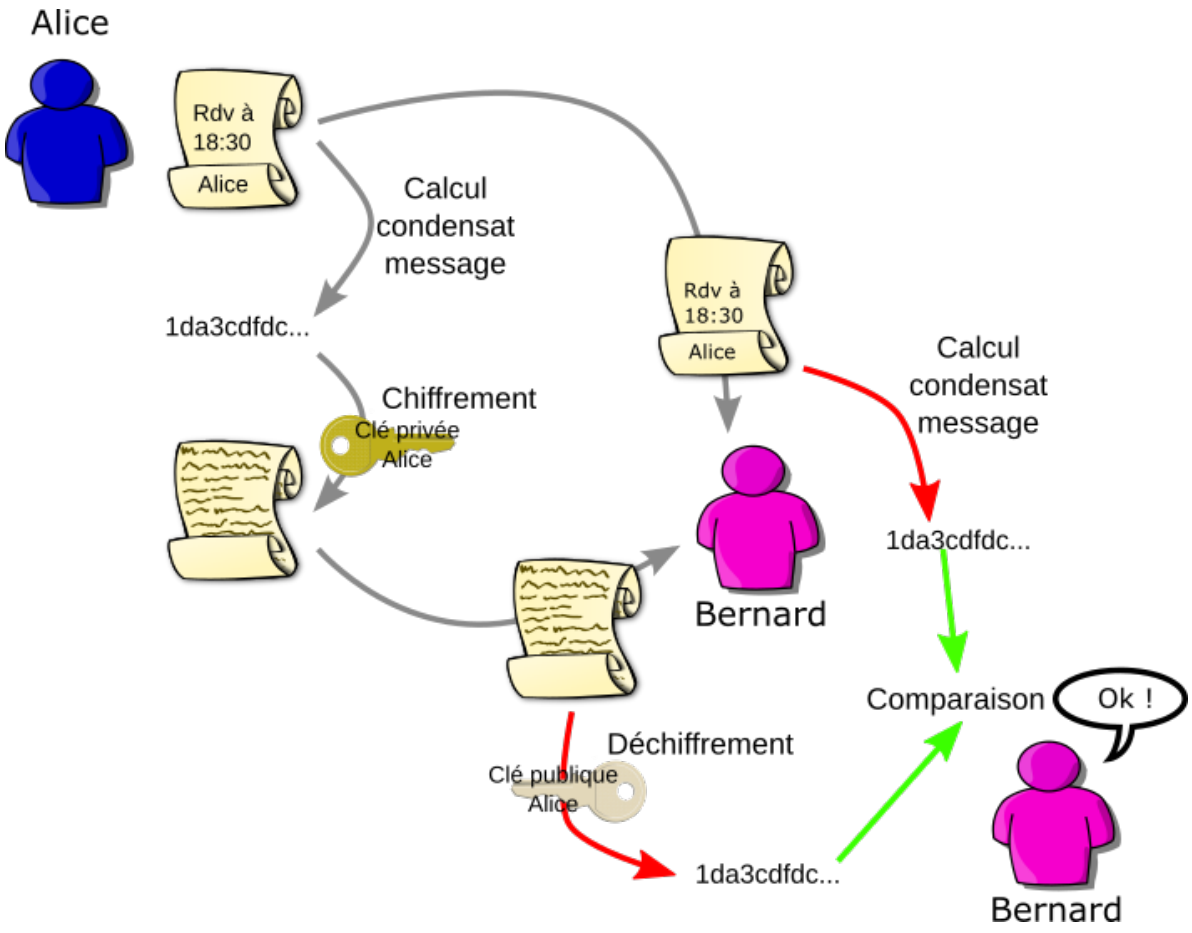
Vérification de la signature

1. Destinataire reçoit message et la signature associée.
2. Calcule condensat message (sans sa signature) $\Rightarrow$ premier condensat
3. Déchiffre (avec clé publique émetteur) la signature $\Rightarrow$ second condensat
4. Compare les deux condensats.

S'il y a identité :

1. le message a bien été émis par Alice ET
2. le message n'a pas été modifié en cours de transmission.

Signature numérique



Bien utiliser la cryptographie

On peut se faire très mal en pensant que la cryptographie est une solution miracle.

- La cryptographie ne protège que des données mortes (non utilisées) $\Rightarrow$ les données sont toujours utilisées sous forme déchiffrée.
- Une mauvaise utilisation de la cryptographie produit une fausse impression de sécurité.

Exemple typique

Notre produit utilise TLS donc il est sûr.

Clés publiques et certificats

La question à se poser

Une clé publique appartient-elle bien à qui elle prétend appartenir ?

Rien ne l'assure a priori.
Nécessité d'établir une chaîne de confiance.

Chaîne de confiance

Généralement par certification des clés publiques (d'où le terme de certificats).

GPG signature (gratuite) par des tiers.

TLS signature (payante) par des autorités de certification.

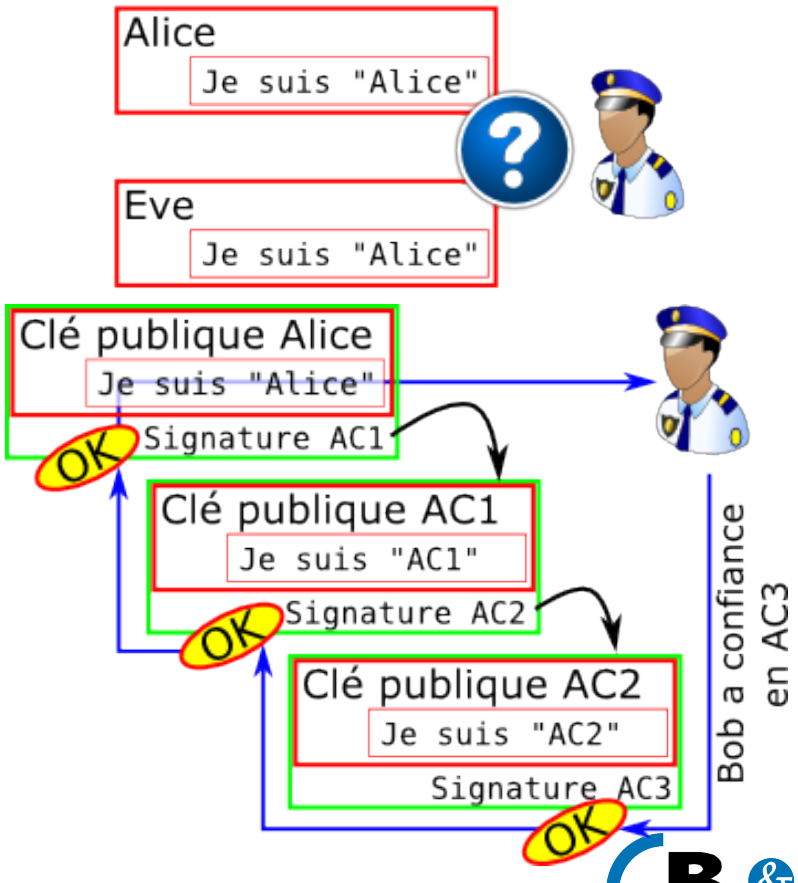
Rien ne vous empêche de monter votre autorité de certification interne.

La chaîne de confiance

Aussi appelée chaîne de certification. Permet (quand elle est bien utilisée) de résoudre le problème de l'authenticité des clés publiques.

Notez bien

Le bon fonctionnement de la chaîne de certification suppose que toutes les AC font parfaitement leur travail.



Utilité de la chaîne de confiance

Inutile de se poser des questions

La chaîne de confiance est vitale. Sa vérification est au cœur du bon fonctionnement de la cryptographie asymétrique.

- Tout outil reposant sur des clés cryptographiques doit les valider en contrôlant l'intégralité de la chaîne de confiance.
- Sinon, il devient facile de briser le chiffrement en interposant un relais dans le flux.

La vérification d'une chaîne de confiance « profonde » peut être complexe (récupération des clés publiques des AC intermédiaires). Certains outils permettent de définir la longueur maximale d'une chaîne de confiance (souvent 1).

B&A CONSULTANTS SECURITE INFORMATIQUE

Avertissement aux lecteurs !

Rappels

- signature cryptographique = condensat du message, chiffré avec la clé privée de l'émetteur.
- certificat = clé publique signée cryptographiquement par une autorité de certification

Si l'AC utilise un algorithme de hachage fragile (MD5, SHA1, personne dans la salle ?), **sa signature peut être contrefaite**. Février 2017, première collision SHA1 complète (deux documents avec même condensat).

Un article intéressant

Septembre 2014
`https://konklone.com/post/why-google-is-hurrying-the-web-to-kill-sha-1`

Gestion de la chaîne de confiance

Il n'y a pas de miracle

Amorçage local : pré-configuration statique clés publiques d'AC.
Doit être possible d'ajouter ou de supprimer des clés publiques (gestion de la confiance).

Les certificats signés par ces AC sont automatiquement validés.

Conséquence immédiate

Avant d'utiliser de la cryptographie asymétrique, **vérifiez** que

1. vos outils savent gérer la chaîne de confiance des certificats.
2. vos développeurs savent le faire, et qu'ils le font,
3. l'AC que vous utilisez est correcte (procédures, algorithmes...)
4. Pas d'AC inconnues dans votre portefeuille

Attention ! ! !

- Toute personne/organisation pouvant ajouter une clé publique dans votre portefeuille de clé **peut sans aucune difficulté** mettre en place un déchiffrement systématique de toutes vos connexions chiffrées.
- Exemples : Nokia 2013, Minéfi 2013, Lenovo 2015 (et autres plus récents)

Le pire

Ce n'est ni compliqué, ni onéreux. Ce peut être difficilement détectable si vous n'avez pas pris quelques précautions.

B&A CONSULTANTS SECURITE INFORMATIQUE

TLS – Transport Layer Security

Objectifs principaux de TLS [2] : assurer confidentialité et intégrité entre deux communicants (client et serveur)

- confidentialité, par usage de cryptographie symétrique
- authentification, par usage de cryptographie asymétrique
- intégrité des échanges, par usage de MAC (message authentication code)

2. Version actuelle 1.2. Versions précédentes vulnérables.

TLS, confidentialité et authentification

- Secret partagé négocié entre communicants
- Algorithmes et clés négociés avant toute transmission de donnée
- Négociation faite sans risque d'interception ou d'altération
- Authentification (réciproque ou non) via certificats

Perfect Forward Secrecy

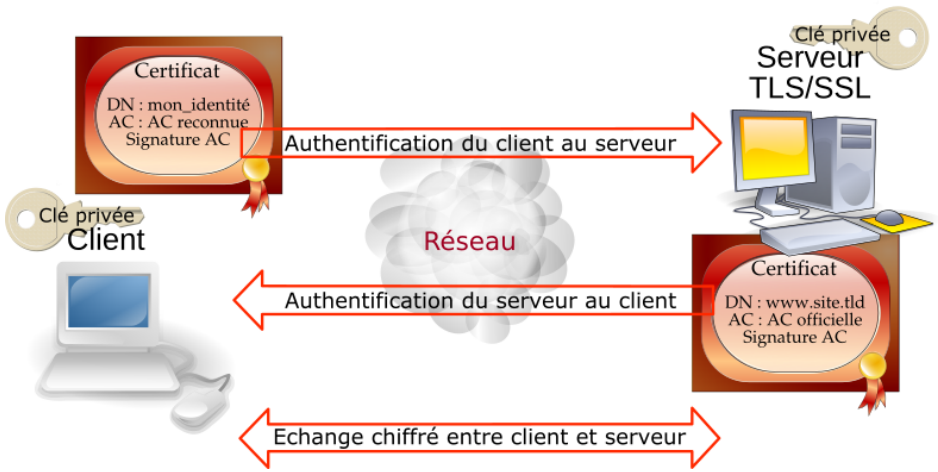
Caractéristique selon laquelle la compromission des clés privées ne compromet pas les sessions précédentes, même interceptées.
Algorithmes : DHE-RSA [a], ECDHE-RSA, DHE-DSS, ECDHE-ECDSA, DHE-PSK, ECDHE-PSK.

a. DHE : Diffie-Hellman éphémère

TLS v1.3

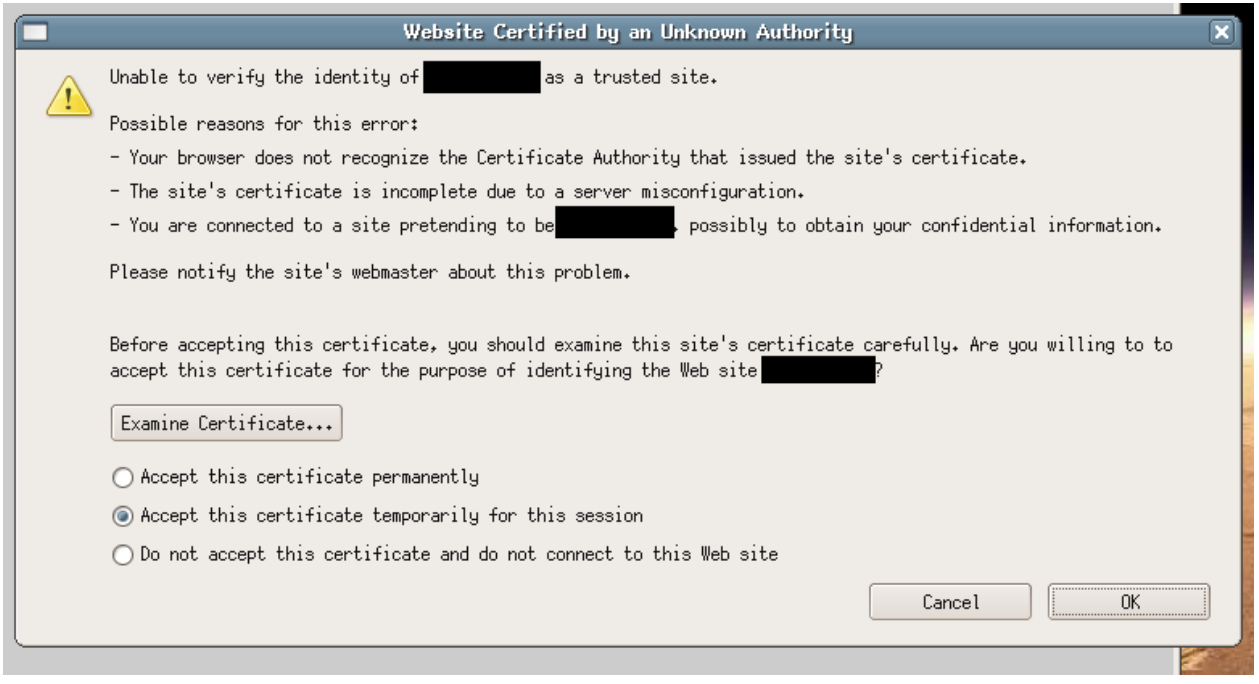
- Future version de TLS, en cours d'élaboration (https://tools.ietf.org/html/draft-ietf-tls-tls13-12)
- Elimination de tous les algorithmes n'assurant pas la PFS
- Algorithmes valides : DHE-RSA, ECDHE-RSA, ECDHE-ECDSA

Principe de TLS



1. Les deux communicants échangent leurs clés publiques (souvent, le serveur est le seul à envoyer sa clé publique).
2. Et peuvent dialoguer de manière sécurisée.
3. En théorie (faiblesses non-cryptographiques permettant le déchiffrement).

Vérification du certificat TLS



Non validation de la signature du certificat

Poser la question à l'utilisateur. On suppose/espère qu'il sait ce qu'il faut faire.

Serveurs et TLS

Un problème particulier

Serveur utilisant TLS (web, messagerie, VPN, etc.) :

- Doit démarrer automatiquement.
- Doit accéder à la clé privée de chiffrement TLS.
- Clé privée protégée par un mot de passe solide.

Aucune bonne solution à ce problème :

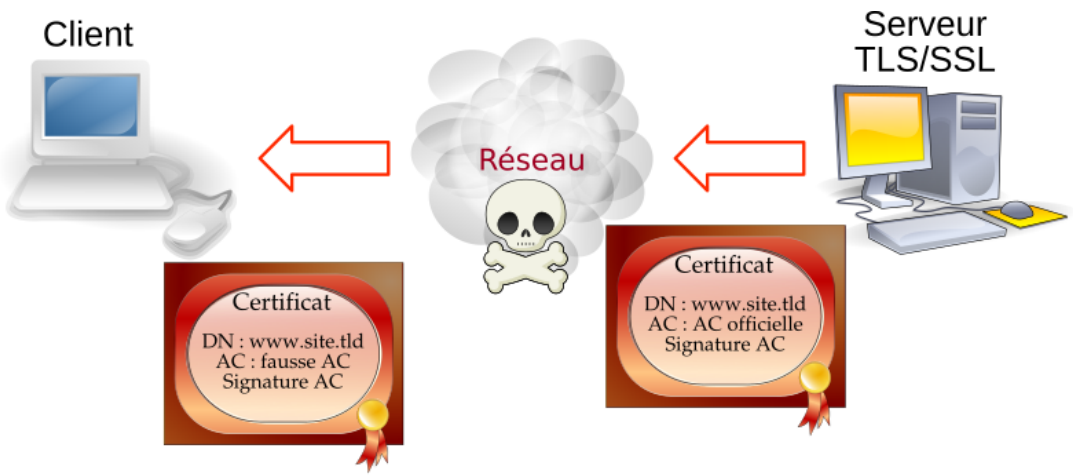
- Serveur ne démarre pas automatiquement (très rare),
- Outil qui « fournit » mot de passe (autres problèmes),
- Suppression du mot de passe (le plus courant).

En conséquence

Protection du fichier contenant la clé privée – y compris sur les sauvegardes et partout où elle pourrait être dupliquée.

Sans validation des certificats

L'étape de validation (complète, y compris chaîne des AC) d'un certificat est critique.

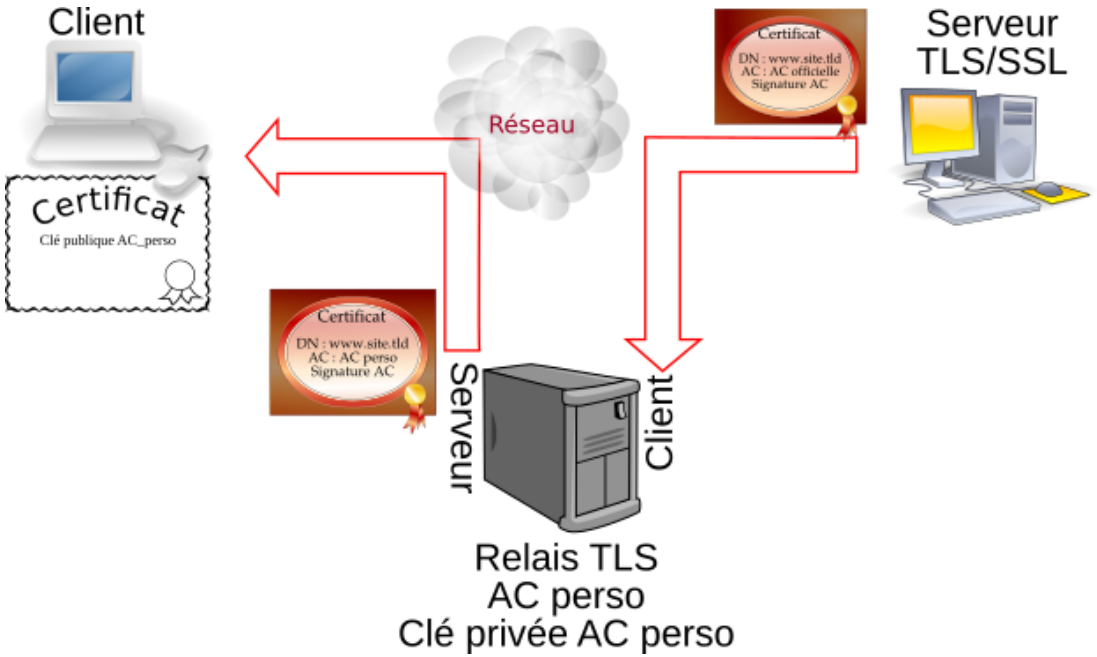


Sinon...

- On s'interpose dans le dialogue (MITM).
- On envoie au client notre propre certificat (généré spécifiquement).
- Et le tour est joué.

Avec validation des certificats

Ce n'est pas miraculeux pour autant.



Relais de décodage « légitime ».
Fonctionne dès maîtrise plate-formes clientes ou contrôle certificat de signature approprié.

Conclusion

Même quand on utilise correctement de la vraie cryptographie,

1. Sur un réseau que vous ne contrôlez pas,
2. Avec un poste que vous ne contrôlez pas,

vous ne contrôlez pas toujours la transaction chiffrée.

SAUF

1. si vous avez noté au préalable l'empreinte digitale du certificat de votre correspondant,
2. quand vous étiez sûr que c'était lui,
3. et que vous faites l'effort de la vérifier.

ou bien

Client utilisation observateurs externes (type Perspectives)

Serveur Certificate Pinning

Déchiffrement TLS

Cybercafés, boîtiers (légitimes) de déchiffrement TLS d'entreprise, interceptions légales...

Politique de sécurité

Pas de flux chiffrés chez moi.
Contrôle systématique du contenu des échanges.

Problèmes

- Le contenu intégral de vos échanges chiffrés est analysé.
- Si authentification de l'utilisateur (certificat personnel : TVA, impôts, sites web), confier sa clé privée au boîtier, sans mot de passe.
- Statut juridique de ces boîtiers ?

Plan

B&A CONSULTANTS SECURITE INFORMATIQUE

Infrastructure de gestion de clés

Infrastructure à Clés Publiques (ICP), Infrastructure de Gestion de Clefs (IGC), Public Key Infrastructure (PKI).
Ensemble de

- Composants physiques (ordinateurs, équipements cryptographiques, cartes à puces)
- procédures humaines (vérifications, validations)
- et logiciels (système et application)

Objectif

Gérer le cycle de vie des certificats.

Sauf que...

Construction procédurale, « humaine », pour tenter de pallier les problèmes de confiance associés aux clés publiques. Ca marche bien, avec de **très nombreuses situations de défaillance totale**.

Services d'une IGC

Pré-requis

Confiance absolue dans l'intégrité/fiabilité de l'AC

- Production, renouvellement, révocation et publication de certificats,
- Publication des listes de révocation,
- Enregistrement, identification et authentification des utilisateurs ou éléments techniques.

L'IGC peut « boucler sur elle-même » : identification et authentification des utilisateurs qui se servent de l'IGC.

Exemples de besoins pouvant amener à une IGC

- Contrôle d'accès à des applications, systèmes, locaux (badges, cartes, etc.).
- Authentification d'utilisateurs (contrôle accès ou d'habilitations)
- Identification croisée d'équipements communiquants.
- Authentification de messages ou documents.
- Chiffrement de communications traversant des réseaux non sûrs (VPN).

Une IGC peut aussi...

Quelques options plus dangereuses :

- archivage des certificats,
- séquestre et
- recouvrement des clés.

Elles sont dangereuses car les clés privées ne sont plus aux mains de leur seul porteur $\Rightarrow$ on peut commencer à mettre en doute la signature (répudiation, intégrité) et le chiffrement (confidentialité).

Recouvrement des clés privées

- Quelques cas d'école :
  - utilisation de cryptographie pour le chiffrement de documents
  - perte de la clé privée
  - comment accéder aux documents chiffrés ?
- Autres cas classiques :
  - indisponibilité du collaborateur
  - oubli du mot de passe de la clé privée
  - départ du collaborateur.

Déjà évoqué par ailleurs

La cryptographie, c'est dangereux quand on se rate !

B&A CONSULTANTS SECURITE INFORMATIQUE

Recouvrement des clés

- Problème difficile sur deux plans :
  - La duplication des clés privées n'est jamais souhaitable
  - Pas plus que la perte de données qui ont été chiffrées.
- Il n'existe pas de « bonne » solution, toutes sont des compromis
  - Entre la sécurité offerte par la cryptographie
  - Et la continuité d'activité nécessaire à l'entreprise.

Tiers de séquestre

- Le tiers de séquestre dispose des clés privées d'un ensemble d'utilisateurs :
  - Pour la restauration (en cas de perte de la clé privée)
  - Pour le déchiffrement SANS la participation du porteur (procédures judiciaires... ou moins officielles)
- Le tiers doit être capable d'assurer INDEFINIMENT la confidentialité des clés privées qui lui sont confiées.

Plan

Plan

Composition d'une IGC

Autorité de certification Signature des certificats et des listes de révocation.

Autorité d'enregistrement Production des certificats et vérification des identités des porteurs.

Autorité de dépôt Stockage des certificats et listes de révocation.

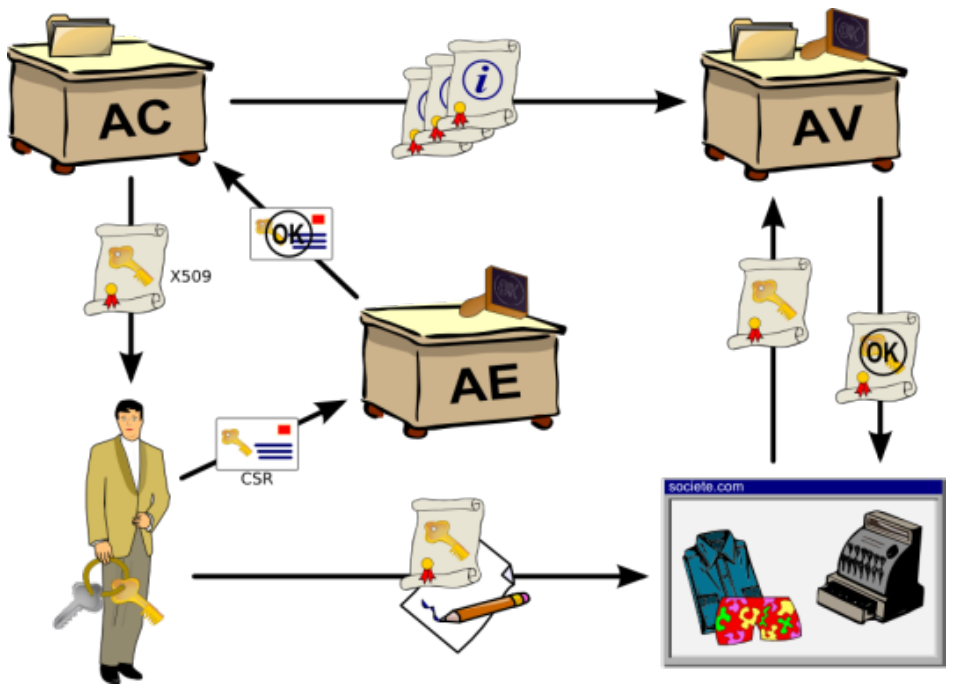
Entité d'extrémité Porteur du certificat.

Note d'optimisation

Regroupement possible AC et AE (petites/moyennes IGC)

B&A CONSULTANTS SECURITE INFORMATIQUE

Dynamique de fonctionnement



CSR : demande de certification (Certificate Signing Request).
X509 : clé publique signée, certificat au format X509.
AV : autorité de validation.

Séparation des rôles

- Les responsabilités doivent être claires et distinctes.
- L'autorité d'enregistrement est critique dans les contrôles qu'elle réalise.
- Fusion possible de l'AC et de l'AE.
- Bi-clé et demande de certificat peuvent être produits par l'EE (bi-clé devrait toujours l'être)
- La rigueur la plus ABSOLUE est nécessaire.

L'autorité de certification

- Composant fondamental de la chaîne de confiance.
- Lorsqu'une AC certifie une clé publique, **elle s'engage sur l'identité** du porteur de la clé.
- L'AE doit avoir réalisé **tous** les contrôles nécessaires.
- L'AC est civilement engagée en cas d'erreur et de préjudice lié à cette erreur.
- L'AC doit gérer les révocations et renouvellements de certificats.

Certification d'une clé publique

En signant une clé publique, l'autorité de certification « déclare » que cette clé publique appartient bien à qui elle prétend appartenir.

Cas typique

Certificat de chiffrement pour un serveur Web.

Objectif

Garantir l'identité du porteur (de la clé privée associée) pour ceux qui consomment la clé publique.

Validité d'un certificat

- Un certificat est valide pour une certaine période de temps :
  - Date de début d'utilisabilité
  - Date de fin d'utilisabilité
- La durée de vie est fixée par l'autorité de certification, pas par le porteur.

  Not Before : May 6 12 :56 :14 2016 GMT
  Not After : Jun 25 12 :56 :14 2016 GMT

Remarque anodine

Rien n'oblige une AC à produire des certificats valides exactement une année. Sauf l'intérêt économique bien compris.

Révocation et suspension

Composante importante d'une IGC.

- Suspension = révocation temporaire d'un certificat (« on hold »).
- Il peut être nécessaire d'invalider (souvent « très rapidement ») un certificat, par exemple :
  - Vol ou perte d'un équipement disposant d'une possibilité de connexion au VPN de l'entreprise.
  - Départ d'un collaborateur.
  - Changement de fonction d'une EE.
- Révocation mal (ou pas) gérée ⇒ maintenir voire créer des accès non autorisés.

Les CRLs

Gestion des révocations via des listes de révocation de certificats (CRL) :

- Produites par l'AC
- Signées par l'AC
- Diffusées ou téléchargées par les outils de l'IGC
- Contrôlées durant la phase de validation d'un certificat.

Attention

Les clients de l'AC doivent (théoriquement) récupérer les CRLs à jour, à la fréquence de diffusion de l'AC (globalement inconnue : c'est au besoin).

Plan

Principe

- `https://letsencrypt.org/`
- Favoriser l'adoption massive de TLS, notamment pour les serveurs Web
- en automatisant **toutes** les phases de gestion des certificats
- y compris le déploiement sur le serveur Web et sa configuration
- et le renouvellement des certificats

En résumé

TLS pour le serveur *Fan de Twilight* de mes nièces.

Une brève histoire du temps

- 18 novembre 2014 : lancement
- 14 septembre 2015 : signature du premier certificat
- 15 octobre 2015 : version béta privée (sur invitation)
- 19 octobre 2015 : certificat racine `Let's Encrypt` signé par Iden Trust
- 3 décembre 2015 : version béta publique
- avril 2016 : mise en production
- juin 2017 : 100 millionième certificat signé

Les versions béta

Limitation du nombre de requêtes par domaine (5/semaine), par IP (10/3 h), par compte (300/semaine)

Fonctionnement classique

- AC comme les autres (gratuite)
- Apporter une « preuve de propriété » du nom concerné
- Tout le reste est (pratiquement) automatique :
  - Demande du certificat
  - Configuration du serveur Web (Apache, ngnix)
  - Renouvellement
- Révocation possible

Attention

Nous n'avons pas testé la configuration automatique du serveur Web.

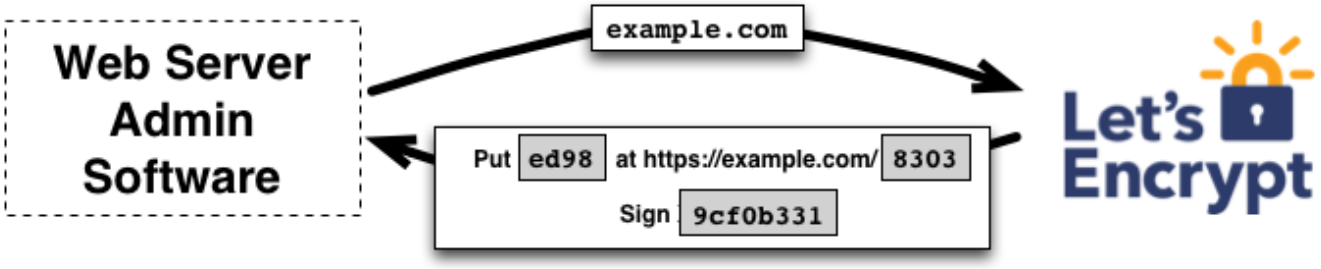
Automatic Certificate Management Environment

- Protocole ACME : `https://github.com/letsencrypt/acme-spec` et `https://github.com/ietf-wg-acme/acme/`
- Lors du premier échange avec serveurs ACME
  - envoi clé publique client
  - information d'identité (adresse électronique)
- Deux principales étapes
  - Preuve de propriété du nom (Domain Validation)
  - Gestion du certificat (soumission, renouvellement, révocation)
- Message envoyé par serveurs ACME un mois avant expiration certificat

B&A CONSULTANTS SECURITE INFORMATIQUE

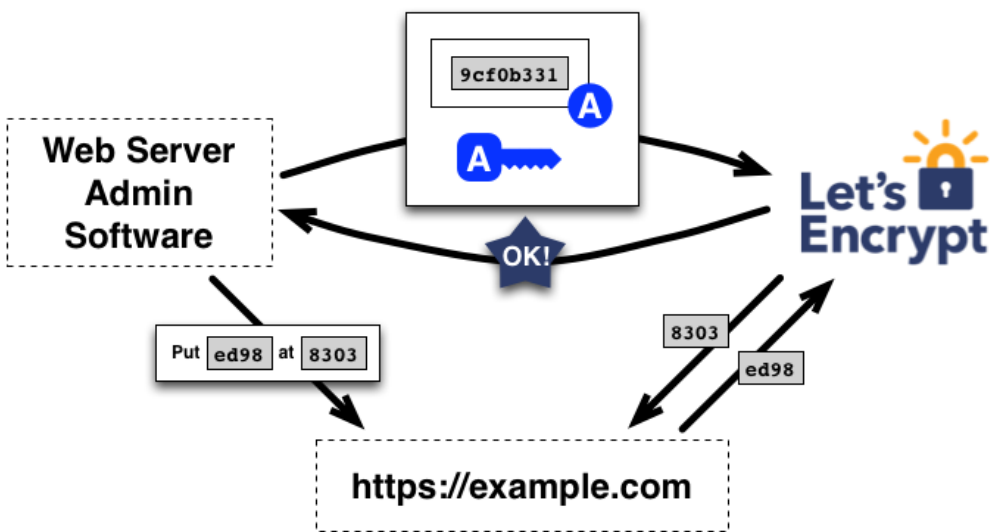
La preuve de propriété

- Via un enregistrement spécifique dans le DNS (théoriquement)
- ou via une URL spécifique
- le « spécifique » étant défini par le serveur ACME à l'enrôlement
- le serveur LE va récupérer ce challenge signé par la clé privée du client
- et le valide (ou pas) avec la clé publique reçue au préalable



La preuve de propriété

- Via un enregistrement spécifique dans le DNS (théoriquement)
- ou via une URL spécifique
- le « spécifique » étant défini par le serveur ACME à l'enrôlement
- le serveur LE va récupérer ce challenge signé par la clé privée du client
- et le valide (ou pas) avec la clé publique reçue au préalable



Premier inconvénient

- Validation par URL ⇒ accès à l'URL
- Donc serveur doit être accessible depuis Internet
- Ennuyant pour systèmes purement internes
- Et pour serveurs non-HTTP
- Peut demander quelques « adaptations » pour la phase de preuve de propriété

Validation via DNS

Posera le même type de difficulté : exposition temporaire nom privé

Certificats publiés

Certificats produits par LE sont publiés (`certificate-transparency.org` et `crt.sh`). Quid des serveurs privés ?

Certificats produits

https://crt.sh/?Identity=<regExp SQL>&iCAID=7395

crt.sh Identity Search

Criteria	Identity LIKE '%[redacted]%'; Issuer CA ID = 7395

Issuer Name	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X1		
Certificates (2)	Not Before	Not After	Subject Name
	2015-12-05	2016-03-04	CN=[redacted]
	2015-12-05	2016-03-04	CN=[redacted]

Clients ACME

- Plus de client officiel
- LE recommande CertBot (https://certbot.eff.org/)
- Nombreux clients alternatifs (shell, C, etc.)
- Voir https://letsencrypt.org/docs/client-options/
- Reste à valider les composants – surtout si exécution `root`
- Qualité de codage variable

Certificats multi-noms

- Extension `Subject Alt Name` reconnue
- Il suffit de donner plusieurs noms sur la ligne de commande
- Chacun de ces noms doit être validé

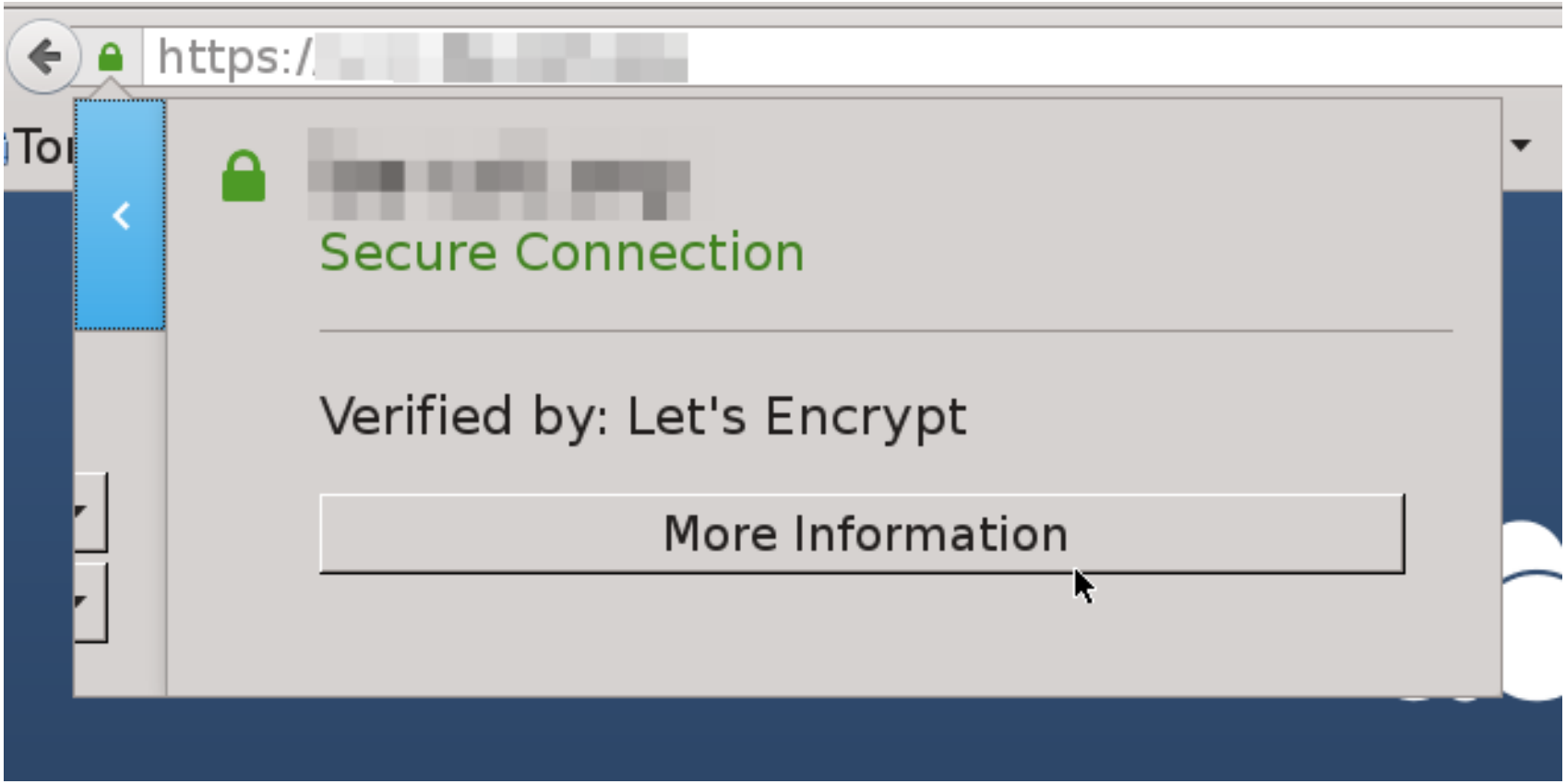
Ca donne quoi ?

```
Make sure your web server displays the following content at
http://[redacted]/.well-known/acme-challenge/TSJc6d9-Ce6n60FlcEIEbPHAcuFkxGw
TT2cP7VqL798 before continuing:

TSJc6d9-Ce6n60FlcEIEbPHAcuFkxGwTT2cP7VqL798.FV5YhedqScD-c8Y1gHbGCvUJUduG6xLveJSf
4kSUEmQ

Content-Type header MUST be set to text/plain.

If you don't have HTTP server configured, you can run the following
command on the target server (as root):

mkdir -p /tmp/letsencrypt/public_html/.well-known/acme-challenge
cd /tmp/letsencrypt/public_html
printf "%s" TSJc6d9-Ce6n60FlcEIEbPHAcuFkxGwTT2cP7VqL798.FV5YhedqScD-c8Y1gHbGCvUJ
UduG6xLveJSf4kSUEmQ > .well-known/acme-challenge/TSJc6d9-Ce6n60FlcEIEbPHAcuFkxGw
TT2cP7VqL798
# run only once per server:
$(command -v python2 || command -v python2.7 || command -v python2.6) -c \
"import BaseHTTPServer, SimpleHTTPServer; \
SimpleHTTPServer.SimpleHTTPRequestHandler.extensions_map = {'': 'text/plain'}; \
s = BaseHTTPServer.HTTPServer(('', 80), SimpleHTTPServer.SimpleHTTPRequestHandle
r); \
s.serve_forever()"
Press ENTER to continue
```

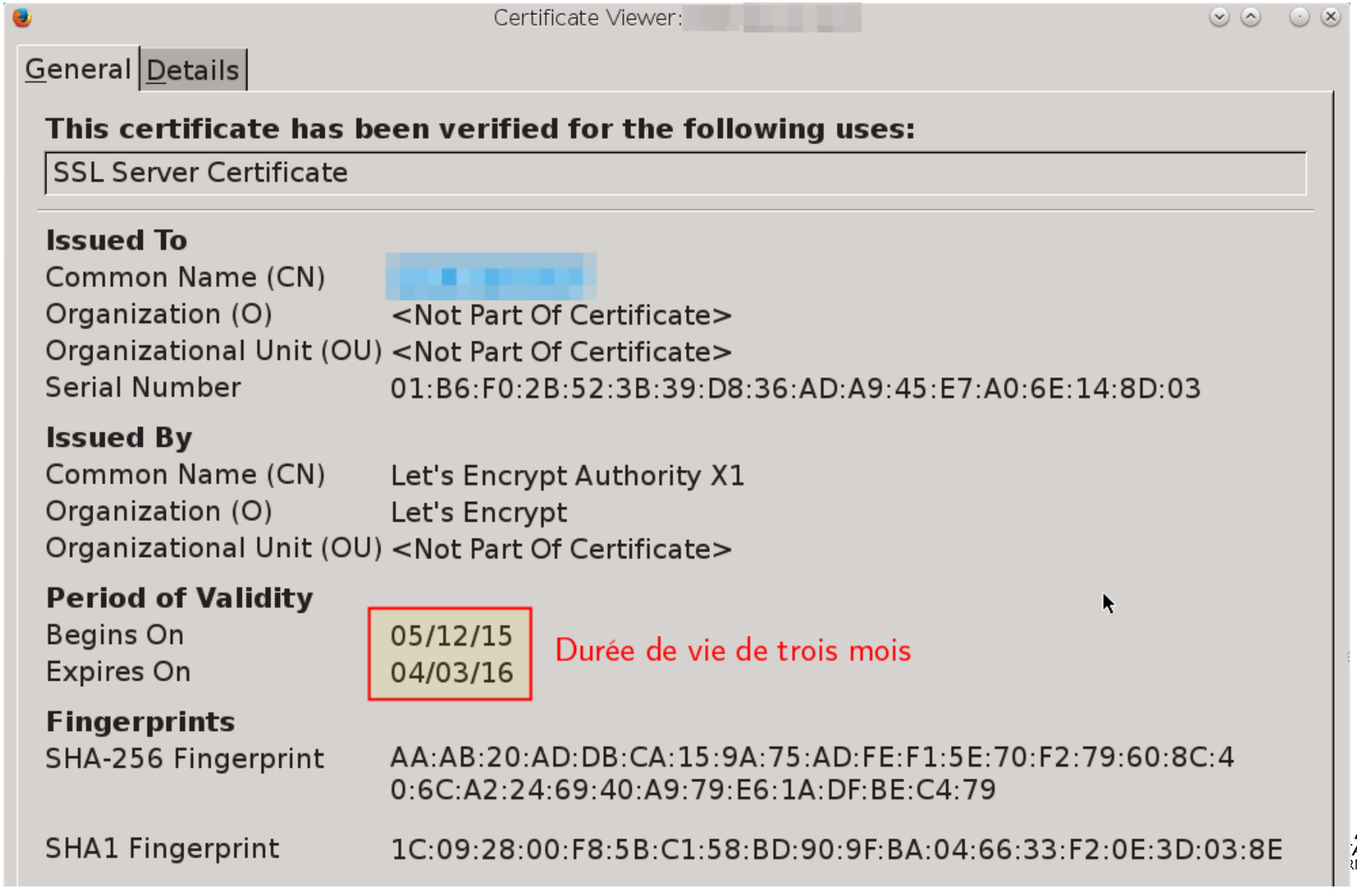
Ca donne quoi ?

```
IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at
   /home/pyb/etc/letsencrypt/live/[redacted]fullchain.pem. Your
   cert will expire on 2016-03-04. To obtain a new version of the
   certificate in the future, simply run Let's Encrypt again.
pyb@veterini ~ $
```

Ca donne quoi ?



Ca donne quoi ?



Ça marche !

- Automatiser contrôle renouvellement (`cron` hebdomadaire)
- Attention si déploiement automatique certificats (`root` ? reprise sur incident ?)
- Etudier semi-automatique (renouvellement auto, déploiement sous contrôle) ?

Encore du travail

- Validation via DNS devrait marcher
- OCSP (passage à l'échelle ?)
- Serveurs ACME internes, pour systèmes privés

LetsEncrypt va-t-il tuer les AC classiques ?

- Avenir sombre pour AC qui n'apporte pas de plus-value à la production de certificats
- Pour les autres besoins...
  - Pas encore de certificats joker (`*.mondomaine.amoi`)
  - Pas de certificats « Organization Validated » ou « Extended Validation »

À savoir

Pas d'obligation d'utiliser les serveurs de LE. Il suffit d'accéder à un serveur respectant le protocole ACME.

Plan

Politique de certification

- Mise en oeuvre par l'Autorité d'Enregistrement.
- Document nécessaire pour le bon fonctionnement de l'IGC.
- Ensemble de règles définissant les exigences auxquelles l'AE se conforme pour la vérification des CSR qui lui sont soumis.
- Règles très étendues (y compris facturation éventuelle, responsabilités, etc.)

Contenu général

La politique de certification définit :

- Obligations et responsabilités
  - De l'AC
  - De l'AE
  - Des porteurs de certificats (EE)
  - Du responsable de la certification (intermédiaire entre EE et AE)
- Disposition juridiques applicables
- Contrôles de conformité réalisés
- Tarifs (éventuels) des différentes opérations

Obligations typiques

- Respecter les règles d'utilisation des certificats (AC, AE, EE)
- Accepter les contrôles et leurs résultats (AC, AE)
- Gérer les listes de révocation (AC)
- Vérifier les identités des EE (AE)
- Choisir des phrases de verrouillage solides (EE)
- Fournir une identité correcte (EE, RC)

Vérifications faites par l'AE

- Varient grandement en fonction des situations et organisations.
- Règles systématiques :
  - Nommage (CN surtout),
  - Valeurs des champs du Sujet (C, ST, O, OU...)
  - Règles de vérification de l'identité du porteur
- En cas de doute, un CSR n'est pas transmis à l'AC.

Note

Dans la réalité, les AE transmettent trop souvent des CSR « douteux » à leur AC.

Vérification de l'identité

- Procédures qui permettent de garantir qu'un certificat « appartient bien » à son propriétaire logique (DN/CN)
- Deux grandes techniques :
  - Demande de documents spécifiques
  - Rendez-vous face à face
    - direct (entité mono-site) ou
    - avec un responsable intermédiaire (Responsable Certification sur chaque site)

Attention !

Si les certificats sont trans-entités, s'assurer que toutes les AE font un travail de même qualité.

Exemple : demande de certificat TLS web

Apporter la preuve du droit d'utilisation du nom (DN) :

- Kbis (société), copie inscription INSEE (profession libérale), copie passeport (particulier)...
- Preuve de droit sur le nom de domaine (whois, lettre de pouvoir)
- Preuve de coordonnées physiques (adresse, téléphone).

Il peut y avoir bien d'autres contrôles.

Clauses de révocation

- Doivent être définies dans la Politique de Certification.
- Doivent être connues de tous les acteurs.
- Supposent la mise en place de procédures
  - de contrôle (vérifications de conformité) et
  - d'alerte (soupçon de compromission, perte)
- Nécessitent la vigilance des porteurs.
- Ne doivent pas être perçues comme des sanctions.

Exemples de révocations

- Compromission, perte ou vol d'une clé
- Modification de l'identité de l'EE
- Cessation d'activité de l'EE
- Changement d'affectation de l'EE
- Non-respect des obligations d'utilisation (sanction)

Procédure d'engagement

La révocation doit être validée avant d'être entérinée.

- Validation par l'AE :
  - Origine de la demande de révocation et autorité à demander cette révocation
  - Certificat à révoquer
- Transmission à l'AC pour inclusion dans les listes de révocation.
- Production éventuelle nouveau certificat basé sur une nouvelle clé privée.

Plan

Plan

Le diable est dans les détails

De façon très simplifiée, les modèles de confiance en informatique fonctionnent :

- Selon un mode décentralisé, de type pair-à-pair, à contrôle réparti, ou
- Selon un mode centralisé, à contrôle concentré.

Pour faire confiance à un tiers inconnu :

décentralisé s'assurer de l'identité de celui-ci. Peut être transitif. Modèle de GPG. Fiable, mais passage à l'échelle difficile.

centralisé une autorité centrale garanti l'identité du tiers. Modèle de TLS. Passage à l'échelle trivial, mais confiance absolue dans l'autorité nécessaire.

B&A CONSULTANTS SECURITE INFORMATIQUE

Problème du modèle centralisé

- Autorité ne gagne pas d'argent à chaque vérification d'identité d'un inconnu.
- Gagne de l'argent à chaque enregistrement d'un nouvel inconnu.
- Business model : enregistrer un maximum d'inconnus
  1. le plus rapidement possible,
  2. en minimisant les vérifications.

$\Rightarrow$ attaques sur TLS, toutes liées à la « complicité » d'une autorité de certification et aux aléas de la vraie vie :

- fusions-acquisitions,
- sous-traitance,
- mutiplication d'autorisés racine,
- appât du gain,
- interceptions (plus ou moins) légales.

B&A CONSULTANTS SECURITE INFORMATIQUE

Fusions-acquisitions et disparitions

- Certificat d'autorité racine à très longue durée de vie (20 à 30 ans)
- L'AC peut mourir ou disparaître avant cette expiration
- Qui contrôle alors ses certificats ?
- Sachant qu'ils peuvent être déployés dans tous les navigateurs

Mozilla Dev, 02 avril 2010 10:19:35 -0700

Recommend Removing RSA Security 1024 V3 root certificate authority

B&A CONSULTANTS SECURITE INFORMATIQUE

Sous-traitance et/ou incompétence

- Contrainte économique triviale : la présence mondiale implique une présence locale, facile à mettre en place avec des partenaires locaux.
- Le partenaire sous-traitant d'une autorité de certification peut être « moins pertinent » quant à la qualité de ses contrôles (respect de la Déclaration des Pratiques de Certification).
- Sans parler d'un sous-traitant de niveau deux ou trois. . .

Comodo (2008)

Obtention par des tiers de faux certificats Mozilla

`blog.startcom.org/?p=145`

Prolifération d'autorités racines officielles

- Existence de très nombreuses autorités nationales indépendantes les unes des autres (pas de racine unique)
- Problème parfois politique
  Mozilla Debates Whether to Trust Chinese CA
  `www.freedom-to-tinker.com/blog/felten/mozilla-debates-whether-trust-chinese-ca`
- Autorité parfois très nationale (DPC de l'autorité hongroise non traduite `srv.e-szigno.hu/menu/index.php?lap=dokszab`)

Confiance, vous avez dit confiance ?

Chaque autorité racine a le même statut ultime. Leur faites-vous absolument confiance à toutes ?
Diginotar (juillet 2011) `isc.sans.edu/diary.html?storyid=11500`

Money is money

Une première dose gratuite pour accrocher les clients. . .

Délivrance de certificats gratuits à courte durée de vie.

Abus de confiance !

- Possible d'obtenir un certificat de signature, durée de vie 60 jours
- Sans présenter aucun justificatif.
- Permet de compromettre l'iPhone (applications signées)
- Permet de signer un exécutable ou une macro Office pour une attaque unique.

B&A CONSULTANTS SECURITE INFORMATIQUE

Le meilleur pour la fin

Autorités Coopératives – pour les interceptions

- Équipements d'interception légale, déchiffrement flux TLS
- grâce coopération autorités racines (nationales) qui fournissent certificats ad-hoc

`www.wired.com/threatlevel/2010/03/packet-forensics`

Été 2011

Diginotar, cas d'école, souligne risques modèle actuel de certification.

Hiver 2013

Bercy, AC interne (sous-autorité IGC/A), boîtiers Fortinet...

Plan

Les IGC en pratique ?

- Nombreuses difficultés qu'il faut connaître
  - Pour les éviter (si le projet IGC ne peut être déphasé), ou
  - Pour éviter de lancer un projet cher, très consommateur de ressources et qui n'atteindra pas ses objectifs.
- Souvent, sur situations « non nominales » mais très courantes : (gros) problème d'échelle.

Il faut le reconnaître

C'est désagréable à entendre (surtout si vous avez dépensé beaucoup d'argent pour mettre en place votre IGC).

B&A CONSULTANTS SECURITE INFORMATIQUE

Défaillances courantes d'une IGC

- Manque ou insuffisance de contrôles par l'AE.
  - Production de « faux » certificats.
  - Identités invalides.
  - Fourniture de certificats valides à d'autres EE que leur porteur légitime.
- Durée de vie trop élevée, découplée de la durée d'utilisation du certificat.
- Oubli de l'élément humain,
- Piratage/compromission de l'IGC
- Gestion des incidents calamiteuse.

Quelques exemples

Attaquer une IGC revient à attaquer toute l'infrastructure gérée.

Fin 2008, RapidSSL

- collision MD5 + numéros de certificats prédictibles + dates de validité prévisibles.
- 300 PS3 en parallèle.
- Résultat : obtention d'un FAUX certificat d'AC reconnu par tous les navigateurs.

B&A CONSULTANTS SECURITE INFORMATIQUE

Comodo, mars 2011

Mar 24, 2011 | 03:39 PM | 1 Comments

By Kelly Jackson Higgins
Dark Reading

Comodo's revelation yesterday that nine SSL certificates had been issued for fraudulent websites posing as domains for high-profile sites serves as a wake-up call for a certificate process that security researchers long have warned is riddled with holes.

The certificate authority (CA) reported that the certificates were issued for mail.google.com, www.google.com, login.skype.com, addons.mozilla.org, login.live.com, and global trustee, and three different ones for login.yahoo.com. Only one of the login.yahoo.com certificates was spotted as up and running on the Internet.

- Piratage de l'AC Comodo
- Obtention de vrais faux certificats
- Attaques par intermédiations possibles

Encore eux ?

Comodo a déjà été épinglé en 2008 pour des « procédures de vérifications allégées ».

B&A CONSULTANTS SECURITE INFORMATIQUE

Diginotar, juillet 2011

- Autorité de certification hollandaise, société privée.
- Piratée (au moins) en juillet 2011, probablement bien avant (2009, intrus turcs ?)
- Production de faux certificats ($>$ 530 certificats au total).
- Attaques par intermédiation (GMail, projet Tor), signature logiciels (WindowsUpdate), etc.
- Détection par des utilisateurs : avertissements inhabituels navigateurs.

19 septembre 2011

Faillite de Diginotar.

`http://www.networking4all.com/en/ssl+certificates/ssl+news/time-line+for+the+diginotar+hack`
`http://www.f-secure.com/weblog/archives/00002228.html`

Les faux certificats

CN et estimation nombre certificats frauduleux émis (04/09/2011)

..com	1	*.*.org	1	*.10million.org	2
*.JanamFadayeRahbar.com	1	*.RamzShekaneBozorg.com	1	*.SahebeDonyayeDigital.com	1
*.android.com	1	*.aol.com	1	*.azadegi.com	1
*.balatarin.com	3	*.comodo.com	3	*.digicert.com	2
*.globalsign.com	7	*.google.com	26	*.logmein.com	1
*.microsoft.com	3	*.mossad.gov.il	2	*.mozilla.org	1
*.skype.com	22	*.startssl.com	1	*.thawte.com	6
*.torproject.org	14	*.walla.co.il	2	*.windowsupdate.com	3
*.wordpress.com	14	Comodo Root CA	20	CyberTrust Root CA	20
DigiCert Root CA	21	Equifax Root CA	40	GlobalSign Root CA	20
Thawte Root CA	45	VeriSign Root CA	21	addons.mozilla.org	17
azadegi.com	16	friends.walla.co.il	8	login.live.com	17
login.yahoo.com	19	my.screenname.aol.com	1	secure.logmein.com	17
twitter.com	19				

B&A CONSULTANTS SECURITE INFORMATIQUE

AC qui ne vérifient rien

Bien plus courant qu'intellectuellement confortable.

> This prompted me to create another certificate through them, but this time by using a domain name which should never be issued to me. For the purpose of testing, I selected the domain mozilla.com (I'm certain they will forgive me). Five minutes later I was in the possession of a legitimate certificate issued to **mozilla.com - no questions asked - no verification checks done - no control validation - no subscriber agreement presented, nothing**.
>
> With the understanding about MITM attacks, the severity of this

Figure – 23/12/2008, blog.startcom.org/ ?p=145

La norme X500

Issue de l'historique des télécommunications des années 1970

- Grandes sociétés monopolistiques
- Réseaux très onéreux et à faibles débits (par rapport à aujourd'hui)
- Très peu d'utilisateurs (par rapport à aujourd'hui)

Les certificats (norme X509)

- X509 dérive de X500, qui se voulait un « répertoire global de toutes les informations ».
- A hérité de toutes les lourdeurs liés à une optique « administration bureaucratique centralisée ».
- Et ce n'est pas rien.

Une clé publique

```
Certificate:
Data:
Version: 3 (0x2)
Serial Number: 95:36:aa:fc:60:bf:d8:a8
Signature Algorithm: sha256WithRSAEncryption
Issuer: C=FR, ST=Midi-Pyrenees, L=Toulouse,
    O=B&A Consultants, OU=Autorité de certification,
    CN=Certification B&A/emailAddress=ca-auth@ba-consultants.fr
Validity
    Not Before: Oct 23 09:03:00 2016 GMT
    Not After : Oct 23 09:03:00 2017 GMT
Subject: C=FR, ST=Occitanie, L=Toulouse, O=B&A Consultants,
    OU=Accès distants, CN=daya
```

Plus des informations cryptographiques.

Identification d'une clé publique

Le sujet (Distinguished Name, ou DN) de la clé publique est composé de plusieurs champs :

Country	C=FR
STate	ST=Occitanie
Organization	O=B&A Consultants
Organisational Unit	OU=Accès distants
Common Name	CN=pyb@ba-consultants.fr

Le sujet constitue la partie « sémantiquement intéressante » de la clé publique. C'est l'identité du porteur.

Les dénominations

- L'organisation du DN n'est pas naturelle (C, ST, L, O, OU, CN...).
- Ne correspond pas à des situations réelles courantes.
- Faut-il changer de certificat à chaque fois qu'on change de service (OU), de société (O), ou que l'on déménage (ST) ?
- Faut-il connaître tout le pedigree d'une EE pour récupérer sa clé publique ?

Que signifient les champs ?

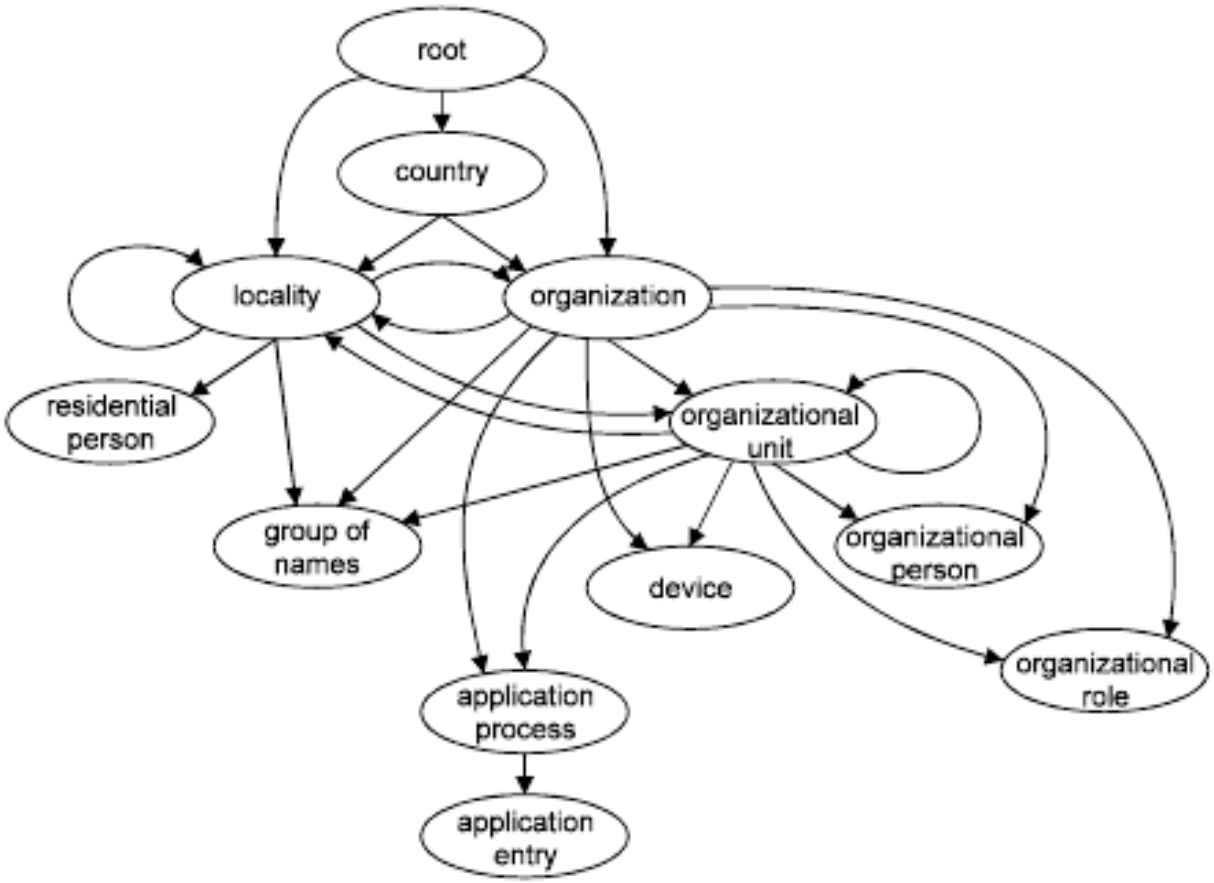


Figure – Diagramme réel pris dans la norme X521

Signification des champs ?

- Pour une société, C, ST, L sont...
  - L'emplacement de la société ?
  - Du siège social ?
  - De l'agence ou du bureau ?
- Pour un particulier, ce sont...
  - Le lieu de naissance ?
  - De résidence ?
  - De travail ?
  - Et les personnes très mobiles ?

Collisions de noms

- Possibles et triviales.
- Nomenclature CN=Marie Dupont 1, CN=Marie Dupont 2 ?
- Qui aura le numéro 1 ?
  - Par ordre d'embauche ?
  - Et si le numéro 4 est hiérarchiquement supérieur aux numéros 1 à 3 ?
- Problème humain difficile à régler
- Surtout que le résultat doit être utilisable !

Notez bien

On retrouve les mêmes types de problèmes avec les adresses électroniques.

Exemples d'ailleurs

SPKI Identifiant significatif pour le domaine applicatif local

- Compte sur un serveur
- Numéro de carte bancaire
- Identifiant applicatif

GPG le nom (équivalent du DN) est composé

- d'un libellé (structure libre)
- de l'adresse électronique (structure presque libre)
- d'éventuels synonymes (structure libre).

Les révocations

- Les listes noires de certificats (CRL) ne **peuvent pas** fonctionner efficacement.
- Et induisent des risques subtils dans la gestion de la confiance.
- Dans les faits, la vérification des CRLs est, au mieux, légère et mal faite.
- Bien souvent, elle est ignorée totalement (absence de CRL, ou CRL jamais mise à jour).

Les CRL ne marchent pas

- Ne sont pas produites suffisamment souvent pour être efficaces.
- Distribution très coûteuse en termes techniques (volume, réplication sur tous les systèmes, etc.)
- Si blocage distribution d'une CRL, certificats ne sont pas révoqués.
- Révocation du certificat auto-signé d'une AC (CRL signée par l'AC avec le certificat qu'elle révoque) ?

La vie est très injuste

Dans l'univers limité, centralisé et monopolistique des telcos des années 1970, cela fonctionnait très bien.

Problèmes plus subtils

- Dates des révocations peuvent être rétroactives (le certificat ABC a été révoqué le 17 janvier 2016).
- Conséquence : notion de certificat « valide aujourd'hui » n'est donc plus significative.
- CRL peut être publiée rétroactivement (31/12/2016, publication CRL datée du 01/01/2016 avec les révocations de 2015).
- Conséquence : destruction du concept de non-répudiation.

Téléchargement des CRL

- CRL valide depuis . . . jusqu'à . . .
- A l'expiration :
  - Tous les consomateurs de la CRL se connectent au serveur du CA pour récupérer la nouvelle CRL
  - Augmentation massive de la charge (système, réseau)
  - Déni de service (non distribution de la CRL)

Un petit calcul

Une CRL de 1 Mo avec 10 millions de clients, mise à jour toutes les minutes $\Rightarrow$ débit de 162.7 Go/s.

Compromis immédiat

- Compromis : mémorisation des CRL au-delà de leur durée de vie.
- Donc validation possible de transactions faites avec certificats révoqués.
- Révocation devrait être « immédiate », en flux tendu, pour beaucoup d'applications :
  - Contrôle d'accès
  - Signature électronique

Un parallèle

Imaginez que l'opposition sur votre carte bancaire mette 3 mois à être reconnue sur les DAB et les TPE...

Les CRLs pour de vrai

- Demande « instantanée », au moment de la vérification d'une clé publique.
- Connexion vers un serveur qui renvoie la réponse (OK/NOK).
- Exemple typique : vérification des cartes bancaires.
- Contrainte : robustesse serveur(s) et infrastructure technique.

On ne rêve même pas

OCSP : Online Certificate Status Protocol.

OCSP

- Optionnel, mais conseillé à toutes les AC
- Elimine nombreux problèmes des listes de révocations
- En ajoute quelques nouveaux
- Dans certificat, ajout extension `Authority Information Access` avec URL répondeur OCSP AC
- Clients devraient toujours faire vérification OCSP si information présente

Limites OCSP

- Confidentialité : AC est informée de l'accès d'un navigateur à un de ses clients (reçoit requête OCSP)
- Dépendance tiers (AC) ralentit navigation
- Forts volumes : serveurs OCSP pour Google, Facebook et autres GAFA ? A chaque ouverture de session TLS ?
- Défaut client souvent « fail open » : pas de réponse du serveur $\rightarrow$ certificat OK
- Compromission clé privée serveur : attaquant en interposition, peut interférer avec requêtes OCSP

Évolution

Estampillage OCSP fait par le serveur via son AC

Estampillage OCSP

- Coût requête OCSP incombe au serveur (fournisseur) lui-même et pas au navigateur (client)
- A intervalles réguliers, serveur demande statut OCSP à son AC
- Statut OCSP estampillé (date d'expiration), **signé par AC**
- Client utilise l'extension `Certificate Status` dans établissement session et ajoute cette demande
- Statut OCSP joint au certificat (réponse selon extension `Certificate Status`)
- Pas de réponse, client contacte serveur OCSP
- Échec vérification statut (mauvaise signature, périmé) : rupture connexion

B&A CONSULTANTS SECURITE INFORMATIQUE

Conclusions sur la révocation

Une AC doit mettre en place toutes les fonctions nécessaires à la gestion fine et rapide de la révocation de ses certificats

- Procédures de signalement
- Procédures de validation, révocation, enregistrement, diffusion
- Répondeur OCSP
- Estampille OCSP

Sinon

Mauvaise gestion des révocations $\rightarrow$ maintient situation de compromission potentielle.