**Travaux pratiques**

**TP3 : Sécurité de bout en bout de la messagerie électronique**

Plan du TP

[**1.** **Objectifs du TP** 4](#_Toc55574247)

[**2.** **Prérequis du TP** 4](#_Toc55574248)

[**3.** **Introduction** 4](#_Toc55574249)

[**a)** **Pourquoi utiliser PGP et S/MIME ?** 4](#_Toc55574250)

[**b)** **Mécanismes de PGP et S/MIME** 4](#_Toc55574251)

[**c)** **Logiciels PGP** 4](#_Toc55574252)

[**d)** **Logiciels PGP utilisés dans le TP** 5](#_Toc55574253)

[**e)** **Visualiser les algorithmes supportés** 5](#_Toc55574254)

[**f)** **Configurer Thunderbird** 5](#_Toc55574255)

[**4.** **Créer des clés PGP** 6](#_Toc55574256)

[**a)** **Créer sa paire de clés PGP** 6](#_Toc55574257)

[**b)** **Autocrypt et Pretty Easy Privacy** 6](#_Toc55574258)

[**c)** **Ajout d’Enigmail à Thunderbird et création de biclés** 7](#_Toc55574259)

[**d)** **Visualiser les informations sur sa clé PGP** 7](#_Toc55574260)

[**e)** **Analyser le contenu de sa clé PGP** 7](#_Toc55574261)

[**5.** **Envoyer des mails sécurisés par PGP** 8](#_Toc55574262)

[**a)** **Signer un fichier** 8](#_Toc55574263)

[**b)** **Vérifier un fichier signé : résultat bonne signature** 8](#_Toc55574264)

[**c)** **Vérifier un fichier signé : résultat mauvaise signature** 8](#_Toc55574265)

[**d)** **S’envoyer un mail signé et/ou chiffré** 8](#_Toc55574266)

[**e)** **Chiffrement de l’objet des mails** 8](#_Toc55574267)

[**f)** **Analyser un mail signé** 9](#_Toc55574268)

[**g)** **Analyser un mail chiffré** 9](#_Toc55574269)

[**h)** **Envoyer un mail signé** 9](#_Toc55574270)

[**i)** **Échanger sa clé publique PGP par mail** 10](#_Toc55574271)

[**j)** **Envoyer un mail chiffré** 10](#_Toc55574272)

[**6.** **Utiliser un serveur de clés PGP** 10](#_Toc55574273)

[**a)** **Configurer le serveur de clés à utiliser** 10](#_Toc55574274)

[**b)** **Exporter sa clé vers le serveur de clés** 10](#_Toc55574275)

[**c)** **Importer des clés depuis le serveur de clés** 10](#_Toc55574276)

[**7.** **Déterminer la confiance dans les clés** 11](#_Toc55574277)

[**a)** **Risque de contrefaçon des clés** 11](#_Toc55574278)

[**b)** **Certification des clés** 11](#_Toc55574279)

[**8.** **Certifier des clés PGP** 11](#_Toc55574280)

[**a)** **Certification des clés** 11](#_Toc55574281)

[**b)** **Validité des clés PGP** 12](#_Toc55574282)

[**c)** **Visualiser la validité d’une clé PGP** 12](#_Toc55574283)

[**d)** **Empreintes digitales des clés PGP** 12](#_Toc55574284)

[**e)** **Visualiser une empreinte digitale** 12](#_Toc55574285)

[**f)** **Vérification des clés publiques PGP** 12](#_Toc55574286)

[**g)** **Vérification des clés publiques PGP** 13](#_Toc55574287)

[**h)** **Certification des clés publiques PGP** 13](#_Toc55574288)

[**i)** **Certifier une clé publique PGP** 13](#_Toc55574289)

[**j)** **Visualiser les certificats des clés publiques** 14](#_Toc55574290)

[**k)** **Utiliser une chaîne de confiance PGP** 14](#_Toc55574291)

[**9.** **Révoquer des clés PGP** 14](#_Toc55574292)

[**a)** **Révoquer sa clé publique PGP** 14](#_Toc55574293)

[**b)** **Révoquer sa clé publique PGP** 14](#_Toc55574294)

[**c)** **Prendre en compte la révocation des clés PGP** 15](#_Toc55574295)

[**10.** **Créer des certificats S/MIME** 15](#_Toc55574296)

[**a)** **Certification des clés S/MIME** 15](#_Toc55574297)

[**b)** **Fonctions d’une PKI** 15](#_Toc55574298)

[**c)** **Logiciels S/MIME utilisés dans le TP** 16](#_Toc55574299)

[**d)** **Créer sa paire de clés S/MIME** 16](#_Toc55574300)

[**e)** **Afficher les composants de sa clé privée** 16](#_Toc55574301)

[**f)** **Créer un CSR (Certificat Signing Request)** 16](#_Toc55574302)

[**g)** **Afficher le CSR** 17](#_Toc55574303)

[**h)** **Récupérer les fichiers de l’AC** 17](#_Toc55574304)

[**i)** **Créer son certificat S/MIME** 17](#_Toc55574305)

[**j)** **Afficher le certificat S/MIME** 17](#_Toc55574306)

[**k)** **Format PKCS #12** 17](#_Toc55574307)

[**l)** **Exporter clé privée et certificat en PKCS #12** 18](#_Toc55574308)

[**11.** **Envoyer des mails sécurisés par S/MIME** 18](#_Toc55574309)

[**a)** **Importer clés et certificats dans le magasin** 18](#_Toc55574310)

[**b)** **Configurer Thunderbird** 18](#_Toc55574311)

[**c)** **S’envoyer un mail signé et/ou chiffré** 18](#_Toc55574312)

[**d)** **Envoyer un mail signé et/ou chiffré** 19](#_Toc55574313)

[**e)** **La révocation des certificats S/MIME** 19](#_Toc55574314)

[**f)** **Envoyer le rapport du TP** 19](#_Toc55574315)

1. **Objectifs du TP**

L'objectif de ce TP est de mettre en œuvre différents services de sécurité, essentiellement **confidentialité** et **authentification**, dans la messagerie électronique. On utilisera pour cela différents logiciels supportant les standards IETF **OpenPGP** et **S/MIME**.

1. **Prérequis du TP**

-Avoir installé Ubuntu 20.04

-Avoir une adresse mail Microsoft du type : [prénom.nom@hotmail.com](mailto:prénom.nom@hotmail.com)

-Avoir une connexion Internet

1. **Introduction**

* PGP et S/MIME sont deux techniques permettant de sécuriser les échanges de mail de **bout en bout**
* Signature
* Chiffrement
* Ce sont des standards IETF (RFC)
* OpenPGP : Open Pretty Good Privacy
* S/MIME : Secure/Multipurpose Internet Mail Extensions

1. **Pourquoi utiliser PGP et S/MIME ?**

* Problèmes de sécurité liés à la messagerie électronique
* Écoutes : sniffers
* Usurpation d’identité : « telnet 25 »
* Litiges : répudiation par l’émetteur ou le destinataire
* Modification des mails en transit : attaques de type « Man in the Middle »
* Services de sécurité offerts par PGP et S/MIME
* Confidentialité par chiffrement
* Authentification par signature électronique
* Non-répudiation de l’émetteur par signature électronique
* Intégrité par signature électronique

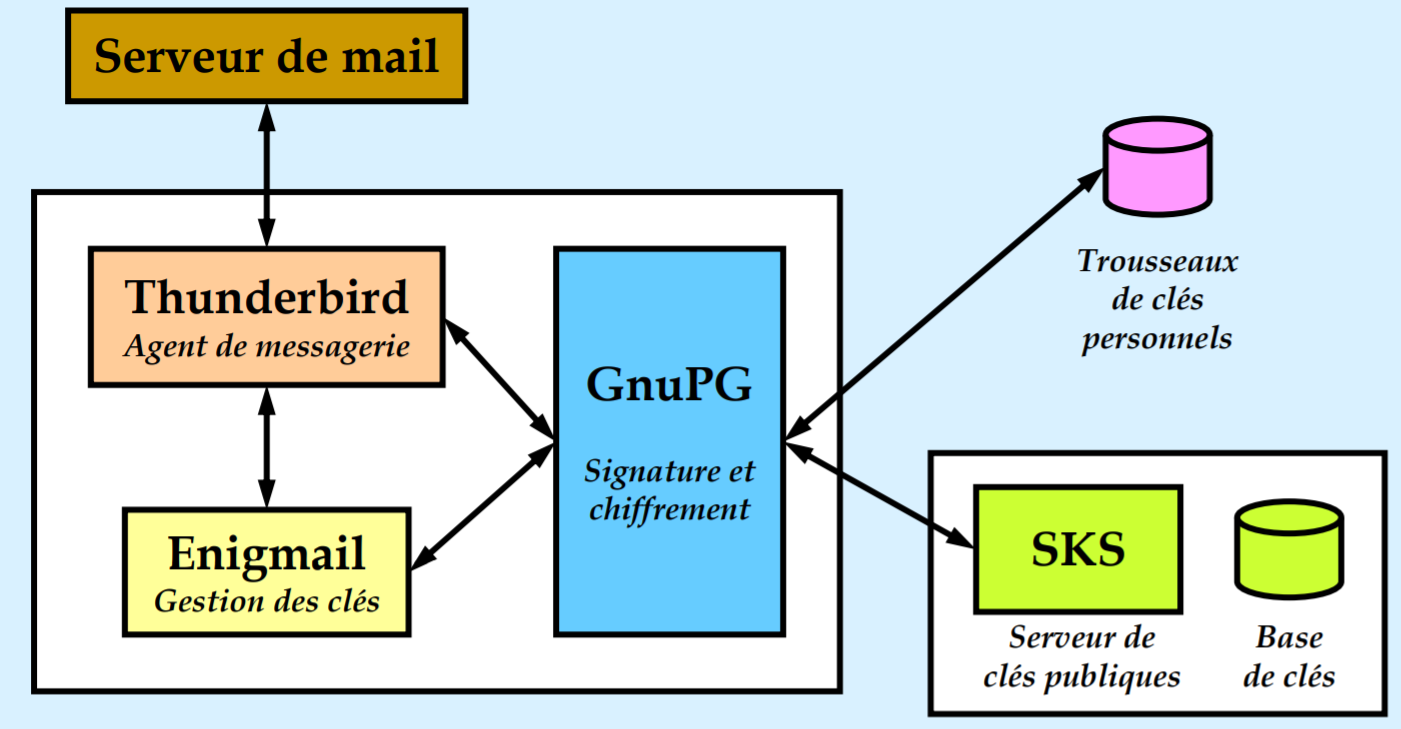
1. **Mécanismes de PGP et S/MIME**

* Les architectures de messagerie utilisent différents mécanismes de sécurité
* Les échanges entre agents de messagerie et serveurs de mail, ainsi que les échanges entre serveurs de mail, doivent être protégés au moyen des protocoles TLS/SSL lorsque c’est possible
* PGP et S/MIME offrent en complément un service de sécurité de niveau applicatif entre les individus (de bout en bout)
* PGP et S/MIME utilisent plusieurs techniques
* Cryptographie asymétrique
* Cryptographie symétrique
* Fonctions de hachage
* Certification des clés publiques

1. **Logiciels PGP**

* Logiciels PGP
* Logiciel libre : GnuPG (GNU Privacy Guard)
* Logiciel commercial : PGP (Symantec)
* Ces logiciels s’intègrent dans les agents de messagerie au moyen de modules complémentaires (plugins)
* Thunderbird : Enigmail
* Outlook, Outlook Express : GpgOL (inclus dans Gpg4win)
* Zimbra : zimlet OpenPGP (à installer sur le serveur)
* Des plugins existent pour Eudora, Lotus Notes…
* Pour les webmails, des extensions existent pour les différents navigateurs web (Mailvelope, WebPG…)

1. **Logiciels PGP utilisés dans le TP**



1. **Visualiser les algorithmes supportés**

* Ouvrez un terminal dans la VM
* Visualisez les algorithmes disponibles dans votre version du logiciel GnuPG : gpg –-version
* Identifiez les algorithmes de cryptographie asymétrique, symétrique, de hachage et de compression

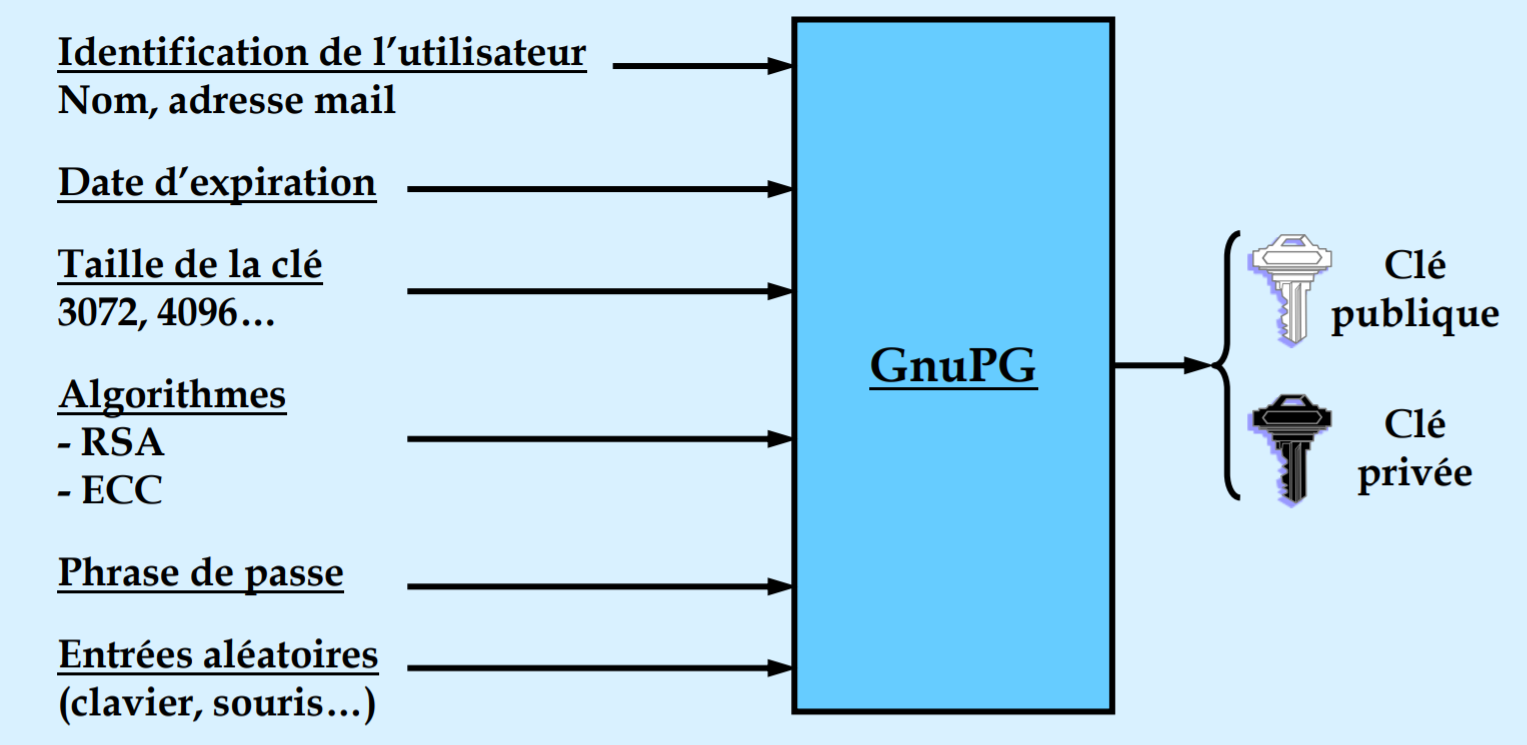
1. **Configurer Thunderbird**

* Lancez Thunderbird dans la VM
* Annulez la fenêtre demandant un mot de passe
* Configurez votre compte de messagerie : Édition / Paramètres des comptes
  + Nom : Prénom Nom (Formation/Année)
  + Adresse électronique : prenom.nom@hotmail.com
* Configurez le serveur entrant : Paramètres serveur
  + Nom d’utilisateur : prenom.nom@hotmail.com
* Configurez le serveur sortant : Serveur sortant (SMTP) / Modifier…
  + Nom d’utilisateur : prenom.nom@hotmail.com
* Validez et affichez le contenu de votre boîte aux lettres
* Entrez votre mot de passe de messagerie et cochez l’option permettant de conserver le mot de passe en cache dans le gestionnaire de mots de passe
* Testez en vous envoyant un mail à votre adresse et cochez à nouveau la conservation du mot de passe

Pour installer thunderbird sur ubuntu utiliser la commande :

$ sudo apt install thunderbird

1. **Créer des clés PGP**



1. **Créer sa paire de clés PGP**

* Gestion des clés privées par GnuPG
* Les clés privées sont enregistrées dans un fichier qui ne doit être accessible qu’à l’utilisateur : droits d’accès du système d’exploitation, utilisation d’une clé USB externe ou d’une smartcard…
* Ce fichier est également chiffré avec la phrase de passe
* La phrase de passe est demandée lorsque GnuPG doit accéder à la clé privée
* Les clés privées sont conservées dans un cache de GnuPG pendant un certain temps (par défaut, 2h avec un temporisateur d’inactivité de 10 mn, configurable dans les préférences de GnuPG et d’Enigmail)
* Ceci permet d’éviter d’avoir à entrer sa phase de passe trop souvent
* Gestion des mots de passe par GNOME
* Sous Linux, l’environnement GNOME propose aussi de mémoriser les mots de passe (processus « gnome-keyring-daemon »)
* Le bouton « Enregistrer dans le gestionnaire de mots de passe » affiché lors de la saisie d’une phrase de passe correspond au gestionnaire de GNOME
* Il est conseillé de ne pas utiliser le gestionnaire de GNOME pour les phrases de passe PGP, car il présente un niveau de sécurité moindre (les mots de passe sont chiffrés avec le mot de passe de connexion)

1. **Autocrypt et Pretty Easy Privacy**

* Deux techniques permettent de simplifier l’utilisation de PGP pour le grand public
* Autocrypt détecte lorsque deux personnes utilisent PGP, les mails qu’ils échangent sont alors automatiquement chiffrés et signés et les clés publiques sont envoyées dans les en-têtes des mails
* Pretty Easy Privacy (p≡p) génère automatiquement les bi-clés (sans phrase de passe), les mails sont automatiquement chiffrés et les clés publiques sont envoyées en attachement des mails
* Dans Enigmail, p≡p est activé par défaut pour les nouveaux utilisateurs (mode junior), et Autocrypt est activé par défaut si p≡p ne l’est pas (Privacy by Default)
* Dans le cadre de ce TP, on désactive Autocrypt et p≡p de manière à conserver une gestion manuelle de ses clés

1. **Ajout d’Enigmail à Thunderbird et création de biclés**

Ajouter Enigmail avec la commande (N’oublier pas de fermer Thunderbird pour que l’extension soit apparue dans la barre d’outils):

$ sudo apt install enigmail

* Ouvrez l’outil de gestion des clés d’Enigmail :
  + Enigmail / Gestion des clés
* Cet outil permet de gérer son trousseau de clés : créer des paires de clés, importer et exporter des clés, afficher les informations sur des clés, supprimer des clés…
* Il est possible qu’une paire de clés ait déjà été créée automatiquement par Pretty Easy Privacy
* Si c’est le cas, vous allez supprimer cette paire de clés avant d’en créer une nouvelle (avec une phrase de passe)
* Si une paire de clés PGP est déjà présente dans votre trousseau de clés, effectuez un clic droit sur cette clé
* Supprimer la clé
* Créez votre paire de clés PGP :
  + Générer / Nouvelle biclé
* Entrez les informations suivantes :
  + Phrase de passe (deux fois)
  + Expiration de la clé
  + Onglet « Avancé… » : type de clé et taille de la clé
* Ne créez pas de certificat de révocation pour le moment

1. **Visualiser les informations sur sa clé PGP**

* Effectuez un double-clic sur votre clé
* Vérifiez les informations générales sur votre clé : identifiant utilisateur, date de création et d’expiration
* Cliquez sur l’onglet « Structure »
* Observez la structure de votre clé, qui se compose d’une clé principale de signature et d’une sous-clé de chiffrement

1. **Analyser le contenu de sa clé PGP**

* Enregistrez votre clé publique dans un fichier
* Effectuez un clic droit sur votre clé
* Exporter les clés vers un fichier
* Choisissez l’option « Exporter les clés publiques seulement »
* Enregistrez votre clé dans un fichier ma-cle.asc
* Analysez le contenu de votre clé publique
* Affichez le contenu binaire de votre clé publique avec l’utilitaire PGPdump :

$ pgpdump -i ma-cle.asc

* Affichez le contenu binaire de votre clé publique avec GnuPG :

$ gpg --list-packets --verbose ma-cle.asc

* Les paramètres constituant la clé publique sont notés pkey[0], pkey[1]…
* Il se peut qu’on aura besoin d’installer pgpdump, pour se faire tapper la commande suivante sous ubuntu : $ sudo apt-get install pgpdump

1. **Envoyer des mails sécurisés par PGP**
2. **Signer un fichier**

* Créez un fichier texte abcd.txt contenant quelques mots : $ gedit abcd.txt
* Signez le fichier abcd.txt : $ gpg --clearsign abcd.txt
* Le processus de signature nécessitant votre clé privée, vous devrez donc entrer votre phrase de passe PGP
* Le fichier signé est enregistré dans le fichier abcd.txt.asc, il contient le texte en clair et sa signature numérique
* Visualisez ce fichier : $ cat abcd.txt.asc

1. **Vérifier un fichier signé : résultat bonne signature**

* Supprimez le fichier texte initial abcd.txt
* Vérifiez la signature du fichier signé abcd.txt.asc : $ gpg --verify abcd.txt.asc
* Cette commande fournit un résultat parmi les trois suivants : «Bonne signature», « Mauvaise signature » ou « Clé publique non trouvée »
* Vérifiez que le résultat obtenu est « Bonne signature »

1. **Vérifier un fichier signé : résultat mauvaise signature**

* Simulez une attaque de type « Man in the Middle » en modifiant le texte en clair dans le fichier signé abcd.txt.asc : $ gedit abcd.txt.asc
* Vérifiez à nouveau la signature du fichier signé : $ gpg --verify abcd.txt.asc
* Vérifiez que le résultat obtenu est « Mauvaise signature »

1. **S’envoyer un mail signé et/ou chiffré**

* Depuis thunderbird Envoyez à vous-même un mail signé en sélectionnant l’icône de signature
* Lisez le mail signé reçu
* Envoyez à vous-même un mail chiffré en sélectionnant l’icône de chiffrement (choisir l’option « Protéger l’objet »)
* Lisez le mail chiffré reçu
* Envoyez enfin à vous-même un mail à la fois signé et chiffré
* Lisez le mail signé et chiffré reçu
* Cliquez sur les logos en forme d’enveloppe ou de verrou à droite pour afficher les informations de sécurité : identifiant et empreinte de la clé, algorithmes utilisés…

1. **Chiffrement de l’objet des mails**

* Lors du premier envoi d’un mail chiffré, Enigmail propose de chiffrer également l’objet du mail en plus de son contenu
* Ce choix est ensuite enregistré dans les préférences d’Enigmail (Enigmail / Préférences, Avancé, Chiffrer l’objet par défaut)
* Dans ce cas les mails chiffrés présenteront comme objet un texte standard non significatif au lieu de l’objet réel
* L’objet réel du mail ne pourra être affiché que par le destinataire après déchiffrement du contenu du mail avec sa clé privée

1. **Analyser un mail signé**

* Effectuez un clic droit sur le mail signé reçu
* Enregistrer sous…
* Enregistrez le mail signé dans le fichier **mail-signe.eml**
* Afficher le contenu du mailsigné avec la commande :

$ **cat mail-signe.eml**

* Affichez le contenu binaire du mail signé avec l’utilitaire PGPdump :

**$ pgpdump -i mail-signe.eml**

* Identifiez les informations contenues dans le fichier signé :
  + Algorithme asymétrique utilisé
  + Algorithme de hachage utilisé
  + Date de la signature
  + Identifiant de la clé publique utilisée
  + Contenu binaire de la signature (texte en clair haché, puis chiffré avec l’algorithme asymétrique)

1. **Analyser un mail chiffré**

* Effectuez un clic droit sur le mail chiffré reçu
* Enregistrer sous…
* Enregistrez le mail chiffré dans le fichier **mail-chiffre.eml**
* Affichez le contenu du mail chiffré avec la commande :

**$ cat mail-chiffre.eml**

* Affichez le contenu binaire du mail chiffré avec PGPdump :

**$ pgpdump -i mail-chiffre.eml**

* Identifiez les informations contenues dans le mail chiffré :
  + Première partie : clé de session symétrique chiffrée avec l’algorithme asymétrique (identifiant de la clé publique, algorithme asymétrique utilisé et contenu binaire de la clé de session chiffrée)
  + Deuxième partie : contenu du mail chiffré avec la clé de session symétrique
* Affichez la clé de session symétrique utilisée
* Déchiffrez le mail chiffré en affichant la clé de session utilisée :

**$ gpg --decrypt --show-session-key mail-chiffre.eml**

* Le nombre indiqué avant le caractère « : » dans la clé de session indique l’algorithme symétrique utilisé (2 pour 3DES, 7 pour AES-128, 9 pour AES-256…)

1. **Envoyer un mail signé**

* Envoyez à votre interlocuteur un mail signé
* Lisez le mail signé reçu de votre interlocuteur
* La vérification de la signature n’est pas possible car vous ne disposez pas actuellement de la clé publique de votre interlocuteur
* Un bouton permet d’importer une clé manquante depuis des serveurs de clés prédéfinis
* Dans un premier temps, vous allez échanger votre clé publique avec votre interlocuteur par mail

1. **Échanger sa clé publique PGP par mail**

* Composez un nouveau mail à votre interlocuteur, puis cliquez sur Enigmail / Joindre ma clé publique
* Lisez le nouveau mail reçu de votre interlocuteur
* Effectuez un clic droit sur la pièce jointe (fichier d’extension « .asc » contenant sa clé publique)
* Importer une clé OpenPGP
* Vérifiez dans Enigmail que la clé publique de votre interlocuteur apparaît dans votre trousseau de clés publiques
* Lisez à nouveau le mail signé reçu précédemment de votre interlocuteur et vérifiez que la signature est correcte

1. **Envoyer un mail chiffré**

* Envoyez à votre interlocuteur un mail chiffré
* Lisez le mail chiffré reçu de votre interlocuteur
* Envoyez enfin à votre interlocuteur un mail à la fois signé et chiffré
* Lisez le mail signé et chiffré reçu de votre interlocuteur

**NB** : Au cas où on a un échoue d’envoi :

-Ilse peut qu’il y’a un problème de connexion internet, vérifier votre connexion (ex : ping 8.8.8.8)

-Il se peut qu’il y’a un problème avec la clé privée, régénérer une paire de clés

1. **Utiliser un serveur de clés PGP**

* Des serveurs de clés PGP permettent de déposer sa clé publique afin de la rendre accessible en ligne
* Dans ce TP, vous utiliserez un serveur SKS dédié
* Notez l’adresse IP du serveur de clés qui vous est fournie : a.b.c.d
* Vérifiez que le serveur de clés est effectivement accessible : ping a.b.c.d
* Affichez dans un navigateur web la page d’accueil du serveur de clés en entrant l’url : a.b.c.d
* Entrez une chaîne de caractères dans le champ de recherche
* Validez pour visualiser les clés correspondantes
* Effectuez des recherches similaires en affichant les signatures (verbose index), les empreintes digitales et le contenu ASCII des clés

1. **Configurer le serveur de clés à utiliser**

* Enigmail / Préférences, onglet « Serveur de clés »
* Indiquez comme liste de serveurs de clés : hkp:// a.b.c.d :11371

1. **Exporter sa clé vers le serveur de clés**

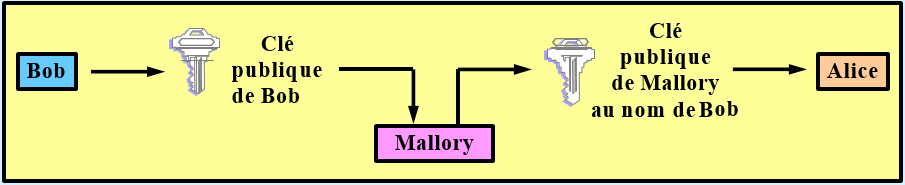
* Enigmail / Gestion des clés, Effectuez un clic droit sur votre clé publique
* Téléverser les clés publiques vers un serveur de clés
* Vérifiez dans le navigateur web que votre clé publique apparaît effectivement sur le serveur de clés

1. **Importer des clés depuis le serveur de clés**

* Enigmail / Gestion des clés /Serveur de clés / Chercher des clés
* Entrez le texte recherché : Formation/Année
* Sélectionnez l’ensemble des clés correspondantes et validez
* Vérifiez que les clés publiques de l’ensemble des étudiants du groupe ont effectivement été importées dans votre trousseau de clés publiques

1. **Déterminer la confiance dans les clés**
2. **Risque de contrefaçon des clés**

* **L’association entre une clé et une adresse mail n’est pas vérifiée**
* Lorsqu’une personne crée une paire de clés associée à une adresse mail, on ne vérifie pas que cette personne est effectivement propriétaire de l’adresse mail
* Un attaquant peut donc créer une paire de clés en l’associant à l’adresse d’une autre personne (contrefaçon)
* **Une clé contrefaite peut ensuite être diffusée sur des serveurs de clés**
* **L’attaquant peut aussi insérer une clé contrefaite dans le cas d’un échange de clé par mail (attaque de type MITM)**

****

* **Si l’attaquant réussit à diffuser une clé contrefaite :**
* Il pourra signer des mails au nom de la victime
* Il pourra déchiffrer des mails chiffrés envoyés à la victime

1. **Certification des clés**

* **Les clés doivent donc être certifiées pour éviter les contrefaçons**
* La certification permet de prouver qu’une clé donnée appartient bien à la personne possédant l’adresse mail associée à la clé
* L’association entre la clé et l’adresse mail doit être signée avec une clé de confiance
* **Les deux techniques les plus utilisées sont PGP et S/MIME**
* Leur fonctionnement est identique en ce qui concerne les fonctions de signature et de chiffrement (cryptographie asymétrique)
* Elles se distinguent par la manière de certifier les clés : elles sont certifiées par chaque utilisateur dans le cas de PGP et par une autorité de certification dans le cas de S/MIME

1. **Certifier des clés PGP**
2. **Certification des clés**

* **La certification des clés PGP**
* Chaque utilisateur valide lui-même les clés publiques de ses interlocuteurs
* Après avoir validé la clé publique d’un interlocuteur, l’utilisateur la signe avec sa propre clé privée, ce qui permet de créer un certificat pour la clé publique de cet interlocuteur
* Le format des certificats utilisés est un format spécifique à PGP (standard IETF)
* **Du fait de ce mode de fonctionnement, PGP est surtout utilisé par des particuliers et non des entreprises**

1. **Validité des clés PGP**

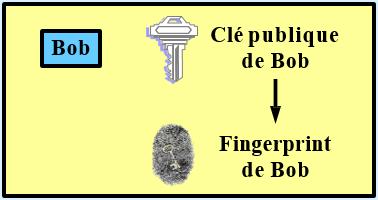
* **La validité des clés PGP peut prendre les valeurs suivantes**
* Confiance absolue : sa propre clé
* Pleine confiance : clé signée par soi-même (ou par un utilisateur dans lequel on a une pleine confiance)
* Confiance modérée : clé signée par un utilisateur dans lequel on a une confiance modérée
* Révoquée
* Expirée
* Inconnue

1. **Visualiser la validité d’une clé PGP**

* Enigmail / Gestion des clés, Vérifiez que votre propre clé a la validité « absolue » et que les autres clés ont une validité inconnue (représentée par un tiret « - »)

1. **Empreintes digitales des clés PGP**

* **Une empreinte digitale ou fingerprint est le résultat d'un calcul effectué sur une clé publique**
* Ce calcul correspond à une fonction de hachage cryptographique
* L’empreinte digitale est plus petite que la clé publique
* Si deux clés ont la même empreinte digitale, elles ont une très forte probabilité d’être identiques



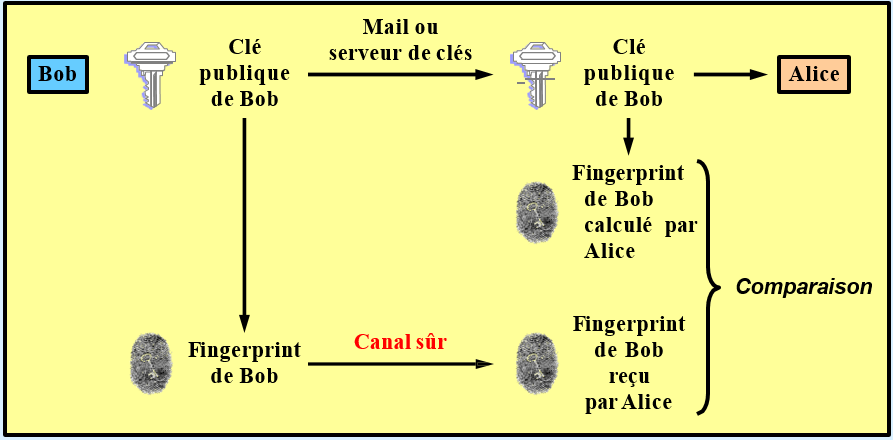
1. **Visualiser une empreinte digitale**

* Enigmail / Gestion des clés, Effectuez un double-clic sur la clé publique de votre interlocuteur
* Visualisez l’empreinte digitale de sa clé publique

1. **Vérification des clés publiques PGP**

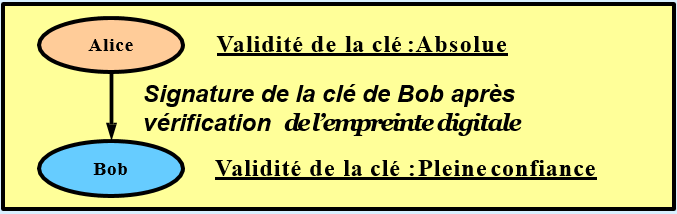
* **Vérification de l’empreinte digitale lors de la réception** **d’une clé**
* Bob envoie sa clé publique à Alice
* Pour éviter une contrefaçon de la clé de Bob lors de son envoi, Alice doit vérifier que la clé reçue est effectivement la clé envoyée par Bob
* Alice calcule l’empreinte digitale de la clé reçue de Bob
* Bob transmet à Alice l’empreinte digitale de sa clé au moyen d’un canal sûr : par téléphone, lors d’une rencontre en face à face, lors de « key-signing parties »…
* Alice compare l’empreinte digitale calculée à partir de la clé reçue de Bob avec l’empreinte digitale de la clé de Bob transmise par ce canal sûr

1. **Vérification des clés publiques PGP**



1. **Certification des clés publiques PGP**

* **Une fois qu’Alice a vérifié l’empreinte digitale de la clé reçue de Bob, elle certifie cette clé en la signant**
* Cette certification atteste que la clé publique de Bob appartient effectivement à Bob
* La validité de la clé de Bob devient « Pleine confiance » pour Alice
* Cette certification pourra être rendue publique afin de signaler aux autres utilisateurs qu’Alice a certifié la validité de la clé de Bob



1. **Certifier une clé publique PGP**

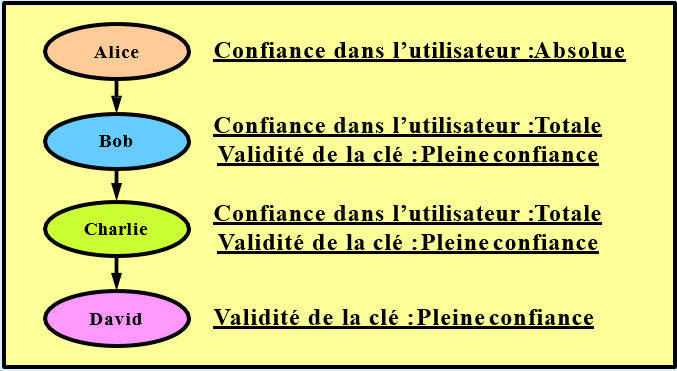
* Enigmail/Gestiondes clés, Effectuez un clic droit sur la clé publique de votre interlocuteur
* Signer la clé
* Choisissez l’option « J’ai vérifié minutieusement »
* Votre phrase de passe PGP est demandée pour créer cette signature au moyen de votre clé privée
* Vérifiez que la validité de la clé de votre interlocuteur devient « fiable » dans Enigmail (pleine confiance)

1. **Visualiser les certificats des clés publiques**

* Enigmail / Gestion des clés, La certification de la clé que vous venez de créer est enregistrée dans votre trousseau de clés publiques
* Effectuez un double-clic sur la clé publique de votre interlocuteur
* Ouvrez l’onglet « Certifications »
* Visualisez les certificats associés : sa clé publique doit contenir des auto-signatures (les clés sont auto-signées au moment de leur création) ainsi que votre propre signature

1. **Utiliser une chaîne de confiance PGP**

* **Il est possible d’utiliser une chaîne de confiance lorsqu’on ne connaît pas le propriétaire d’une clé (web of trust)**



1. **Révoquer des clés PGP**
2. **Révoquer sa clé publique PGP**

* **La révocation des clés publiques**
* Il est parfois nécessaire de révoquer sa clé publique (pourquoi ?)
* Pour cela, on doit créer au préalable un certificat de révocation
* Le certificat de révocation est signé avec la clé privée correspondante (afin de prouver qu’on est bien le titulaire de la clé)
* Le certificat de révocation est sauvegardé dans un fichier texte qui doit être gardé confidentiel (pourquoi ?)
* Il est conseillé de générer le certificat de révocation au moment même où on crée sa paire de clés (pourquoi ?)
* S’il devient nécessaire de révoquer sa clé publique, on récupère le fichier contenant le certificat de révocation et on l’importe dans son trousseau de clés
* Il faut ensuite envoyer sa clé révoquée à ses interlocuteurs ou aux serveurs de clés afin de les informer de la révocation de sa clé

1. **Révoquer sa clé publique PGP**

* **Créer un certificat de révocation**
* Enigmail / Gestion des clés, Sélectionnez votre clé publique
* Générer / Certification de révocation
* Votre phrase de passe PGP est demandée afin de signer le certificat de révocation avec votre clé privée
* Enregistrez le certificat de révocation dans un fichier texte de nom « cert-revocation.asc »
* **Importer le certificat de révocation dans son trousseau de clés**
* Enigmail / Gestion des clés /Fichier / Importer des clés d’un fichier
* Sélectionnez le fichier texte « cert-revocation.asc» contenant votre certificat de révocation (Si le certificat de révocation ne veut pas s’importer, on peut utiliser la commande **$gpg2 --import <certificat**> )
* Vérifiez que la validité de votre clé devient « révoquée »
* Si besoin, pour installer gpg2 utiliser la commande **$sudo apt install gnupg2**
* **Exporter sa clé révoquée vers le serveur de clés**
* Enigmail / Gestion des clés, Exportez votre clé révoquée vers le serveur de clés
* Vérifiez que votre clé est effectivement signalée comme révoquée sur le serveur

1. **Prendre en compte la révocation des clés PGP**

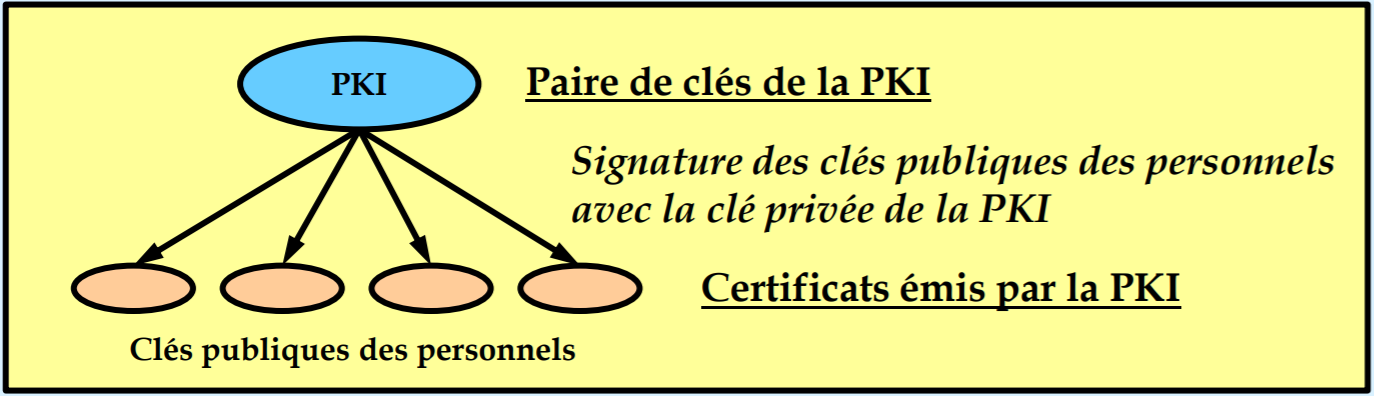
* **Mettre à jour les clés révoquées depuis le serveur de clés**
* Enigmail / Gestion des clés, Mettez à jour l’ensemble des clés publiques de votre trousseau de clés depuis le serveur de clés : Serveur de clés / Actualiser toutes les clés publiques
* Vérifiez que la validité de toutes les clés devient « révoquée »

1. **Créer des certificats S/MIME**
2. **Certification des clés S/MIME**

* **La certification des clés S/MIME**
* Les certificats S/MIME sont fournis par une autorité de certification (AC), ils peuvent être payants ou gratuits (pour une durée limitée)
* L’association entre la clé et l’adresse mail est validée par l'AC : certificats de classe 1 (simple vérification de l'adresse mail) ou de classe 2 (vérification plus poussée de l'identité)
* Le format des certificats utilisés est X.509 (comme les certificats SSL/TLS pour HTTPS)
* **S/MIME est surtout utilisé en entreprise**
* On peut utiliser dans ce cas une PKI (infrastructure à clés publiques) interne à l’entreprise

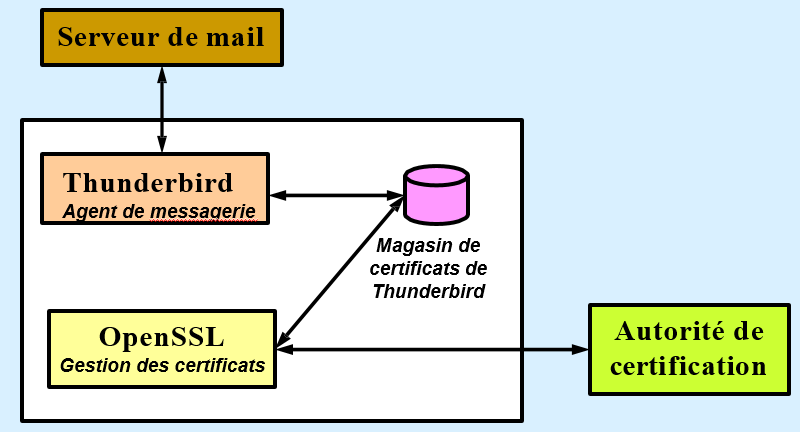
1. **Fonctions d’une PKI**

* **Générer les paires de clés des personnels de l’entreprise**
* **Certifier les clés publiques des personnels (fonction d’AC)**
* La PKI dispose de sa propre paire de clés
* Les clés publiques des personnels sont signées avec la clé privée de la PKI



* **Stocker les paires de clés et les certificats émis (fonction de serveur de clés)**
* Les clés doivent être conservées même après leur expiration (pourquoi ?)
* **Déployer les clés et les certificats vers les postes clients**
* Par exemple avec Active Directory
* **Surveiller les dates d’expiration et régénérer les clés et les certificats au moment de leur expiration**

1. **Logiciels S/MIME utilisés dans le TP**



1. **Créer sa paire de clés S/MIME**

* Aller au terminal
* Créez une clé privée RSA de 3072 bits pour votre certificat S/MIME : openssl genrsa -aes256 –out smime\_prenom\_nom.key 3072
* Vous devrez entrer un mot de passe protégeant votre clé privée
* Vérifiez qu’un fichier smime\_prenom\_nom.key a été créé

1. **Afficher les composants de sa clé privée**

* Aller au terminal
* Affichez les composants de votre clé privée RSA : openssl rsa -text -in smime\_prenom\_nom.key -noout
* Vous devrez entrer votre mot de passe pour déverrouiller votre clé privée

1. **Créer un CSR (Certificat Signing Request)**

* Pour obtenir un certificat S/MIME de la part de l’AC, il faut au préalable créer un CSR
* Créez un CSR pour votre clé privée : openssl req -new -key smime\_prenom\_nom.key -out smime\_prenom\_nom.csr
* Vous devrez entrer votre mot de passe pour déverrouiller votre clé privée
* Le certificat sera associé à un DN (Distinguished Name)
* Entrez les éléments suivants pour votre DN :

Country Name : MA

Organization Name : Societe.com

Common Name : Prénom Nom

Email Address : [prenom.nom@hotmail.com](mailto:prenom.nom@hotmail.com)

Autres champs : entrer un caractère point « . » pour indiquer l’absence de valeur

* Vérifiez qu’un fichier smime\_prenom\_nom.csr a été créé

1. **Afficher le CSR**

* Aller au terminal
* Affichez le CSR que vous venez de créer : openssl req -text -in smime\_prenom\_nom.csr -noout
* Le CSR contient votre DN (Subject), les composants de votre clé publique RSA (Modulus et Exponent), ainsi qu’une signature de ces éléments avec votre clé privée RSA

1. **Récupérer les fichiers de l’AC**

* La prochaine étape est de transmettre votre CSR à l’AC afin que celle-ci génère votre certificat
* Dans ce TP, vous allez générer vous-même votre certificat en jouant le rôle de l’AC
* Récupérez les fichiers nécessaires à la génération du certificat selon les indications données pendant le TP : ca.key : clé privée de l'AC ca.crt : certificat de l'AC

1. **Créer son certificat S/MIME**

* Ouvrir le terminal
* Créez votre certificat signé par l'AC à partir du CSR : openssl x509 -req -days 365 -in smime\_prenom\_nom.csr -CA ca.crt -CAkey ca.key -set\_serial numéro -out smime\_prenom\_nom.crt
* Chaque étudiant doit utiliser un numéro de série différent (code étudiant)
* Ce certificat sera valable pendant un an
* Vous devrez entrer le mot de passe de la CA qui vous sera communiqué pendant le TP
* Vérifiez qu’un fichier smime\_prenom\_nom.crt a été créé

1. **Afficher le certificat S/MIME**

* Ouvrir le terminal
* Affichez le certificat S/MIME que vous venez de créer : openssl x509 -text -in smime\_prenom\_nom.crt -noout
* Le certificat contient le numéro de série, le DN de l’AC (Issuer), l’intervalle de validité, votre propre DN (Subject), les composants de votre clé publique RSA (Modulus et Exponent), ainsi qu’une signature de ces éléments avec la clé privée de l’AC

1. **Format PKCS #12**

* PKCS (Public-Key Cryptography Standards) est une famille de standards de la société RSA
* Le format PKCS #12 est un conteneur permettant de stocker plusieurs éléments cryptographiques
* Il est souvent utilisé pour stocker la clé privée et le certificat associé (ou une chaîne de certificats)
* Il peut être protégé par un mot de passe
* Il permet ensuite d’importer la clé privée et le certificat dans un navigateur web ou dans un agent de mail

1. **Exporter clé privée et certificat en PKCS #12**

* Ouvrir le terminal
* Exporter votre clé privée et votre certificat S/MIME au format PKCS #12 : openssl pkcs12 –export –in smime\_prenom\_nom.crt -inkey smime\_prenom\_nom.key -out smime\_prenom\_nom.p12
* Vous devrez entrer votre mot de passe pour déverrouiller votre clé privée, puis un nouveau mot de passe qui servira à protéger le fichier PKCS #12
* Vérifiez qu’un fichier smime\_prenom\_nom.p12 a été créé

1. **Envoyer des mails sécurisés par S/MIME**
2. **Importer clés et certificats dans le magasin**

* Importez votre clé privée et votre certificat S/MIME au format PKCS #12 dans le magasin de certificats de Thunderbird : Édition / Préférences, Onglet « Avancé », bouton « Gérer les certificats » Onglet « Vos certificats », bouton « Importer… »
* Sélectionner le fichier PKCS #12 créé précédemment
* Vous devrez entrer le mot de passe du fichier PKCS #12
* Effectuez un double-clic sur votre certificat et vérifiez que sa validité ne peut pas être établie
* Importez le certificat de l’AC dans le magasin de certificats de Thunderbird : Onglet « Autorités », bouton « Importer… »
* Sélectionner le fichier ca.crt que vous avez récupéré précédemment
* Cochez l’option « Confirmer cette AC pour identifier les utilisateurs de courrier »
* Vérifier que le certificat de l’AC apparaît dans le magasin et que votre propre certificat est maintenant validé

**NB** :

1. Au cas où le certificat ne s’importe pas, il se peut qu’il y’a des problèmes de permission🡪 il ne faut pas exécuter les commandes openssl en tant que root (sudo)
2. Il faut bien faire attention aux différents mots de passes créer lors de la génération des clés et certificats
3. **Configurer Thunderbird**

* Aller dans thunderbird
* Désactivez l’utilisation d’OpenPGP : Édition / Paramètres des comptes Onglet « Sécurité OpenPGP » Décochez la prise en charge d’OpenPGP
* Configurez l'utilisation de S/MIME :

Onglet « Sécurité »

Signature : bouton « Sélectionner un certificat… »

Sélectionnez votre certificat S/MIME

Acceptez le même certificat S/MIME pour le chiffrement

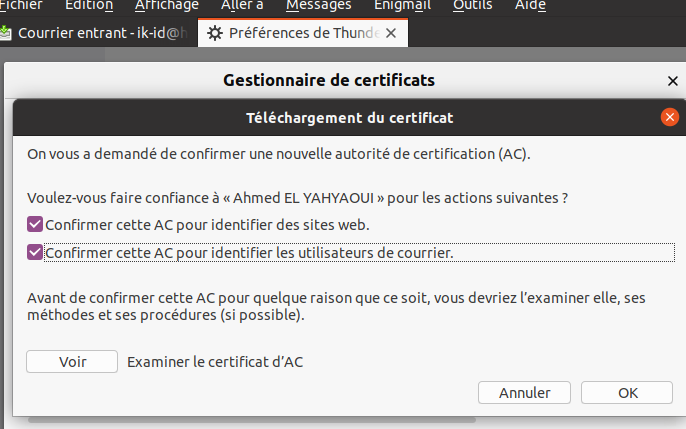
1. **S’envoyer un mail signé et/ou chiffré**

* Envoyez à vous-même un mail signé puis un mail chiffré
* Lisez les mails reçus et cliquez sur les logos en forme d’enveloppe ou de verrou à droite pour afficher les informations de sécurité de S/MIME

NB : Au cas où vous aurez le message d’erreur suivant :

« L’envoi du message a échoué. Vous avez demandé la signature numérique de ce message, mais l’application n’a pas trouvé le certificat spécifié dans les préférences, ou alors le certificat a expiré »

Vous devez supprimer le certificat de l’autorité racine (ca.crt dans notre cas) et l’important à nouveau en vérifiant bien que les options comme dans la figure ci-dessous :



1. **Envoyer un mail signé et/ou chiffré**

* Envoyez à votre interlocuteur un mail signé
* Lisez le mail signé reçu de votre interlocuteur
* Le certificat S/MIME de l’émetteur du mail est inclus automatiquement dans les mails signés
* Vérifiez que le certificat de votre interlocuteur apparaît dans votre magasin de certificats, sous l’onglet « Personnes »
* Maintenant que vous possédez le certificat de votre interlocuteur, envoyez-lui un mail chiffré

1. **La révocation des certificats S/MIME**

* **Il existe principalement deux manières de gérer la révocation des certificats**
* Les CRL (Certificate Revocation Lists) : l’AC gère une liste de certificats révoqués, le client mail interroge régulièrement l’AC afin d’obtenir la liste à jour des certificats révoqués
* Les serveurs OCSP (Online Certificate Status Protocol) : avant d’utiliser un certificat, le client mail interroge le serveur OCSP de l’AC pour savoir si ce certificat a été révoqué
* Les URL des CRL et des serveurs OCSP sont indiquées dans les certificats délivrés par l’AC

1. **Envoyer le rapport du TP**

* *Pour ce point, je vais vous envoyer les informations nécessaires par mail après la première science du TP3.*