Introduction à PGP

Daniel STAN

Cachan Réseau à Normale Sup'

Mardi 22 Mars 2016

Sommaire

Chiffrement symétrique

But : communiquer de manière sûre avec un interlocuteur connu. La clé de chiffrement est supposée connue des deux pairs.

Chiffrement symétrique

But : communiquer de manière sûre avec un interlocuteur connu. La clé de chiffrement est supposée connue des deux pairs.
Exemples d'algorithmes : AES, DES, 3DES, Blowfish, XOR

Hachage

But : Identifier *rapidement* une donnée initiale.
En pratique : Fonction de l'ensemble des données (infini) dans l'ensemble des hash (fini), difficilement inversible.

Hachage

But : Identifier *rapidement* une donnée initiale.
En pratique : Fonction de l'ensemble des données (infini) dans l'ensemble des hash (fini), difficilement inversible.
Exemples d'algorithmes : MD5, SHA1, SHA256

Hachage

But : Identifier *rapidement* une donnée initiale.
En pratique : Fonction de l'ensemble des données (infini) dans l'ensemble des hash (fini), difficilement inversible.
Exemples d'algorithmes : MD5, SHA1, SHA256
Utilisations : Mot de passe, salage, sûreté d'un téléchargement

Chiffrement asymétrique

Clé publique : Tout le monde la possède. Sert à chiffrer des messages qui ne sont pas déchiffrables sans clé privée.
Clé privée : gardée jalousement, elle permet de déchiffrer les messages.

Chiffrement asymétrique

Clé publique : Tout le monde la possède. Sert à chiffrer des messages qui ne sont pas déchiffrables sans clé privée.
Clé privée : gardée jalousement, elle permet de déchiffrer les messages.
Exemples d'algorithmes : RSA, ElGamal

Chiffrement asymétrique

Clé publique : Tout le monde la possède. Sert à chiffrer des messages qui ne sont pas déchiffrables sans clé privée.
Clé privée : gardée jalousement, elle permet de déchiffrer les messages.
Exemples d'algorithmes : RSA, ElGamal
En pratique, on préfère générer une clé symétrique, la chiffrer et l'envoyer, puis continuer avec la communication avec du chiffrement symétrique.

Chiffrement asymétrique

Clé publique : Tout le monde la possède. Sert à chiffrer des messages qui ne sont pas déchiffrables sans clé privée.
Clé privée : gardée jalousement, elle permet de déchiffrer les messages.
Exemples d'algorithmes : RSA, ElGamal
En pratique, on préfère générer une clé symétrique, la chiffrer et l'envoyer, puis continuer avec la communication avec du chiffrement symétrique.
Souvent plus rapide. Plus efficace si l'on chiffre pour plusieurs destinataires à la fois.

Pour la culture : principe de RSA

- On choisit deux (*grands*) entiers p et q premiers et distincts
- On pose $n = p \cdot q$
- On calcule $\varphi(n) =$

Pour la culture : principe de RSA

- On choisit deux (*grands*) entiers p et q premiers et distincts
- On pose $n = p \cdot q$
- On calcule $\varphi(n) = (p-1)(q-1)$

Pour la culture : principe de RSA

- On choisit deux (*grands*) entiers p et q premiers et distincts
- On pose $n = p \cdot q$
- On calcule $\varphi(n) = (p-1)(q-1)$
- Soit e entier inférieur à, et premier avec, $\varphi(n)$

Pour la culture : principe de RSA

- On choisit deux (*grands*) entiers p et q premiers et distincts
- On pose $n = p \cdot q$
- On calcule $\varphi(n) = (p-1)(q-1)$
- Soit e entier inférieur à, et premier avec, $\varphi(n)$
- Calculer $d \equiv e^{-1} \mod \varphi(n)$
- On remarque que pour tout M entier, $M^{e \cdot d} \equiv$

Pour la culture : principe de RSA

- On choisit deux (*grands*) entiers p et q premiers et distincts
- On pose $n = p \cdot q$
- On calcule $\varphi(n) = (p-1)(q-1)$
- Soit e entier inférieur à, et premier avec, $\varphi(n)$
- Calculer $d \equiv e^{-1} \mod \varphi(n)$
- On remarque que pour tout M entier, $M^{e \cdot d} \equiv M \mod n$.

Pour la culture : principe de RSA

- On choisit deux (*grands*) entiers p et q premiers et distincts
- On pose $n = p \cdot q$
- On calcule $\varphi(n) = (p-1)(q-1)$
- Soit e entier inférieur à, et premier avec, $\varphi(n)$
- Calculer $d \equiv e^{-1} \mod \varphi(n)$
- On remarque que pour tout M entier, $M^{e \cdot d} \equiv M \mod n$.

Clef publique : (n, e) Clef privée : d.

Signature

But : opération inverse : prouver son identité.
Clé privée : gardée jalousement, elle permet d'émettre (signer) des messages. Clé publique : Tout le monde la possède. Sert à vérifier les signatures, émises par la clé privée.

Signature

But : opération inverse : prouver son identité.
Clé privée : gardée jalousement, elle permet d'émettre (signer) des messages. Clé publique : Tout le monde la possède. Sert à vérifier les signatures, émises par la clé privée. Exemples d'algorithmes : DSA, RSA, ElGamal, courbes elliptiques.

Signature

But : opération inverse : prouver son identité.
Clé privée : gardée jalousement, elle permet d'émettre (signer) des messages. Clé publique : Tout le monde la possède. Sert à vérifier les signatures, émises par la clé privée. Exemples d'algorithmes : DSA, RSA, ElGamal, courbes elliptiques.
En pratique : on préfère signer le hash du message.

Introduction aux réseaux de confiance

Comment s'échanger les clés publiques initialement ?

- Rencontrer la personne en vrai

Introduction aux réseaux de confiance

Comment s'échanger les clés publiques initialement ?

- Rencontrer la personne en vrai
- Une clé c'est long : l'échanger grâce à une clé, au réseau local, etc

Introduction aux réseaux de confiance

Comment s'échanger les clés publiques initialement ?

- Rencontrer la personne en vrai
- Une clé c'est long : l'échanger grâce à une clé, au réseau local, etc
- La télécharger sur Internet, mais demander à son interlocuteur, dans la vraie vie, de confirmer le hash (empreinte).

Introduction aux réseaux de confiance

Comment s'échanger les clés publiques initialement ?

- Rencontrer la personne en vrai
- Une clé c'est long : l'échanger grâce à une clé, au réseau local, etc
- La télécharger sur Internet, mais demander à son interlocuteur, dans la vraie vie, de confirmer le hash (empreinte).
- Sinon : demander à un (des) intermédiaire(s) de confiance d'attester de l'authenticité de la clé (signatures).

Sommaire

GPG ou PGP ?

OpenPGP : format et protocole de cryptographie.

GPG ou PGP ?

OpenPGP : format et protocole de cryptographie.
GnuPG (GPG) : une des implémentations de OpenPGP, sous licence GNU GPL.

GPG ou PGP ?

OpenPGP : format et protocole de cryptographie.
GnuPG (GPG) : une des implémentations de OpenPGP, sous licence GNU GPL. `apt get install gnupg` (déjà installé sur Debian). Sur les autres OS . . ..

Pour générer une clef PGP

`--gen-key`

- Taille de clé : entre 2048 et 4096
- Plus la clef est longue, plus elle est dure à casser
- . . . mais également plus lourde
- . . . mais également plus longue à générer

Pour générer une clef PGP

`--gen-key`

- Taille de clé : entre 2048 et 4096
- Plus la clef est longue, plus elle est dure à casser
- . . . mais également plus lourde
- . . . mais également plus longue à générer

RSA : (ANSSI : taille 4096 bits pour un usage au delà de 2020).

Remarques sur la génération de clef

- La date d'expiration peut-être repoussée après génération (en cas de perte/vol de la clef)
- Identité : c'est le nom qui apparaîtra pour les autres utilisateurs, ainsi que le mail
- Conseil : indiquer votre adresse d'envoi habituelle
- Passphrase : dernière protection en cas de vol de la clef privée (permet uniquement de gagner un peu de temps en cas de vol).

Générer un certificat de révocation

```
--gen-revoke
gpg --output mon_certif_de_revocation.asc --gen-revoke
```

Une clef PGP est composée de :

`gpg --list-keys 6E1C820B`

- Une clé cryptographique publique de signature
- Des identités Nom/Mail
- Des sous-clefs (chiffrement, signature, authentification, certificat)
- Des signatures (`gpg --list-sigs 6E1C820B`)

Une clef PGP est composée de :

`gpg --list-keys 6E1C820B`

- Une clé cryptographique publique de signature
- Des identités Nom/Mail
- Des sous-clefs (chiffrement, signature, authentification, certificat)
- Des signatures (`gpg --list-sigs 6E1C820B`)
- Les clefs cryptographiques privées associées (si c'est la votre)

Publier sa clef PGP

- `gpg --export 6E1C820B --armour`
- Sur sa page perso, par mail, etc
- Ou alors . . .

Publier sa clef PGP

- `gpg --export 6E1C820B --armour`
- Sur sa page perso, par mail, etc
- Ou alors . . .utiliser un serveur de clé !
- `gpg --send-key 6E1C820B`

Récupérer la clé PGP de quelqu'un

- Sur sa page perso, par mail, etc
- `gpg --import`
- Ou alors . . .

Récupérer la clé PGP de quelqu'un

- Sur sa page perso, par mail, etc
- `gpg --import`
- Ou alors . . .utiliser un serveur de clé !
- `gpg --recv-key 6E1C820B`

Récupérer la clé PGP de quelqu'un

- Sur sa page perso, par mail, etc
- `gpg --import`
- Ou alors . . .utiliser un serveur de clé !
- `gpg --recv-key 6E1C820B`
- `gpg --search-keys dstan`

Récupérer la clé PGP de quelqu'un

- Sur sa page perso, par mail, etc
- `gpg --import`
- Ou alors . . .utiliser un serveur de clé !
- `gpg --recv-key 6E1C820B`
- `gpg --search-keys dstan`

Attention : vous n'avez a priori aucune raison d'avoir confiance en cette clef !

Comment vérifier qu'une clef appartient bien à son propriétaire ?

- Rencontrer la personne en vrai, et échanger physiquement la clef

Comment vérifier qu'une clef appartient bien à son propriétaire ?

- Rencontrer la personne en vrai, et échanger physiquement la clef
- La télécharger (Internet) et échanger physiquement le hash (empreinte).

Comment vérifier qu'une clef appartient bien à son propriétaire ?

- Rencontrer la personne en vrai, et échanger physiquement la clef
- La télécharger (Internet) et échanger physiquement le hash (empreinte).

Dans tous les cas, il faut rencontrer physiquement la personne.

Comment vérifier qu'une clef appartient bien à son propriétaire ?

- Rencontrer la personne en vrai, et échanger physiquement la clef
- La télécharger (Internet) et échanger physiquement le hash (empreinte).

Dans tous les cas, il faut rencontrer physiquement la personne.

Attention : ne pas confondre ID et empreinte de clef !

Intérêt de la signature

- Attester aux autres de l'identité d'une clef

Intérêt de la signature

- Attester aux autres de l'identité d'une clef
- Mais au fond, qu'est-ce qu'une identité ?

Intérêt de la signature

- Attester aux autres de l'identité d'une clef
- Mais au fond, qu'est-ce qu'une identité ?On vérifie souvent une pièce d'identité émise par l'État.

Intérêt de la signature

- Attester aux autres de l'identité d'une clef
- Mais au fond, qu'est-ce qu'une identité ?On vérifie souvent une pièce d'identité émise par l'État.
- La manière dont vous signez une personne peut influencer la confiance qu'elle accordera à votre clé !

Une procédure de signature

On passe en mode édition avec `gpg --edit-key`

- `fpr` (se faire dicter la clef)
- Vérifier une pièce d'identité
- `sign`
- `trust`
- `save`

Une procédure de signature

On passe en mode édition avec `gpg --edit-key`

- `fpr` (se faire dicter la clef)
- Vérifier une pièce d'identité
- `sign`
- `trust`
- `save`

Renvoyer la clef signée : `gpg --send-key 6E1C820B` (ou `gpg --export 6E1C820B`).

Une procédure de signature

On passe en mode édition avec `gpg --edit-key`

- `fpr` (se faire dicter la clef)
- Vérifier une pièce d'identité
- `sign`
- `trust`
- `save`

Renvoyer la clef signée : `gpg --send-key 6E1C820B` (ou `gpg --export 6E1C820B`). NB : d'autres méthodes de signature sont envisageables quand de nombreuses personnes veulent s'entre-signer.

Trousseau de clefs

- Situé dans `~/.gnupg/`
- Contient les clefs privées
- Contient les clés téléchargées

Trousseau de clefs

- Situé dans `~/.gnupg/`
- Contient les clefs privées
- Contient les clés téléchargées
- Et les niveau de confiance !

Pensez à mettre à jour :

- Les clefs de votre trousseau : `gpg --refresh-keys`
- Sa base de confiance : `gpg --update-trust-db`

Sommaire

- `gpg -e -r dstan --armour` nécessite une confiance suffisante en la clé destination

- `gpg -e -r dstan --armour` nécessite une confiance suffisante en la clé destination
- `gpg -d` nécessite de posséder la clef privée associée

- `gpg -e -r dstan --armour` nécessite une confiance suffisante en la clé destination
- `gpg -d` nécessite de posséder la clef privée associée
- Signer `gpg -s`

- `gpg -e -r dstan --armour` nécessite une confiance suffisante en la clé destination
- `gpg -d` nécessite de posséder la clef privée associée
- Signer `gpg -s`
- Version claire : `gpg --clearsign`

Envoyer des mails

- Enigmail (Thunderbird)
- Sylpheed
- Claws Mail
- etc.

Cpassword

- Projet Cranseux de partage de mot de passe
- Plusieurs *roles* : nounous, apprentis, ca
- Plusieurs fichiers
- Il faut être signé par la personne qui chiffre les mots de passe

Cpassword

- Projet Cranseux de partage de mot de passe
- Plusieurs *roles* : nounous, apprentis, ca
- Plusieurs fichiers
- Il faut être signé par la personne qui chiffre les mots de passe

Prérequis :

- Git, Make, etc
- Une clé ssh
- Une clé gpg
- Indiquer la clé dans `gest_crans`
- Pouvoir se connecter en ssh sur `ldap.adm.crans.org`

Ce dont on n'a pas parlé (liste non exhaustive)

- Comment garder sa clé maîtresse à l'abris
- Créer des sous-clés de chiffrement
- Mettre à jour les dates d'expiration
- Batch signer plusieurs personnes et valider leur mail
- Utiliser une smartcard
- Comment gérer/personaliser son réseau de confiance.

Quelques liens

- https://wiki.crans.org/CransTechnique/CransApprentis/SeminairesTechniques/2013-2014?action=AttachFile&do=get&target=gnupg.pdf
- http://doc.ubuntu-fr.org/gnupg
- http://fr.wikibooks.org/wiki/GPG
- http://matrix.samizdat.net/crypto/gpg_intro/
- http://www.gnupg.org/gph/fr/manual.html
- http://www.legifrance.gouv.fr/
- http://fr.wikipedia.org/wiki/Cryptographie
- http://security.stackexchange.com/questions/5096/
- rsa-vs-dsa-for-ssh-authentication-keys
- http://www.linuxquestions.org/questions/linux-security-4/
- gpg-rsa-or-dsa-with-el-gamal-for-new-keys-565242/
- http://docu.fsugar.be/openpgp/openpgp.html
- http://wiki.debian.org/subkeys