Cours Master Info 2

2ème Année

Cryptographie (partie informatique)

Plis fòs ba pengwen là !

Vincent Pagé vpage@univ-ag.fr Université Antilles Guyane

Présentation

Bonjour à tous.

- Vincent Pagé, Mcf en Informatique.
- Email : vpage@univ-ag.fr
- Site Web : http://calamar.univ-ag.fr
  /uag/ufrsen/coursenligne/vpage/new_site/

Bibliographie

Le contenu de ce cours est essentiellement issu de :

- Applied Cryptography (Bruce Shneier)

Un petit peu aussi de :

- Handbook of Applied Cryptography (A. Menezes, P. vanOorschot, and S. Vanstone)
- RSA Security 's official guide to cryptography
- Internet, en fouillant partout...

Contexte (1)

Transaction : échange d'une richesse (bien physique, information...).
Support : le moyen d'existence de cette transaction.
Canal : la voie physique de transport de cette transaction.

Avant l'informatique :

- Support d'information contrôlé (papier)
- Canal contrôlé (présence, postal...)
- Transactions contrôlée (présence)

=> Transactions sûres (peu d'escrocs)

=> Habitudes laxistes (sauf espionnage).

Contexte (2)

Informatique, pré-Internet :

- Information copiable, modifiable.
- Transactions contrôlée (présence lors du login)
- Peu d'utilisateurs, peu d'argent, peu d'escrocs !

Transactions semi sures, protocoles d'échanges laxistes (ftp)

Contexte (3)

Actuel, utilisation massive d'internet pour :

- La transmission de messages.
- Le travail a distance.
- Le commerce.

Des messages transitent sur un canal (internet). Ce canal n'est pas sur (il peut être écouté ET modifié).

Comment conserver une certaine sécurité pendant ces transactions ?

Besoins informatiques (1)

Monde réel A quoi sert un password ? Que nécessite une cb ?

On distingue les 5 besoins suivants pour toutes les transactions :

1. Identification et authentification :
   - Pour une personne *(ex : carte bleue, permis)*, on distingue :
     - Identification

       Connaître l'identité d'une entité (personne, pc..) (souvent : déclaration ou présentation d'un objet)
     - Authentification (authentication) :

       Vérifier cette identité (souvent : vérifier un lien entre le déclarant et l'objet présenté ou entre l'objet présenté et des propriétés requises)..

       Objectif => Accorder des droits a une entité authentifiée (personne, pc)
   - Valable aussi pour l'authentification de messages..

Besoins informatiques (2)

2. Confidentialité
   - S'assurer qu'un tiers ne peut pas accéder au contenu d'un message (s'apparente à du camouflage...voir plus loin)
3. Intégrité de contenu
   - Le message envoyé ne peut être modifié volontairement en un autre message choisi.
4. Non répudiation
   - Je ne peux pas nier avoir fait quelque chose (achat, message). Eventuellement, je ne peux pas nier avoir recu le message (avec sa date)
5. Facilité d'utilisation

voilà la Checklist a vérifier pour chaque protocole ...En fonction des objectifs du protocole

(Monde réel : achat avec du liquide... mais ces distinctions ont sens en info surtout)

Attention !

- Une bonne méthode pour garantir la sécurité n'est pas basée sur le secret de la méthode
  - Parce qu'un secret, ca se diffuse ! (négligence, vantardise, avarice)

  Ex: Le départ (démission, licenciement) d'un détenteur du secret oblige à retrouver une autre méthode.
  - Rien ne vous garanti la sécurité d'une méthode que vous seul connaissez (un cryptanalyste est plus malin que vous en matière de cryptologie ...)

  => Une bonne méthode est une méthode connue de tous et que tout le monde a essayé de craquer (sans succès). Notez l'analogie avec les logiciels libres.

Cryptographie

- Définitions
  - Cryptographie : Science de transformer des messages en quelque chose d'illisible, sauf pour les détenteurs d'un secret. Par extension, science du secret
  - Cryptanalyse : Science du cassage de texte chiffrés.
  - Cryptologie : réunion des deux précédents.

  Cette science (cryptologie) va nous aider a régler les problèmes précédents à l'aide d'outils :
  - des algorithmes (chiffrage / déchiffrage / fonctions de hachage)
  - des protocoles d'échanges de données (pour une utilisation massive : pas un protocole par échange)

En français, on chiffre / déchiffre. In english, one crypts and decrypts.
En français, a partir d'un texte « en clair », on obtient un texte chiffré
In english, from a plain text, one obtains a cyphertext.

The Bad, the Beast and the Ugly

- Un utilisateur malveillant va pouvoir
  - s'attaquer à l'algorithme
  - s'attaquer au protocole.

  Exemple : authentification par mot de passe :
  - Craquer le mot de passe (décrypter le mot de passe chiffré (connu))
  - Feinter le protocole (attaque lors de la transmission du mot de passe, cheval de Troie, suppositions sur le mot de passe (mots du dictionnaire) ...)

  Le plus souvent, l'algorithme n'est pas le maillon faible.
  C'est le « protocole » au sens large (stockage des données, gestion des clefs...)

Outils de base de sécurité (mathématiques)

Trois outils : des algorithmes, non détaillés ici. on les supposera souvent parfaits !

1. Chiffrement symétrique d'un message :

On utilise une fonction à sens unique a trappe f(x,y) (inversible)

un message M se transforme en un message M'=fk(M).
La trappe est k (la clef) et elle permet de retrouver M à partir de M' . M = f-1(M',k)

La clef de chiffrement est la même que la clef de déchiffrement.
On peut le voir comme un coffre blindé (une serrure).

Outils de base de sécurité (mathématiques)

2. Chiffrement asymétrique d'un message :

- on chiffre avec une fonction f et une clef Kpub : clef publique
- on déchiffre avec fonction g et une clef Kpriv : clef privée

(ou en échangeant les clefs, cela dépend de l'objectif... voir plus loin...)

un message M se transforme en un message M'=f(M,Kpub).
La trappe est la clef secrète. M = g(M',Kpriv)

On peut le voir comme une boite aux lettre blindée. La clef publique permet de déposer un message, la clef privée de le récupérer.

Outils de base de sécurité (mathématiques)

3. Empreinte digitale d'un message.

En utilisant une fonction de haschage (h)

- On transforme un message M de taille variable en un message M'=h(M) de taille fixe (souvent courte).

Un document M (et un seul) correspond a M'. M' est l'empreinte digitale de M. Plus exactement, on ne peut pas forger volontairement un message M2 ayant la meme valeur de hachage que M.

Une bonne fonction de hachage garantit la non collision.

Outils de base de sécurité (Théorie de l'information)

Evoquons également dans les outils de cryptographie :

4. Stéganographie

Techniques de camouflage de données au seins d'autres données (dans une image....)

- Non abordé ici (culturel) .

Introduction aux protocoles

Disons qu'Alice veut envoyer un message a Bob...par mail...

Le protocole est le suivant :

1. *Alice envoie une clef secrète a Bob. A et B s'assure de la sécurité du stockage des clefs.*
2. *Alice chiffre son message avec la clef secrète.*
3. *Alice envoie le message chiffré*
4. *Bob déchiffre en utilisant la clef secrète.*

Que vaut ce protocole ?

Introduction aux protocoles

Pour bien comprendre les attaques possibles sur protocoles, on va disposer d'une panoplie de personnages :

- Alice, Bob, Carol, Dave : participants au protocole.
- Eve : Ecoute (attaque passive)
- Mallory : Malicieux (attaque active)
- Trent : Arbitre de confiance.

Il y en a d'autres mais on n'en parlera pas...

Les personnages ne sont reliés (sauf mention contraire) que par réseau (canal non sûr)

Introduction aux protocoles

Résumé d'un protocole :

Tableau 2d (Personnages vs garantie => garanties).

Résumé d'un protocole appliqué à un objectif :

Tableau 1d (Personnages : possibilites.)

Protocoles simples

1. Transmission chiffrée avec une clef secrète unique par couple de participants.

1. Alice et Bob connaissent la clef secrète par avance

2. Alice chiffre son message avec sa clef secrète.

3. Bob déchiffre avec la clef secrète.

=> Confidentialité / Integrite des donnees / authentification / Non repudiation / facilité ??

Confidentialité : Protocole SUR. MAIS : Problème de gestion des clefs car il faut :

- distribution sûre.
- stockage sûr et surtout stockage de beaucoup de clefs.

Protocoles simples

2. Transmission chiffrée avec un couple clef publique / Clef privée.

1. Bob diffuse publiquement sa clef publique.

2. Alice chiffre son message avec la clef publique de Bob

3. Bob déchiffre avec sa clef privée.

=> Confidentialité / Integrite des donnees / authentification / Non repudiation / facilité ??

Confidentialité : *Attaque Man in the middle (Si la clef publique est bien celle de bob, le protocole est sur !)*

=> il faudrait sécuriser les clefs publiques.

Protocoles simples

3. Utilisation d'une Hash function

1. Alice calcule la hash function de son message

2. Alice envoie le message (en clair) + le hash

3. Bob calcule le hash du message et le compare avec celui envoyé.

=> Confidentialité / Intégrité des données / authentification / Non répudiation / facilité ??

Protocoles simples appliqués

Authentification : *Monde réel : Connection base de donnée...*

1. Authentification d'une personne par mot de passe en clair :

1. Alice a un mot de passe, Bob connait ce mot de passe.

2. Alice envoie son mot de passe.

2. Bob vérifie si ce mot de passe correspond à celui qu'il connait.

2. Par mot de passe haché :

1. Alice a un mot de passe, Bob connait le hash de ce mot de passe.

2.Alice calcule le Hash de son mot de passe et l'envoie.

3. Bob vérifie le hash envoyé correspond a celui qu'il connait.

Protocoles simples

2. Authentification par mot de passe haché :

1. Alice a un mot de passe, Bob connait le hash de ce mot de passe.

2.Alice calcule le Hash de son mot de passe et l'envoie.

3. Bob vérifie le hash envoyé correspond a celui qu'il connait.

Problèmes :

- Eve et Mallory peuvent se faire passer pour Alice.
- Eventuellement adapté pour login : mais Bob doit proteger le fichier de mots de passe hachés d'une attaque par dictionnaire)

Protocoles simples

4. Utilisation d'une Hash function et d'un chiffrement asymetrique

1. Alice envoie sa clef publique Kpa a Bob et garde Ksa secrète

2. Alice calcule la hash function de son message

3. Alice envoie le message (en clair) + le hash chiffré avec Ksa

4. Bob calcule le hash du message, déchiffre avec Kpa et le compare avec celui envoyé.

=> Confidentialité / Intégrité des données / authentification / Non répudiation / facilité ??

Protocoles simples
Echange de clefs

Alice et Bob veulent mettre au point une clef secrète commune.

Si ils sont certains de parler l'un à l'autre, il est possible de définir cette clef de façon sûre même lorsque tout le monde les écoute !

Diffie-Hellman : p^(xy) = p^(yx)

Eve ne peut rien faire, alors qu'elle a écouté tous les échanges !

Que peut faire Mallory (?)

Remarques générales

- Dans ce qui précède, rien ne fonctionne ! Ce qui fonctionnera sera une combinaison de plusieurs méthodes.
  - *Utilisation de systèmes plus complets (PGP, SSL....)*

- La plupart du temps, les mots de passes sont vulnérables : ils sont trop simples : faciles à deviner. Il vaut mieux :
  - *une phrase mystère.*
  - *de l'aléatoire.*

- Prévoir une possibilité de répudiation de secret en cas de corruption de celui ci.

Remarques générales

- Cryptographie symétrique :
  - *Chiffrage / Déchiffrage rapide*
  - *Plus sûre (a taille de clef égale)*
  - *Utiliser des clefs temporaires (de session)*
- Cryptographie asymétrique :
  - *Sécuriser les clefs publiques (diffusion massive ou serveur sécurisé)*
  - *Chiffrage / Déchiffrage lent*
- Cryptographie hybride :
  - *Documents chiffrés par cryptographie symétrique*
  - *Clef secrète chiffrée par crypto asymétrique.*
- Toutes cryptographies :
  - *Limiter les redondances du message clair : compresser avant chiffrage !*

Protocoles intermédiaires
Signatures numériques

On veut qu'une signature d'un document :

- *Atteste de l'identité du signataire.*
- *Ne soit pas répudiable.*
- *Ne soit pas réutilisable (sur un autre document)*
- *Atteste de l'intégrité du document.*

Plusieurs solutions en fonction des cas :

- Le document doit il être lisible par tous ?
- Le document peut il être ré-utilisé ?

Protocoles intermédiaires
Signatures numériques

Remarques :

- La signature d'un document peut se faire en chiffrant le document avec une clef personnelle.
  - *Signature avec cryptographie symétrique : implique un tiers de confiance (sinon, qui veut vérifier doit connaître votre clef.)*
  - *Signature avec cryptographie asymétrique plus adaptée.*

- Pour que votre signature soit valable légalement, vous devez être enregistré quelque part !

- Eventuellement : faire un digest (hash) de votre document et signer ce digest. => rapidité et stockage

Protocoles intermédiaires
Signatures numériques

Signature d'un document lisible par tous :

- *1. Alice chiffre le message avec sa clef privée.*
- *2. Alice envoie le résultat à Bob.*

Avantages :

- Seule Alice a pu écrire ce message (si sa clef n'est pas corrompue)
- Quiconque dispose de sa clef publique peut le lire (Et peut en refaire un en prétendant en être a l'origine).
- Il faut éventuellement un dépositaire pouvant statuer sur la correspondance clef-Alice.

Protocoles intermédiaires
Signatures numériques

Ordre de virement : Alice veut envoyer a Bob un chèque numérique.

- *1. Alice prepare un ordre de virement.*
- *2. Alice le chiffre avec sa clef privée.*
- *3. Alice envoie le résultat à Bob.*
- *4. Bob dépose le tout a sa banque (qui déchiffre avec la clef d'Alice).*

Problème :

- Bob peut re-déposer le même message x fois...
- Nécessité d'ajouter un indice de non répétabilité (timestamp, beacon.....)

Protocoles intermédiaires
Signatures numériques

Dépôt de brevet : Alice veut poser un brevet sur une invention sans la révéler :

- *1. Alice calcule le digest du message décrivant son invention*
- *2. Alice signe ce digest avec sa clef privée.*
- *3. Alice envoie ce résultat a Trent qui le stocke.*

Avantages :

- Seule Alice conserve le document (faible stockage pour Trent)
- Elle peut faire valoir ses droit demandant à Trent de témoigner.
- => Cette méthode peut servir pour tous les cas de signature unilatérale sans connaissance du document.

Protocoles intermédiaires
Signatures numériques

Catégorie de problèmes complexes :

Le contrat de travail à deux signataires : (échange d'agrément)

- *1. Alice et son employeur Bob disposent de couples de clefs publiques, privée.*
- *2. Alice calcule le digest (hash) de son contrat.*
- *3. Alice signe ce digest avec sa clef privée et l'envoie a Bob. Bob chiffre le résultat avec sa clef privée et l'envoie.*
- *4. Chacun conserve le document et le digest doublement signé*

Problèmes :

- Sans vérification a l'aide des clefs publiques, on peut tricher (signer avec une clef secrète erronée !)
- Si Bob n'envoie pas sa version signée, il peut garder un contrat parfait => Problème d'atomicité et de rupture de protocole.

Protocoles intermédiaires
Signatures numériques

Catégorie de problèmes complexes :

Le contrat de travail à deux signataires : (échange d'agrément) avec Arbitre : L'arbitre confirme l'atomicité de l'agrément.

Problèmes :

- L'arbitre intervient de façon forte sur chaque signature : grande bande passante et goulot d'étranglement.

Protocoles intermédiaires
Certificats X.509

Sous réserve que la clef publique dont vous disposez soit bien celle de la personne à qui vous pensez parler, la crypto asymétrique est sure (pour l'échange de clefs secrètes par exemple)

D'ou des Certificates et des Certification Authority.

La CA est un tiers de confiance, sa clef publique est connue, sa clef privée est supposée fiable.

Par exemple : certificats de type X.509

Un certificat :

1. Des renseignements en clair : identification + clef publique + limites de validité + usage + identifiant de la CA.
2. Une signature de ce qui précède (un hash chiffré avec la clef privée de la CA)

Protocoles intermédiaires
Certificats X.509

Chaine de certification :

Pour ssl, la plupart des applications ont en « built-in », quelques certificats de CA precis. Ces certificats sont fait pour un usage de certification.

Lorsque vous recevez un certificat, les règles sont les suivantes

On accepte tout certificat signé par ces CA. On accepte aussi tout certificat signé par une CA dont le certificat a été signé par ces CA (avec un usage de certification).

=> décentralisation hiérarchique.

Protocoles intermédiaires
Certificats X509

Si vous recevez un certificat :

- vérifiez que vous faites confiance a la CA.
- vérifiez que le message est bien celui qui est signé.
- verifiez que le certificat est bien utilisé pour le bon usage.
- vérifiez que ce certificat n'a pas été révoqué (jamais fait !)

Si ce qui précède est fait, vous êtes sur de parler à la bonne personne, dans les bonnes conditions.

=> Discutez avec elle en chiffrant avec sa clef publique !

=> Vous pouvez devenir votre CA !

Systèmes complets

PGP : Mécanisme de chiffrage hybride.

Utilisé surtout pour

- les emails (s'ajoute a thunderbird...)
- le stockage de documents...

- Fonctionne sur la base d'un couple clef publique/clef privée supposé sur !

- Se distingue par la gestion des certificats : possibilités de CA, de chaines de CA mais surtout la notion de un réseau de confiance utilisant des certificats marginalements valides.

Aucun CA défini par défaut.

Systèmes complets
ssl

SSL est un protocole de chiffrement placé entre la couche transport (TCP) et la couche application (ftp, http, imap, pop....)

Notamment incluse dans tout échange de type https, ssh...

SSL Comporte 2 étapes :

SSL Handshake protocole (négociation)

SSL Record Protocol (echange de données)

Systèmes complets
ssl

SSL est un protocole de chiffrement placé entre la couche transport (TCP) et la couche application (ftp, http, imap, pop....)

Notamment incluse dans tout échange de type https, ssh...

- Compression
- Chiffrement hybride pour l'encryption.
- MAC pour chaque paquet (avant encryption)

Choix des algorithmes : les plus surs (algos et tailles declefs) parmi les algos disponibles chez les deux protagonistes.

Systèmes complets
ssl

SSL 2 : le serveur peut être authentifié (certificat X509)

SSL 3 : le serveur et le client peuvent être authentifiés.

Objectif majeur : éviter l'attaque Man in the middle !

- Celle ci se produit a l'échange de clefs publiques.

=> pour l'éviter : certificats ou stockages des serveurs déjà connus.

En fonction des besoins, l'un et l'autre peuvent arrêter le protocole si un danger est estimé (l'un n'est pas authentifié par son certificat).

Systèmes complets
ssl

Déroulement d'une connection ssl3 :

Handshake Protocol :

1. *Client envoie Hello :protocol v, random, session id, cypher suite, compression method*
2. *Server Hello : idem)*
3. *Server Certificate (opt) : certificat X.509*
4. *Server Key Exchange (opt if no certificate)*
5. *Certificate Request.*
6. *Server Hello done*
7. *CertificateVerify (opt si client has certificate with signing ability).*
8. *Client Change Cypher Spec*
9. *Client finished*
10. *Server Change Cypher Spec*
11. *Server Finished.*

Systèmes complets
ssl

A l'issue du Handshake :

Client et serveur connaissent les algos utilisés et disposent d'un d'une clef nommée prémaster secret. En fonction des cas, ce prémaster secret peut être :

- envoyé par le client (crypté par la clef publique du serveur)
- négociée par DH.

Ils calculent alors une clef (master secret). Ce master secret va etre utilisé pour :

- *générer la clef de signature des paquet de chaque coté.*
- *générer les clefs de chiffrement symétrique de chaque coté.*

Systèmes complets
ssl

Déroulement d'une connection ssl3 :

Record Protocol :

L'envoyeur :

1. *Envoyeur découpe son message en paquets qu'il compresse, hash, chiffre*
2. *Il envoie*

Le receveur

1. *déchiffre, vérifie le Hash, décompresse , recompose les paquets.*