Sécurité informatique

Yves Denneulin

Un système sûr ?

« C'est un des problèmes les plus durs à résoudre parce qu'on doit prouver qu'un système marche contre un ennemi omnipotent. Or l'omnipotence est, par essence, indéfinissable. »
« La sécurité informatique consiste à montrer qu'un truc marche contre un ennemi dont on ne connaît pas les armes. C'est donc un problème scientifiquement impossible, parce qu'on ne peut pas le poser correctement. »

Gérard Berry, professeur au collège de France, médaille d'or du CNRS, 2016.

C'est quoi alors la sécurité ?

« la sécurité informatique est vouée au compromis. Son postulat c'est : toute attaque est imaginable, mais il faut la rendre trop chère. Un système est sûr non pas quand il est inattaquable – ce qui est théoriquement impossible –, mais quand ça coûte trop cher de l'attaquer.

La sécurité informatique consiste d'abord à s'assurer que les algorithmes ne sont pas faux. Un algorithme faux est une faille. »

Les propriétés attendus d'un système sûr

- Confidentialité -> chiffrement
- Légitimité -> Authentification
- Intégrité -> Signature
- Disponibilité
- Non répudiation -> Notarisation

Confiance et sécurité

- Liste des éléments dans lesquels on place sa confiance pour chaque action
  - matériel, algorithmique, logiciel, réseau
  - confiance dans le concepteur/fournisseur de services
  - garantie de non altération
- La confiance est au coeur de la sécurité
  - elle est gagnée par audit et conservée par vérification
  - … et par des mises à jour régulières!

Les dimensions de la sécurité

- algorithmique : qualité du résultat garanti indépendamment des entrées
- matérielles : conception et vérification de composants fiables
- logiciel : comportement conforme à ce qui a été spécifié
- organisationnelle : rôles et responsabilités de chaque acteur identifiés

Domaines de recherche

- Cryptographie
- Sémantique, preuve de programmes
- Tests automatiques et analyse de logiciels
- Mécanismes dans les systèmes d'exploitation
- Réseaux et protocoles associés
- Vie privée et données personnelles
- …

Confidentialité

- Propriété de non divulgation et non vol
  - Des données stockées et manipulées
  - Des communications

Authentification

- Propriété de contrôle de tout intervenant dans le système
  - personne physique,
  - services,
  - composants matériels.
- Souvent utilisé pour définir des **droits** à partir de **rôle** dans le système

Intégrité

- Propriété de non-altération des données depuis
  - leur écriture (si opération de lecture)
  - leur envoi (si opération de réception)
- Généralement couplée avec l'authentification

Disponibilité

- Propriété de bon fonctionnement du système
  - même en conditions dégradées
- 2 aspects
  - capable de s'exécuter
  - capable de recevoir des requêtes et d'émettre des réponses

Non répudiation

- impossibilité de nier être à l'origine d'une action et/ou d'une communication
- liée à authentification et intégrité

Doctrine

- *Security through obscurity*
  - Sécurité par le secret
- Peu efficace
  - Les désassembleurs
  - Les outils de cartographie
- Contraire au principe de connaissance des failles
- Comme toujours, compromis à trouver

Principle of least privilege

- Chaque entité doit avoir les privilèges juste nécessaire à son rôle
  - Personnes, processus, ...
- Définition des rôles
  - Une personne en a plusieurs
- Permissions accordées suivant les rôles
- Difficile à assurer en pratique

En quels éléments place-t-on sa confiance ?

- On ne peut pas tout contrôler/auditer
  - matériels, logiciels
- Choix à faire, après audit et étude
- Est-on sûr que rien n'a été modifié ?
  - Algorithmes + protocoles permettent de le vérifier
    - mais sont ils dignes de confiance ?

Plan de la séance

- Cryptographie
- Architecture matérielle de confiance
- Données personnelles

Cryptographie

Propriétés à assurer

- Secret
  - Stockage et communication
- Authentification
  - Des personnes, des actions et des ressources
- Intégrité
  - Stockage, communication, dispositifs

Algorithmes cryptographiques

- Science très ancienne
  - Correspond à un besoin ancestral
- Différentes époques
  - Différentes menaces
  - Essentiellement pour les communications
  - Maintenant pour le stockage aussi
    - Problématique différente

Première époque : antiquité-première guerre mondiale

- Chiffrer => détenir ou partager un secret
- Première époque : secret=algorithme
- Exemple : code de césar
  - « Simple » permutation de lettres
    - A->I b->J etc.
  - Trivial à casser
    - Une simple analyse statistique suffit
    - La « force brute » : énumération de toutes les possibilités est faisable facilement

Exemple d'attaques cryptographiques

- Cryptanalyse : analyse de la robustesse des algorithmes cryptographiques
- Attaque par séquences connues
  - Praticable car existence de protocoles
- Attaque par séquences forcées
  - Moins praticable mais peut être utilisée
- Analyse différentielle
  - Différences chiffrées ≠ différences du clair
- Un bon algorithme doit résister à tout cela

Seconde époque

- Algorithme connu
  - Et longuement étudié!
- Secret=clé
- Algorithme à clé secrète
  - **Symétrique** : la même clé sert à chiffrer et déchiffrer
  - E(M,K)=M' => chiffrement du message
  - D(M',K)=M => déchiffrement du message

Limitations des algorithmes symétriques

- Choix de la clé
  - Aléatoire de bonne qualité
  - Pas toujours facile (source?)
- Transmission de la clé
  - Connaissance a priori de la clé nécessaire
  - Commerce électronique ?
- Nombre de clés pour communications multi-points

Algorithmes asymétriques

- Dit aussi « chiffrement à clé publique »
- Une clé = 2 parties différentes, k=(k1,k2)
  - Une pour chiffrer
  - Une pour déchiffrer ce qui a été chiffré
  - Si E(K1,M)=M' alors D(k2,M')=M
  - Si E(k2,M)=M'' alors D(k1,M'')=M
  - E : fonction de chiffrement D de déchiffrement
- Typiquement k1=clé privée et K2=clé publique

Avantages du chiffrement asymétrique

- Pas de connaissance a priori des clés nécessaires
  - Seule la partie publique est nécessaire
  - Peut être récupéré sur un site de confiance (on verra comment plus tard)
- Pas un nombre exponentiel de clés

Exemples d'algorithmes asymétriques

- Le plus connu : RSA (77)
- Opérations identiques pour le chiffrement et le déchiffrement
  - E=D=exponentiation modulaire
- Regardons la création d'une clé

1. Choisir deux grands nombres premier, p et q, au hasard

Calculer n=pq

2. Choisir un entier e premier avec (p-1)(q-1)
3. Calculer d tel que de=1 mod (p-1)(q-1)

- K1=(d,n), K2=(e,n)

Algorithmes asymétriques : RSA (2)

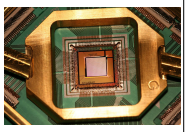
- E(k1,M)=M^d mod n=M'
- D(k2,M')=M^ed mod n=M

- Caractéristiques de RSA
  - Opérations de chiffrement et déchiffrement coûteuses
  - Résistance de RSA dépend de la factorisation d'un nombre en deux premiers
    - Importance de choisir des grands nombres!

Facteurs affaiblissants

720000 millards de pC de 2007, 15 millions d'années
Dans 162 ans, une seconde

- Taille de la clé
  - Rendre la clé introuvable par recherche exhaustive, 128 bits est inatteignable en symétrique
- Clé pas assez aléatoire
- Clé accessible
  - Insuffisamment protégée
  - Problème d'organisation

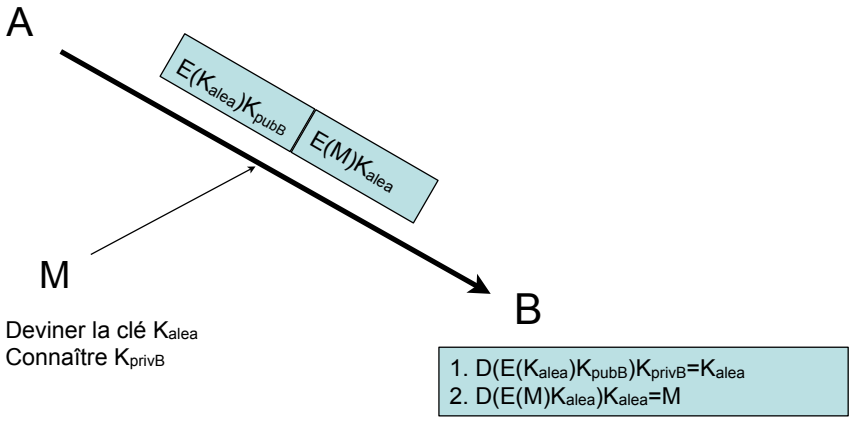


Utilisation de la cryptographie

Assurer le secret

- Stockage : algorithmes symétriques
  - Générer aléatoirement une clé secrète
  - Chiffrer la totalité des données avec
  - Chiffrer la clé avec un mot de passe
  - Robustesse système = robustesse du mot de passe
    - Problème : son stockage!
- Secret de la transmission de A vers B
  1. A choisit une clé symétrique aléatoire
  2. A chiffre le message avec
  3. A envoie le message chiffré et la clé secrète chiffrée avec la clé publique de B

Secret de la transmission



Authentification

- Être sur de l'identité de celui qui
  - Envoie le message
  - A écrit (modifié) la donnée
- Trouver ce qui permet d'authentifier une entité
  - Sa clé privée!
- Solution naturelle
  - Chiffrer tout avec sa clé privée
  - Éventuellement avec la méthode vue précédemment

Authentification (2)

- Méthode lourde
  - Grande quantité de chiffrement
- Est-ce vraiment nécessaire ?
  - Il suffit de ne chiffrer qu'une **empreinte** des informations
- Comment calculer cette empreinte ?
  - Utilisation des fonctions de hachage

Fonctions de hachage

- H(M)=h où M est une suite de bits
- Propriétés
  - Facile à calculer (en matériel)
  - Non inversible (évidemment!)
  - « Impossibilité » à partir d'un hash de trouver une suite de bits ayant le même hash => collision

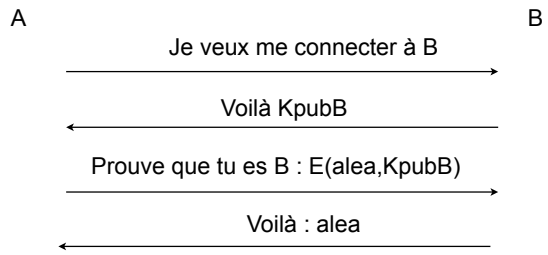
Utilisation des fonctions de hachage

- Authentification : A a écrit/envoyé les données M
  - Message M
  - H(M)=h
  - h est chiffré avec la clé privée de A
- B veut vérifier que c'est bien A qui a signé M
  - Il calcule H(M)=h1
  - Il calcule D(E(H(M),KprivA),KpubA)=h2
  - Si h1=h2 alors il est sur que l'empreinte a bien été généré par A et correspond donc bien au message

Utilisation des fonctions de hachage (2)

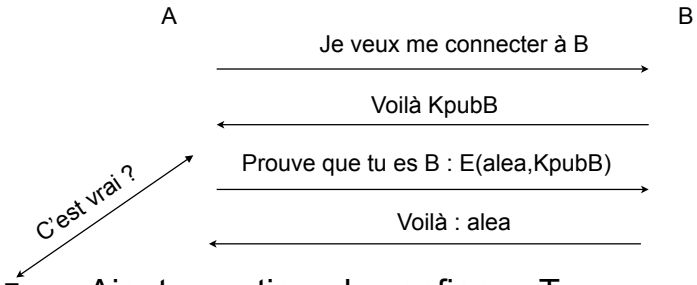
- Si modifications des données
  - H1 ne correspond pas à l'empreinte chiffrée
  - Modifications découvertes
- Intégrité
  - Utilisation identique
    - Les deux propriétés vont de pair

Utilisation du chiffrement dans une communication



- Seule information : celui qui a répondu à la première requête connaît KpubB et KprivB
  - Confiance placée dans le « réseau »
- Comment savoir si KpubB est bien la clé de B?

Utilisation d'un chiffrement dans une communication (2)



- Ajouter un tiers de confiance T
- Ne change rien au problème ?
  - Si! Avec la clé du tiers on peut authentifier beaucoup de sites

Utilisation du chiffrement dans une communication (3)

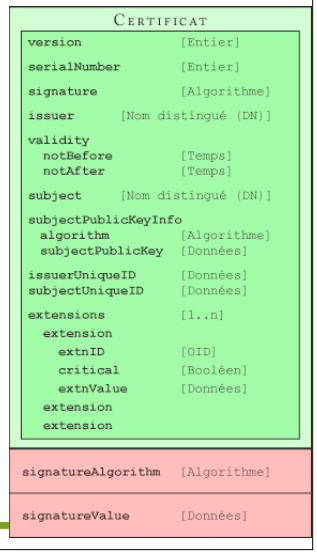
- Reste le problème de la communication préalable systématique
  - Solution : chiffrer la clé de B avec la clé privée du tiers de confiance
  - Utilisation de clés de session pour le secret
- Récapitulons
  - Identité de B
  - Son adresse
  - Sa clé publique

Signé par K_{privT}

C'est un certificat!

Certificats

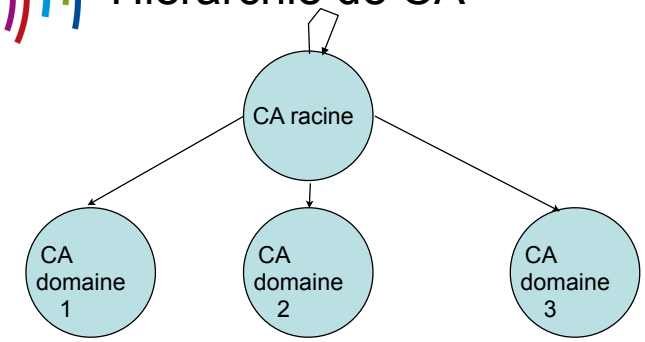
- Contient l'identité de la ressource
  - Serveurs
  - utilisateurs,
  - services, etc.
- Une norme : X509
- Placer sa confiance dans le certificat = placer sa confiance dans le tiers qui l'a émis
  - Autorité de certification (CA)



Autorité de certification

- La clé privée de l'autorité de certification est une ressource critique
  - Elle ne doit pas être diffusée
- Comment gérer une organisation hiérarchique ?
  - Distribution de la clé => mauvais
  - Centralisation de la signature => mauvais en performances
- Un CA devrait pouvoir **déléguer** sa signature

Hiérarchie de CA

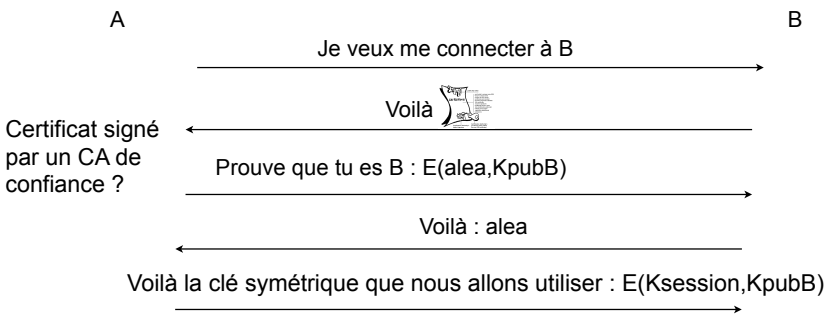


- CA racine signe les certificats des CA des différents domaines
- Chaîne de confiance vérifiable
- Et le CA racine ? Auto-signé!
- Chaque CA maintient ses listes de révocation (CRL)

Autre organisation des chaînes de certification

- Certificats multi-signés
  - Peuvent être signés par d'autres certificats utilisateurs
- Confiance dans un certificat si
  - S'il existe une chaîne de confiance menant à un certificat en qui on a confiance
    - Confiance dans un de ceux qui l'a signé
    - Récursif : confiance se propage
  - « Key signing party »

Fonctionnement complet : ouverture d'une session



- Clé de session (chiffrée avec KpubB) symétrique
- La clé de session doit être robuste!
- Elle expire au bout d'un certain volume de données

Confiance et infrastructure

Sécurité = confiance

- Comment être sûr qu'aucun élément n'a été modifié ?
  - attaque de la bonne malveillante

Éléments

- Matériel
  - Dispositifs d'espionnage
- Logiciel
  - BIOS
  - Hyperviseur (virtualisation)
  - Systèmes d'exploitation
  - Bibliothèques système
  - Application

Assurer la confiance

- Être sûr des éléments
- Vérification des éléments de la chaîne
  - Facile en théorie
    - hash+signature pour logiciels
    - vérification de clés connus pour dispositifs matériels
  - Difficile en pratique
    - On ne peut vérifier que ce qui est au-dessus de nous dans la pile
  - Il faut une vérification globale
    - Par une entité indépendante et inaltérable
      - Forcément matériel

Trusted computing

- Association
  - Garantit qu'aucun élément n'a été modifié
  - Y compris par son propriétaire!
- Vérification des éléments
- Évolution du Trusted Platform Module
  - Appelé aussi Palladium dans Windows
- Composants matériels et logiciels

Confiance dans les composants matériels

- Signature évidemment!
  - Par une clé stockée dans les dispositifs
- Vol de la clé ?
  - Exemple des clés CSS
- TPM : endorsement key
  - Unique à chaque famille de dispositif
  - Fixé en usine

Autres fonctions du trusted computing

- Protection de la mémoire
- Stockage fermé
  - Format fermé et accès restreint (DRM)
- Certification du code exécuté
  - console de jeu
  - téléphone
  - … de plus en plus de dispositifs

Utilisation des fonctions du TCG

- Chiffrement du disque dans Windows
  - Empêcher l'attaque de la bonne malveillante
- Vérification du matériel dans MacOS
  - Pas de « hack-intosh »
- Protection des données
  - Certification de la pile qui y accède
- Jeux en ligne
  - Argent en jeu
  - Pas de confiance dans le client (tricheur)

Données personnelles ?

Une donnée personnelle est une information qui permet de vous identifier ou de vous reconnaître, directement ou indirectement. Il peut s'agir d'un nom, prénom, date de naissance, adresse postale, adresse électronique, adresse IP d'un ordinateur, numéro de téléphone, numéro de carte de paiement, plaque d'immatriculation d'un véhicule, empreinte digitale, ADN, photo, numéro de sécurité sociale... .

Pourquoi les protéger ?

- Éviter de donner trop de connaissances sur les utilisateurs
  - principe du moindre privilège
- Permettre l'oubli
- Empêcher les recoupements entre base

La maîtrise des données personnelles est un des enjeux majeurs de la société de l'information

Gestion des données personnelles

- Si collecte déclaration obligatoire
- Protection : mesures de sécurité
  - physiques
  - logiques
  - « adaptées à la nature des données et aux risques présentés par le traitement. »

Techniques d'identification

- Individualisation : trouver un individu dans une base
- Corrélation : relier entre elles des données appartenant au même individu
- Inférence : déduire des données à partir d'autres

Techniques d'anonymisation

- Randomisation : ajout de bruit en modifiant les valeurs d'origine en conservant la distribution originale des valeurs
  - il faut le bruit soit cohérent
- Randomisation, Permutation : mélanger les valeurs des attributs pour conserver la même distribution mais pas les mêmes enregistrements
  - ne pas permuter deux attributs corrélés
- Agrégation et k-anonymité : regrouper les individus en groupe de k
  - attention à ne pas créer des groupes tous équivalents (l-diversité et t-proximité)
- Pseudonymisation: remplacer/supprimer un attribut (attention aux autres!)

Pseudonymisation

- table de correspondance
  - faiblesse évidente
- algorithme de chiffrement
  - réversible facilement, secret=clé
- hachage
  - non réversible
    - si la fonction est bonne et la force brute impraticable
  - il ne faut pas que la donnée hachée soit devinable

quid de l'open data ?

- Rapport parlementaire de avril 2014
- Publication/communication possible si
  - recueil du consentement de la personne concernée
  - anonymisation par l'autorité détentrice
  - autorisation par une disposition législative ou réglementaire spécifique

Faiblesses de l'anonymisation

- Nature des données
  - 2006 : base de données de navigation AOL
  - informations médicales
    - 89% de fiabilité avec hôpital, mois/année de naissance, sexe, mois de sortie, durée du séjour
- croisement d'informations
  - avec d'autres données publiées
  - Netflix en 2008, croisement rotten tomatoes, amazon
  - sera de plus en plus efficace avec l'augmentation des données publiées

C'est quoi la *privacy* ?

- L'identité de l'utilisateur
  - ce qu'il est/possède
  - défi : empêcher un attaquant d'associer des données à un utilisateur
- Les données de l'utilisateur
  - ce qu'il produit
  - but de l'attaquant : extraire ces données ou les déduire ou les exploiter
- Les catégories ne sont pas étanches

Protéger l'identité

- notion de k-anonymité
  - il faut extraire k éléments pour reconstruire l'identité
- ils peuvent être extraits de différentes bases!
  - exemple de Netflix
  - ou d'éléments croisés dans la base (si pas k-anonymité)

Protéger les données

- elles doivent être collectées pour rendre un service
  - Compromis utilisabilité/confidentialité
- ajout de bruit :
  - propriété de *differential privacy*
  - données stockées altérées avec une probabilité proportionnelle au nombre d'informations qui peuvent être corrélées

Differencial privacy, compromis privacy/precision

- compromis entre privacy et précision
  - plus de bruit=moins de précision
- plus il y a de requêtes faites, plus la confidentialité des informations stockées diminue
  - notion de *privacy budget*
  - données typiquement utilisées pour éduquer un apprentissage automatique
    - impact du *privacy budget* https://www.usenix.org/system/files/conference/usenixsecurity14/sec14-paper-fredrikson-privacy.pdf

Conclusion

- Aspect essentiel
  - éducation indispensable
- enjeu de la confiance des utilisateurs
  - la gagner est compliquée, la perdre est très rapide
  - ne pas extrapoler sur la situation actuelle
- Enjeu légal
  - s'en affranchir peut coûter cher
  - une étude récente : http://techscience.org/a/2015103001/