# Introduction au chiffrement symétrique et asymétrique

# Table des matières

# 1. Objectifs

- Découvrir le chiffrement symétrique ;
- Découvrir le chiffrement asymétrique ;
- Connaître des technologies utilisant un chiffrement asymétrique.

# 2. Contexte

**Durée** : 2h

**Environnement de travail** : Linux en ligne de commande

**Pré-requis** : Aucun

Le chiffrement est une technique qui permet de garantir la confidentialité et l'origine des informations électroniques échangées ou stockées.

Pour garantir la confidentialité d'une information on applique un code qui rend le message incompréhensible tant que l'on ne connaît pas la manière de décoder. On appelle clé l'information qui permet de coder et de décoder.

Le chiffrement est aussi une façon de signer des informations afin de garantir qu'on en est l'émetteur.

L'usage du chiffrement s'est fortement développé ces dernières années notamment sur le Web avec la diffusion du protocole HTTPS. La multiplication des fuites de données et le développement de la cybercriminalité en sont une première cause.

Les révélations d'Edward Snowden en 2013 ont également montré que les états et les grandes entreprises espionnaient les échanges sur Internet. Le chiffrement est donc également un outil pour préserver l'intimité de chacun en ligne.

# 3. Découverte du chiffrement

### Objectifs

- Découvrir la notion de chiffrement ;
- Connaître un algorithme basique de chiffrement.

### Mise en situation

Lorsqu'on échange des messages sur Internet, c'est un peu comme si on communiquait avec des cartes postales. C'est à dire qu'il est très facile pour un intermédiaire de les lire. Les messages ne sont pas confidentiels.

La seule façon de communiquer de façon confidentielle est de **chiffrer** les messages. Cela consiste à définir un code secret que seuls les deux correspondants connaissent. Ainsi les messages sont toujours lisibles par des tiers, mais il ne sont plus en mesure de comprendre quoi que ce soit. C'est comme si sur ma carte postale j'écrivais : DZMIY Z YPO. On peut toujours la lire, mais sans le code il est difficile de comprendre le message. Si je vous donne le code, alors vous serez en mesure de le **déchiffrer** c'est à dire de l'appliquer pour retrouver le message original. Mais comme un code de chiffrement doit rester secret, je ne le donne pas dans une vidéo.

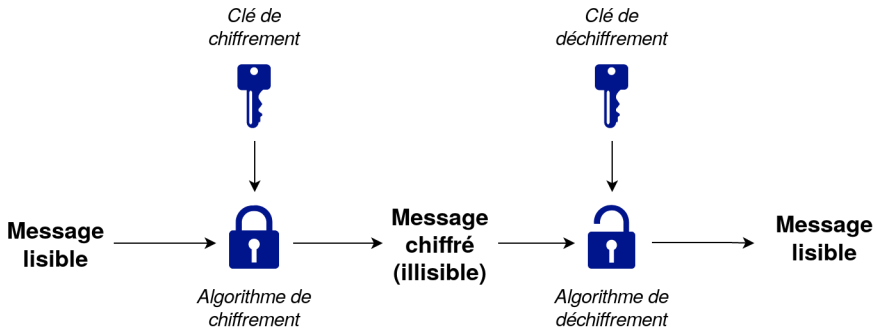
Le code que j'ai utilisé ici est très simple, il sera facilement décrypté. **Décrypter** un code signifie parvenir à en trouver la teneur originale sans en avoir été informé. Je vous laisse décrypter mon message.

#### Processus de transmission d'un message chiffré

Fondamental

La transmission d'un message chiffré se fera en trois étapes :

1. Chiffrement du message à l'aide d'un algorithme de chiffrement auquel on fournit une clé de chiffrement.
2. Transmission du message chiffré.
3. Déchiffrement à l'aide d'un algorithme de chiffrement auquel on fournit une clé de déchiffrement.




Fondamental

Le chiffrement est un procédé cryptographique permettant de coder un message de telle façon que sa lecture ne soit possible que par le seul possesseur de la clé de déchiffrement

Remarque

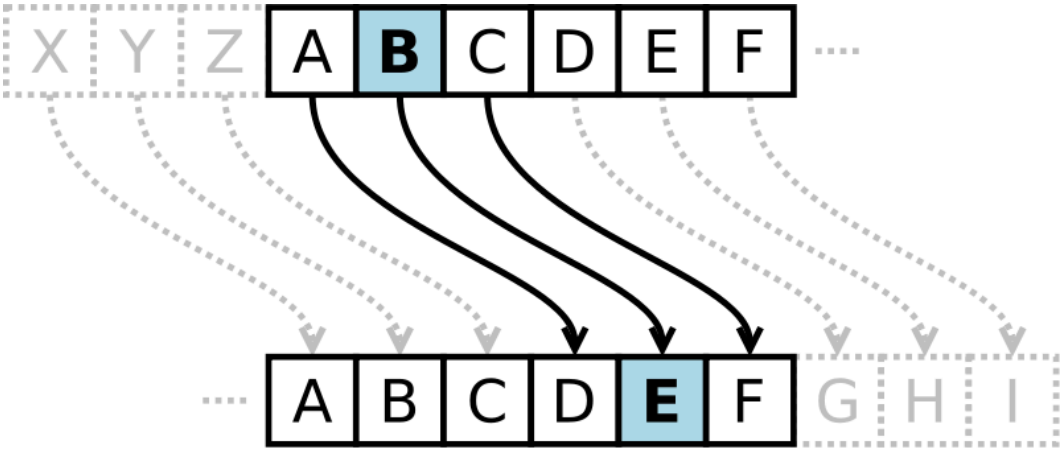
- les **algorithmes de chiffrement/déchiffrement** : ces algorithmes sont le plus souvent disponibles librement.
- les **clés** : elles sont un ensemble de paramètres à fournir à l'algorithme afin qu'il puisse réaliser sa tâche. Pour garantir la confidentialité du message, la clé de déchiffrement doit être privée.
- l'**unicité de la paire de clés** : pour une clé de chiffrement il n'existe qu'une clé permettant de déchiffrer et l'inverse est également vrai (aucun intermédiaire ne peut deviner cette clé).

### Le chiffre de César

Exemple

Aussi appelé chiffrement par décalage, ce chiffrement très simple consiste à décaler toutes les lettres d'un message. Dans ce cas, les clés de chiffrement et de déchiffrement sont identiques et correspondent à l'amplitude du décalage.

Par exemple pour un décalage de 2, les « A » deviendront des « C », les « B » deviendront des « D », etc. Le mot « shannon » donnera « vkdqqrq » avec un décalage de 3.




*Chiffre de César*

### Chiffrer, pas crypter

Complément

La mot crypter n'existe pas en français. Un message est donc chiffré mais pas crypté.

Néanmoins, on peut parler de « décryptage » lorsque l'on cherche à déchiffrer un message sans disposer de la clé nécessaire.

### Sécurité du chiffrement

Complément

Le problème du chiffrement par décalage est son manque de sécurité. Il est très simple de trouver la clé de déchiffrement par essais successifs.

La sécurité de ces algorithmes doit reposer sur des propriétés mathématiques fortes et/ou sur une bonne transformation de l'information.

Baser la sécurisé du chiffrement sur le fait que son procédé soit inconnu par un tiers revient à faire de la **sécurité par l'obscurité**. Cacher un procédé protège beaucoup moins que d'utiliser un procédé de chiffrement public et solide.

### Hachage cryptographique et condensat

⊕ Complément

Le **hachage cryptographique** est le second procédé cryptographique très utilisé. Il permet de produire des **condensats** (aussi appelé empreinte numérique). Un condensat est une chaîne de caractères **unique** de taille fixe pour chaque donnée pouvant exister. Ainsi, deux messages différents (même d'un caractère) auront un condensat différent.

Il n'est pas possible d'inverser le processus et de passer d'un condensat au message d'origine (ceci l'oppose au chiffrement). Les algorithmes classiques sont : SHA256, SHA1, MD5, etc.

## À retenir

- Le chiffrement permet de transformer un message afin qu'il ne soit lisible que par une personne possédant la clé de déchiffrement.
- La sécurité d'un chiffrement repose sur de bonnes propriétés mathématiques et sur de bonnes opérations de transformation de l'information.

# 4. Appliquer la notion

## Question

[solution n°1 p. 31]

Décrypter le message suivant, chiffré par décalage :

```
nc ocejkpg gpkioc pgvckv rcu ugewtkugg
```

On ne connaît pas la clé, mais on sait que le premier mot est **la**.

### Indice :

On peut calculer le décalage à partir de notre connaissance du premier mot : il correspond à la différence de position entre les lettres **n** et **l**, ou **c** et **a**.

# 5. Chiffrement symétrique

### Objectifs

- Découvrir le chiffrement symétrique ;
- Utiliser un algorithme de chiffrement symétrique.

### Mise en situation

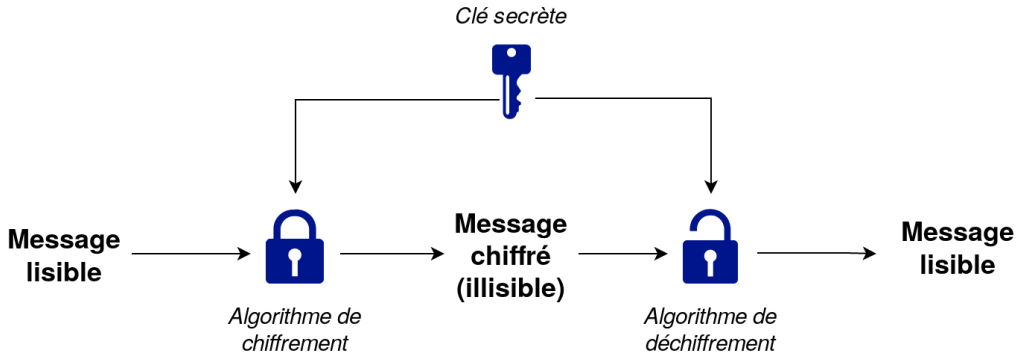
Lorsque deux personnes qui communiquent entre elles partagent exactement la même technique de chiffrement, on parle de chiffrement symétrique. C'est en général une méthode insuffisante pour les communications entre plusieurs personnes. Il est possible d'utiliser une clé symétrique pour communiquer avec quelqu'un, mais dans ce cas la clé sera changée à chaque nouvelle communication.

Le chiffrement symétrique est aussi utilisé lorsque l'on souhaite chiffrer des données sans les partager. C'est le cas lorsque l'on chiffre son disque dur pour que, même en cas de vol, seul le propriétaire puisse accéder aux données. Ainsi, le but n'est plus de sécuriser une communication mais le stockage lui-même.

#### Le chiffrement symétrique

Az Définition

Le chiffrement symétrique est un chiffrement dans lequel la clé de chiffrement sert également à déchiffrer. On parle alors de clé secrète.




#### Schéma d'utilisation classique

Exemple

Bob souhaite chiffrer son disque dur pour qu'il soit le seul à pouvoir accéder à ses fichiers.

1. Lors de l'installation de son système d'exploitation Bob décide de chiffrer son disque dur, il choisit un mot de passe P.
2. La totalité du disque est chiffré avec une clé K générée par le système, cette clé est stockée sur le disque dur.
3. La clé K est à son tour chiffrée grâce au mot de passe P (elle ne doit pas être accessible en clair sur le disque).
4. À chaque déverrouillage de son ordinateur Bob entre le mot de passe qui permet de déchiffrer la clé K ; elle est alors chargée en mémoire afin d'être disponible rapidement.
5. À chaque fois qu'un fichier est accédé en lecture il est déchiffré avec K ; à chaque accès en écriture il est chiffré avec K.

Ainsi, Bob est certain que tant que son mot de passe reste secret, ses données sont sécurisées même si quelqu'un accède au disque de son ordinateur.

### Le partage de clés

⚠ Attention

Il arrive parfois que la clé soit connue par plusieurs personnes ou soit présente sur plusieurs serveurs du propriétaire.

Le transfert de la clé doit absolument être sécurisé pour que le chiffrement ne soit pas compromis.

Plusieurs stratégies existent et une des plus efficaces est d'utiliser un autre type de chiffrement pour chiffrer la clé secrète et d'envoyer ce message au destinataire qui pourra récupérer la clé secrète en toute sécurité. Transférer la clé au travers d'une connexion SSH est une stratégie commune.

### Le chiffrement AES

Az Définition

L'*Advanced Encryption System* est un standard très répandu pour réaliser du chiffrement symétrique. Il possède énormément de bonnes propriétés : facile à calculer, implémentation possible au niveau logiciel comme au niveau matériel (implémentation câblée). Ce type de chiffrement est utilisé notamment pour des protocoles tels que SSL (sécurisant les connexions HTTP) ou encore pour chiffrer son disque dur (VeraCrypt[1]).

### Générer sa propre clé secrète

Méthode

Voici une commande basique pour générer une clé secrète. Il existe des implémentations plus complexes et sécurisées. Ici, la clé est simplement une suite aléatoire de 32 octets. Pour une utilisation réelle, il est conseillé d'utiliser des implémentations robustes et de confiance pour générer sa clé secrète.

```
1 openssl rand 32 > cle_secrete.pem
```

On obtient ici la clé de chiffrement dans le fichier `cle_secrete.pem`.

### Utiliser une phrase secrète ou un mot de passe

Conseil

Il est possible de sécuriser sa clé secrète en spécifiant un mot de passe lors de la génération de la clé. Une telle pratique permet de conserver la sécurité même si un attaquant réussit à récupérer la clé. Il est tout de même fortement conseillé de générer une nouvelle clé dès lors qu'une clé est compromise.

1. https://fr.wikipedia.org/wiki/VeraCrypt

### Chiffrer et déchiffrer

Méthode

On peut utiliser les commandes dans un repl Bash ou un terminal, pour chiffrer et déchiffrer un message.

Pour chiffrer un fichier `msg.txt` en un nouveau fichier `msg.txt.enc` avec AES et une clé secrète générée `cle_secrete.pem` on utilise :

```
openssl aes-256-cbc -pbkdf2 -iter 100000 -in msg.txt -out msg.txt.enc -pass
file:cle_secrete.pem
```

Pour déchiffrer un fichier `msg.txt.enc` chiffré avec AES en un fichier `msg.txt` et une clé secrète générée `cle_secrete.pem` on utilise :

```
openssl aes-256-cbc -pbkdf2 -iter 100000 -d -in msg.txt.enc -out msg.txt -pass
file:cle_secrete.pem
```

## À retenir

- Le chiffrement symétrique permet de sécuriser ses propres données avec une unique clé.
- Le chiffrement symétrique permet d'échanger des données avec des pairs s'ils disposent eux aussi de la clé.
- Le partage de clé est très complexe et doit rester exceptionnel pour conserver sa confidentialité.
- L'algorithme AES permet de réaliser un chiffrement symétrique.

# 6. Appliquer la notion

## Question

[solution n°2 p. 31]

Le fichier `aes.priv` contient une clé AES utilisée pour le chiffrement de fichiers.

*Enregistrer le fichier.*
(cf. aes.priv)

Le fichier `secret_data.enc` a été chiffré avec la clé `aes.priv`.

*Enregistrer le fichier.*
(cf. secret_data.enc)

À l'aide d'un terminal ou d'un Repl Bash retrouver le contenu du message grâce à la commande `openssl`.

### Indice :

Le message a été chiffré OpenSSL. Le paramètre `iter` vaut 10000.

### Indice :

*Chiffrement symétrique* *(cf. p.9)*

# 7. Chiffrement asymétrique

### Objectifs

- Découvrir le chiffrement asymétrique ;
- Utiliser un algorithme de chiffrement asymétrique.

### Mise en situation

L'inconvénient du chiffrement symétrique est que si trois personnes souhaitent communiquer entre elles deux par deux, et bien le troisième pourra toujours espionner les deux autres, puisque le code est commun. On pourrait imaginer que chaque couple de personnes possède une clé spécifique, mais si 1000 personnes échangent entre elles, cela fera pour chacune, 999 clés à gérer. On voit bien que cela n'est pas satisfaisant pour les communications sur Internet.

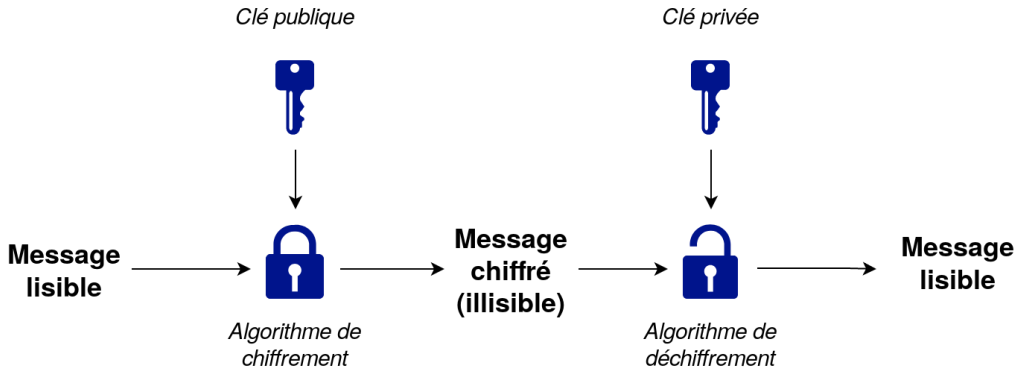
La solution est fournie par le chiffrement asymétrique. Celui-ci est basé sur les propriétés mathématiques d'une paire de clés : la première est publique, elle est communiquée au monde entier et elle sert à chiffrer les messages. La seconde est privée, elle n'est communiquée à personne et elle sert à déchiffrer les messages. Ainsi si quelqu'un souhaite m'envoyer un message, il peut le chiffrer avec ma clé publique, seul moi pourrai le lire car je suis le seul à disposer de la clé privée.

On peut voir cela comme des milliers de boites inviolables que je diffuserais dans le monde entier. N'importe qui peut prendre une de mes boîtes et glisser un message dedans. Je suis le seul à en posséder la clé, donc seul moi pourrai accéder au message.

#### Le chiffrement asymétrique

Az Définition

Le **chiffrement asymétrique** vient concrétiser la différence entre la clé de chiffrement et de déchiffrement. En pratique, la clé de chiffrement sera nommée **clé publique** car elle sera librement communiquée. La clé de déchiffrement sera nommée **clé privée** car elle ne doit être communiquée sous aucun prétexte.




Fondamental

- N'importe qui pourra utiliser la clé publique d'une personne pour lui envoyer un message (cette clé publique est connue de tous).
- Ce message ne sera déchiffrable qu'avec la clé privée de cette même personne (qui n'est détenue que par elle-même).

### Schéma d'utilisation classique

Exemple

Bob veut envoyer un message à Alice :

1. La clé publique d'Alice est disponible sur un site web.
2. Bob la récupère et chiffre son message grâce à cette clé.
3. Bob envoie le message chiffré à Alice.
4. Alice reçoit puis déchiffre le message grâce à sa clé privée.

Remarque

A l'issue de la communication, Alice et Bob sont certains que leur communication a été confidentielle... à moins qu'Alice n'ait pas correctement protégé la confidentialité de sa clé privée (à cause d'une erreur ou d'une attaque).

### Chiffrement RSA

Exemple

Le chiffrement RSA est l'un des algorithmes les plus connus lorsque l'on parle de chiffrement asymétrique. Il est utilisé dans de nombreux contextes notamment :

- Par le protocole SSH, qui permet d'accéder à un serveur distant de manière sécurisée.
- Pour transmettre une clé de chiffrement symétrique. La clé de chiffrement symétrique doit rester confidentielle : elle est transmise à l'aide d'un chiffrement asymétrique lorsqu'elle doit être partagée à un tiers autorisé.

### Générer ses propres clés

Méthode

Il existe plusieurs manières de générer des clés RSA. La plus simple est de générer des clés SSH qui sont déjà des clés RSA. Le plus souvent la paire de clés sont nommées de la manière suivante : `nom_clé.pub` pour la publique et `nom_clé` pour la privée. Il est commun de posséder plusieurs paires de clés. On préfère garder une nomenclature similaire pour le nom des clés pour ne pas les mélanger. Ces clés pourront être directement utilisées *pour accéder à un serveur distant.*

```
ssh-keygen
```

Pour la suite, on génère les clés RSA directement avec `openssl`.

```
# Génère une clé privée protégée par mot de passe dans le fichier private.pem
openssl genrsa -des3 -out private.pem 2048
# Extrait la clé publique de la clé privée dans le fichier public.pem
openssl rsa -in private.pem -outform PEM -pubout -out public.pem
```

### Utiliser une phrase secrète ou un mot de passe

Conseil

Il est possible de sécuriser sa clé privée en spécifiant un mot de passe lors de la génération de la clé.

Ainsi, la clé privée est elle-même chiffrée par chiffrement symétrique.

Une telle pratique permet de conserver la sécurité même si un attaquant réussit à récupérer la clé privée. Il est tout de même fortement conseillé de générer une nouvelle clé si sa clé actuelle est compromise.

### Paire de clés RSA

Exemple

Voici à quoi ressemble une clef privée RSA, protégée par mot de passe (**il ne faut jamais publier une telle clé si elle est en service).**

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,EB9CF5A7B25026DE

/nJK7OYp0kT26M3bDlTcO6RVAcoIBpZ8oA18p78Nl0mWOQPE0oZTaDJbTuEQLNNW
UWL7c+D7unGg+UdO7DWc52Z4M8JgoYhy7hvZ1BzFs4bUxJ+OP0sWWCyE9eWScK+4
vXXrEhp6JhUSx654cbyPtUYwVGdxKoq25TzEDvA5eSQU1MH/LviNErTYLtZcbPjw
vrbeOivbcSjcTzB83cdgeM/y+ourduIqWjwM1BkhgwcZ5gN6jdWPFMKGe9iF2gsy
IRvfAAjbzmJVM2DXEjHaSSRkXMIusdJN1lyEuH833XGBql9RmslQ9BzjUuYr1Fzd
A1n9EBvaY3pT8UPa3epFwO9kT7ag2Hrx7jYNtlUk/7kyPnm95cJWUvBvGrZxhCDD
4+lsCWrIRSslI5RnlgZQY/FgPOT2blesgkk2Ize/10PflM6w2fPfSsSMKYE9dEY8
zc3F+WwslxopQkWmRYZ8wwzv3Tng60/3DIU4X2oUPPajTFEjIk/KZ5tiHf6bSTcK
VQtE6w/bIHOyu/4ge7EfBDTZAuLGKqtfMuT8Dcz5behK/TgcLEq+4ps2KUu4f/rE
aGcI9sGSYmrRn18bRPATb+VkfKH8ak48zj43UBfrXhTRUlCx7FMDwfu9Csn5aEDl
dOANQ0sz51E6h3DDGHJU8Z0o5mQtFvcPyw/s2uc5ooTlfQr0e+36EQHSqfJIEVzO
Y1Cm2jq+bTPuI4yAp1rlEng3Z+jS3FIkP9lP8Iz7tyhvYWhGc0bK0gebGXO2uFOm
6X6eObQQkQ+sbdLQ7wCe3J4UmOEkq6wmvBlt5ZwH+Eo2icKC++434olMxCBKHAPH
2uXwNbZORhqP77Pl7dNge8bhZtdAqWG0Mb8DiGHYweSgC6m8h3xYAtQDdnWyDY4D
0qpLjI+mw+EVk8aMoNVTO3sDWuArsJVYOpoIfQ7/tnBhmJ1Rr9Axr8woDMA6Z501
7zvw4g4zYID61v5hl+xvAO+BUHwGIwuJ92gnSm4WOPk63GLiu0ita73IJgdL6FSG
bsrcmPOLa0+ATCU6CMRLJ3mzefJOviwThkC3P88Fy8kjSg3MTgnXASqqfEix8+Y2
eLtXJmWQXfhWdT24+f4EUEU5+YIy50qL8VGNdRgqKgclC6nUM1uW5txTEqs+fTjH
NhTr5s7gPMWbQYwPkcdNhPaxNhrykTFEfmKL6AEzx64sGn3/kzuqHv0/cH2jR3OR
sp6nd4iarbPYk9zKbUbiBMNvpjegbJ7WLcEllWHMApwfNTufJp0MHIk8AymjbZth
oTDvFZ6BThSW3kJMz31gBX18PswYBUs3bB7FTC3fXDxoSjHk3ClW4aclGIdbWAba
BJNTSieLL6TTlQ3cAD2bb3xJDnAuHScZ+tauu83YDTGfiUsQ4ew9alqypI+dbrPc
utSu9PVOb/8F+MZENW+DtLJ0IvZgcTj0diOqL5Cd1S4J+v/0s60XjyuTBDl1Yj8U
DID6IHFDm/qIfFcNfNzxTgag+nOwXFnn/d5S38YGHiokgXNCugeYw/HeekBP69tm
VjCq2YzIXnOepfYron5KaMt+4y1leXdJzNPHU+7FAr0w6k/IC4r+wg==
-----END RSA PRIVATE KEY-----
```

Voici une clef RSA publique : c'est celle que l'on partage avec des tiers pour des communications.

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAv/Lg6dBGCZKsuIHv/eJi
B2izeWybrH9HjVEwpMBfpS8Nh02hXNPaw547HXnsBC47yTCrNoDDKPBgIIJ0mLNE
R9XhiryK3ni7yo/lE3qE2YKE905a4ALrjejk9J+afnSAfmcfGkfLYl0n2mKo+hy7
xok4/sDLtv938Bbsyg/V1o+YeSucHIXe+b7S6dKR3fOwHw/TB3R+1Qu/nl/RIv7s
OnkRwBxs/NvO6Na+Xny3f0PxpWjjL+CCTePRYaZ/VrNI/Uym3vGGhLzDrlDPu8IB
0S8oF9xgQSHxUrHWNKJ6k0+Fjr5BzoUnrqy6ipC5A1vEEyYRc95ZyKI0vkAYRihI
ewIDAQAB
-----END PUBLIC KEY-----
```

### Chiffrer et déchiffrer

Méthode

Pour chiffrer et déchiffrer un message, il est nécessaire d'utiliser les commandes ci-dessous.

Pour chiffrer un fichier :

```
openssl rsautl -encrypt -pubin -inkey public.pem -in msg.txt -out msg.enc
```

Pour chiffrer une chaîne de caractères :

```
echo "Message confidentiel" | openssl rsautl -encrypt -pubin -inkey public.pem -
out msg.enc
```

Pour déchiffrer un message :

```
openssl rsautl -decrypt -inkey private.pem -in msg.enc -out message_en_clair.txt
```

### Chiffrer avec la clé privée pour signer

Complément

Il est possible d'inverser le rôle des clés en chiffrant avec la clé privée et en déchiffrant avec la clé publique. Le but d'une telle pratique n'est pas de garder le message secret mais de vérifier l'identité de l'expéditeur. Seul le propriétaire de la clé privée peut générer un message déchiffrable avec la clé publique, ce qui garantit son identité.

## À retenir

- Le chiffrement asymétrique permet un partage de la clé de chiffrement tout en garantissant la confidentialité de la clé de déchiffrement.
- La clé publique d'une personne est disponible par ceux qui veulent lui envoyer des messages.
- La clé privée d'une personne n'est jamais publiée et est utilisée par la personne pour déchiffrer les messages reçus.
- L'algorithme RSA permet de réaliser un chiffrement asymétrique.

# 8. Appliquer la notion

Les fichiers `public.pem` et `private.pem` contiennent respectivement des clés publique et privée RSA.

*Enregistrer le fichier `public.pem`.*
(cf. public.pem)

*Enregistrer le fichier `private.pem`.*
(cf. private.pem)

## Question

[solution n°3 p. 31]

Le fichier `msg.enc` ci-dessous a été chiffré avec la clé privée. Le mot de passe qui protège la clé privée est `test` (ne refaites pas ça chez vous !)

*Enregistrer le fichier `msg.enc`.*
(cf. msg.enc)

Trouver le contenu du message.

### Indice :

On utilisera une commande de la forme :

```
1 openssl rsautl -decrypt -inkey cle_privée -in message_chiffré.enc -out
  message_en_clair.txt
```

### Indice :

On peut utiliser `cat` pour afficher le contenu d'un fichier.

# 9. OpenPGP et GnuPG

### Objectifs

- Découvrir le standard OpenPGP ;
- Savoir utiliser GnuPG.

### Mise en situation

Il existe de nombreux algorithmes de chiffrement avec pour chacun des utilisations ciblées : échange d'informations et signatures typiquement. Pour faciliter les communications, il est nécessaire d'utiliser des standards assurant un bon niveau de sécurité, c'est à dire qui garantissent que le décryptage est difficile.

Parmi les nombreux standards existants pour les protocoles cryptographiques, l'un est devenu très répandu : OpenPGP.

Le logiciel GnuPG ou GPG (pour GNU Privacy Guard) est une implémentation libre du standard OpenPGP. Ce logiciel réputé très robuste est largement utilisé sur tous les systèmes d'exploitation actuels.

#### OpenPGP

Fondamental

OpenPGP est un format de cryptographie permettant de standardiser messages, chiffrement et signatures. Ce système cryptographique est dit *« hybride »* car il a recours à la fois à de la cryptographie symétrique et à de la cryptographie asymétrique.

À la manière des chiffrements uniquement asymétriques, l'OpenPGP demande l'utilisation d'une clé publique et d'une clé privée (qui sera protégée par une phrase de passe). Ainsi, le fonctionnement extérieur de l'OpenPGP ressemblera de très près aux protocoles asymétriques bien qu'il y ait des recours à de la cryptographie symétrique.

#### Origine du standard

Complément

Le nom de ce standard vient du logiciel PGP (Pretty Good Privacy) qui était l'outil historique d'échange sécurisé de données. Il a été développé au début des années 90 par P. Zimmermann. Aujourd'hui la société a été rachetée et de plus en plus de monde se tourne vers l'alternative libre Gnu Privacy Guard (aussi nommé GnuPG ou GPG).

### Utilisations classiques d'OpenPGP

Il y a trois utilisations très répandues d'OpenPGP :

- **Envoi sécurisé de documents ou de messages** : il permet de chiffrer des messages ou des fichiers quelconques.
- **Signature** : il peut produire des signatures permettant de vérifier l'identité de l'expéditeur. La signature est vérifiée par le destinataire grâce à la clé publique.
- **Vérification de documents** : il permet de vérifier que le document qui vient d'être téléchargé était bien celui prévu. Cette pratique est répandue pour le téléchargement de logiciels afin de s'assurer qu'un attaquant n'a pas altéré ou modifié le fichier prévu initialement.

### Générer ses propres clés GPG

Méthode

Pour générer une paire de clés avec GnuPG, il suffit de lancer la commande ci-dessous. Après avoir entré cette commande, le nom et l'adresse e-mail de la personne sont demandés. Il est également conseillé de *choisir un mot de passe très sécurisé.*

```
gpg --generate-key
```

Vous allez être redirigé sur un utilitaire qui va vous permettre de créer la paire de clef. Il faut saisir prénom nom et une adresse e-mail. Une phrase de passe pour la clé privée sera aussi demandée. Vous obtiendrez une sortie similaire à cela :

```
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: key 44E6F43C654279CF marked as ultimately trusted
gpg: directory '/home/john/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as
'/home/john/.gnupg/openpgp-
revocs.d/DA27A19AE74135DEB4BF30B144E6F43C654279CF.rev'
public and secret key created and signed.

pub   rsa4096 2020-04-28 [SC] [expires: 2020-04-29]
      DA27A19AE74135DEB4BF30B144E6F43C654279CF
      uid                      John Smith <john@smith.xyz>
      sub   rsa4096 2020-04-28 [E] [expires: 2020-04-29]
```

La paire de clé publique et privée ainsi qu'un certificat de révocation ont ici été générés.

### Exemple de clé publique

Exemple

Chaque clé publique a un identifiant (une version raccourcie de la clé) qui permettra de simplement rechercher des clés sur des serveurs de clés. Dans l'exemple précédent, il s'agit de :

```
DA27A19AE74135DEB4BF30B144E6F43C654279CF
```

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBF6n/dABEADLLFILp0manadsACxIeunoGIwLzEgaswnNcK7+6xOldCNBFjXT
tYLPBa685n6ekZfEribnhsswbdvypYif3nycNqun/vzALVS1FKA4yFqvBkBNmuWk
iSCGc+d7DKDOeIXEp1lh554vPn+RUHne8pLETrPfNq78P+vuem4U+JmNUHPJYrkn
8IYkOn5683Lk5BxJ6d0zDBZEtTsgt25CM3sKq9o4XoWcYqs4jy92nZejtdDkb1Ax
/tRKoMxgQfJXfSn0VaEA3/HXOXCeH35rZI32k/Q9rUz52GMb2hOomIMJz0B+Q0MI
nsCg196qgT/qlowsFe/UcME2iN4COuKCbokH4aTRWHri9ygCRYO3pv0rrTIr2aro
8abDU1vexHrbMAC1opwQnJElc0Lkl+K4qBRzY7DbBx5ceu259AFjT/unJVkzDaXG
hO0VMH7zC8IR8bYF2iis1F+4Hzb7f9yweKkJsg3ablzAeEdc2LDgkhy1tZ8BZpQI
uBNsDJCbeYmRqfCi0gRvJrG7o2U1xMhU3A7SyqZ59H2CG24NkY6d4i2PUi9wV4YK
fI+D0awnsBcT2rYjZ69bFX8rxVFGrNxwuCD8LKJkK7bJXAhP8Y7g0dhfmP4qoXJZ
4skSF0KObtmjgj2YIJgj9RxNKnkTvLy6T66lS7pAASKBx1zeQlrljS8OcQARAQAB
tBtKb2huIFNtaXRoIDxqb2huQHNtaXRoLnh5ej6JAlQEEwEIAD4WIQTaJ6Ga50E1
3rS/MLFE5vQ8ZUJ5zwUCXqf90AIbAwUJAAFRgAULCQgHAgYVCgkICwIEFgIDAQIe
AQIXgAAKCRBE5vQ8ZUJ5zyOlD/99QV2RR98qMqkFgRLsn6j7H2cYqpzWloTqBFZq
3ngPG8BcY0M3nY0m3BHjnJiRLJMZKOv+KJkQpIOcB5exwT3D8fzUvv/otLIofQjw
ONFPRQmHvf+mQkTcbOTl1dv/6QCqTkUA69tWIXcOffWjYQo5DCxHLleJTYS1i1X6
ZeFnof+VeJDo1TYLwXN6YB6MCQKO9OqSWWgfI2SszvcPs3i0E48Lcv7gNroAijDw
fj0C/vd2+JC75+CsBFi8S8bNdkhsmKhe0HgxeT7jwLab1gNaj8e0aknZAuetrJg0
l41PulabuALNWlHoAQ+IGFYKhcMx3/o/mcBlUkUsnpsM0b7s3KHCzcJKBSdIsyyL
K4lxLaE1fkf6LKK4nEnnSaDhtA9Czdoe6AqAK3HC8qeTD3YFRasFs55aQaHFB3H/
JVUj7PTlyydPVFWWbXqahLFjLEtglZC6uyR9yv3gxgg1jOLYRDZzn3ejvPhrwU9d
ioOuUdfmbvaDPwEEyh0C4gBo21552sTqwbJEiG04lMbetA8fARWmWEwV4fMs35Ow
k0n8/TaiQeu8lNe8lZUfXqL1MWc+iUyjoiVBZ64D3vKF+vY1Llv+VgAsfjO3jJox
CEKhFTSxS24XH8Sl00i1gqhqfnGPgYyG3HvZ9NyewIVA6X4zc587EAfd09ia/8/P
qgaUjbkCDQRep/3QARAAxGCv2RolhTi7zVBbi3dDBbmqeG8SApC9Fhx67sJbdLjU
wnzDG686ToA+IQvg1z8hlm/hiIA6uYOZGYwZFo9csmOcvvLUopGXzWHVX/aTTYz4
Ct5fPfgfPxLZFrN0Q4Myij/dLQySDn1KQATkJI+OBTy601+oTa9jYl3nukISJAii
Log5S23vxY+3V3A2qL/VBwJPBiq3hCN7R4L/1FsDixDEeDIe8GyHCBN4mmrQLGku
2MYF51YLWrt++n+zObHN5+v+KKRYvUadB3t50fp/2ONT48jXiAmpsTJhpWeLOF0E
DbxxJ5tSAZ/VD1pe6KyGGgLOS0iPtGRy7a4++WMX6fZUeUASIKeihbziEIJrti7S
TW4YCU88WzmviCFnDTQUdTEFLh6GolSWKROksUT6A36h/CfKPgVt4b+1QDsPaXpl
qe5lYfZ6TCssANTNIy5zTHleNjI9K7EdMAy5a1PDKuJ3Lrii1npnj35ciKuG3on1
Ynp+dPd+GyWvXEuE1MSkwgAIiEZusUqzVsXbTbnsuxu2VygRP1fZCR3Ql/SNcpXL
mdGaB1WTJQtnN1Yh4zWrF53fgihWODdNDZZ6fWYAj6U3/pOJ+kF7x7q2eaYYs/iv
jBVeMJTJ3Ri4iPzWyLbuOjLezzix31OmGboakG+G2gzIk+AF1mX46uTbCUz2+9UA
EQEAAYkCPAQYAQgAJhYhBNonoZrnQTXetL8wsUTm9DxlQnnPBQJep/3QAhsMBQkA
AVGAAAoJEETm9DxlQnnP90AP/0p1HcOM5HgbA164jLmN2n6E0FQ6f6MyxN2L2zKT
aMTXpiRyRnuB6dyfeNZl76s3GHlbuQh2mNaQTU/jl6rK2IAnVEVIoioiMQnJCaVi
krC4dmMTUINQwRcKK+IlArelndc9lT1MPnRt/k7rqxR5UnZ1kHeZHv6g9Lfd1h4i
0SaHmqsZqvekNJe1MifpSFYtAGG/t7r+8ppaR6Twtu/SdKtjNH38/JgtlimQKzRd
6SB6PIfcFV10tpyLDhivrYxQFYkTQ4+jh5GR72Z4l6h9lJHXHsuSjBmK9Wo/bTLi
P3bTjSHhSjHviECR5OM9Hu6j6JHoiVW8xdlS24zG40R6TKWlMx5MnRon4ildTkY/
1tgp/Bp6RE7TPU8OfMAcsvQjEH+lsVxEpRlmZMjc2Vz5LnqAGqghk/TvzbTDsq0c
Whf8BU6ccy1+DSiws9zkvIEbmk21+lXsuIsU9o4mgvjMS6LwcUi+OiyYvW9M8Uel
Uch9EUU/gtwYBhTZaUyI080dxHSRltVE+VVUdzEEj4voNAApSe66o0tzvxvrj7ic
wGc2V2wtE1tokbcLde6nt5VZaXhrkecO8++U77BIlldq1VzfMgRh6nyLRhAxms7M
xHyKdobucgcjVeGAMzIthiT6YGX9St3pcLvKOdJqligXZDtZK+t5qNW8gVnAd0LY
Shsi
=gXHO
-----END PGP PUBLIC KEY BLOCK-----
```

### Chiffrer et déchiffrer

Méthode

Pour chiffrer et déchiffrer un message, il est nécessaire d'utiliser les commandes ci-dessous. Il est possible tester ces commandes dans la console d'un repl Bash. Pour chiffrer, on spécifie l'adresse mail de la clef publique du destinataire (`recipient`) du message `message.txt`.

```
1 gpg --output msg.txt.gpg --encrypt --recipient paul@blabla.bla message.txt
```

Pour déchiffrer, on utilise :

```
1 gpg --output message.txt --decrypt msg.txt.gpg
```

Ici, la clé par défaut de l'utilisateur sera utilisée pour déchiffrer.

### Révoquer sa clé

⊕ Complément

Il peut arriver qu'une clé privée fuite. Ainsi, la sécurité et l'identité du propriétaire peuvent être atteintes. Un attaquant pourra utiliser la clé pour déchiffrer les documents ou même pour usurper l'identité en signant des mails trompeurs grâce à cette clé privée.

Pour éviter cela, le propriétaire de la clé pourra générer, grâce à sa clé privée, un certificat de révocation qu'il transmettra à ses contacts. Ce certificat est une sorte de preuve cryptographique que la clé n'est plus d'actualité. Ainsi, un attaquant ne pourra plus se servir de la clé privée.

Dans l'exemple de génération donné plus haut, le certificat de révocation a été créé et enregistré dans le fichier :

```
1 '/home/john/.gnupg/openpgp-
  revocs.d/DA27A19AE74135DEB4BF30B144E6F43C654279CF.rev'
```

## À retenir

- OpenPGP permet de sécuriser et certifier ses communications.
- GnuPG est un outil très simple permettant de générer, gérer et utiliser des clés OpenPGP.

# 10. Appliquer la notion

Nous allons manipuler des clés GPG sur un repl Bash.

## Question 1

[solution n°4 p. 31]

Générer une paire de clés pour un utilisateur « John Smith » d'adresse e-mail `john@smith.xyz`.

## Question 2

[solution n°5 p. 31]

Créer un fichier `message.txt` de contenu :

```
Message secret
```

Et le chiffrer pour vous même avec `gpg` en un fichier `msg.txt.gpg`.

### Indice :

On utilisera une commande de la forme.

```
gpg --output fichier_chiffré --encrypt --recipient addresse_email fichier_à_chiffrer
```

## Question 3

[solution n°6 p. 31]

Faire une copie de ce fichier nommée `copy.txt.gpg` et changer quelques octets dans la copie.

### Indice :

On pourra utiliser la commande `cp`.

## Question 4

[solution n°7 p. 32]

Déchiffrer le fichier original `msg.txt.gpg` et le fichier altéré `copy.txt.gpg`.

Que se passe-t-il dans le cas du fichier altéré ?

# 11. Signer ses mails

### Objectifs

- Comprendre ce qu'est une signature au sens OpenPGP ;
- Savoir signer ses mails.

### Mise en situation

Dans le cadre de communications électroniques, la confidentialité du message n'est pas le seul élément important. Il est également nécessaire de pouvoir être certain de l'identité de la personne qui nous envoie le message.

Les attaques par hameçonnage consistent typiquement à se faire passer pour une personne de confiance, afin de recueillir des informations confidentielles, comme par exemple un mot de passe.

La cryptographie asymétrique est également utilisable pour signer des messages.

On utilise dans ce cas sa clé privée pour chiffrer une signature qui est diffusée en même temps que le message. N'importe qui disposant de la clé publique peut déchiffrer la signature et ainsi vérifier que le message a bien été signé par le détenteur de la clé privée correspondante. On note donc que les clés privées et publiques peuvent être utilisées toutes deux pour chiffrer et pour déchiffrer.

OpenPGP et GnuPG permettent également de gérer les signatures.

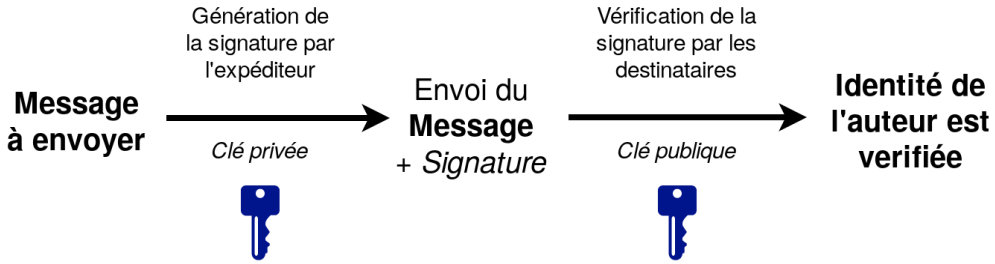
Il est possible par exemple d'utiliser GnuPG avec un logiciel de mail, comme Thunderbird, pour envoyer des messages chiffrés et signés.

#### Signature grâce à GPG

Fondamental

La signature consiste à chiffrer (avec la clé privée) le condensat du message et de joindre ce cryptogramme au message à envoyer. Le destinataire n'aura qu'à déchiffrer la signature avec la clé publique pour bien vérifier que le message a été envoyé par le propriétaire de la clé privée. Cette identification du signataire est permise car le chiffrement asymétrique garantit qu'il n'existe qu'une clé publique pouvant déchiffrer un message de la clé privée (et l'inverse est également vrai).

Dans le même temps, la signature garantit l'intégrité du message : le condensat du message reçu est comparé avec le condensat chiffré (signature). Si les deux condensats sont différents, le message a été altéré en chemin et on ne peut pas lui faire confiance.



### Utilité

Conseil

Signer un mail permet d'assurer son identité à ses interlocuteurs. Si ces interlocuteurs signent également leur mails, cela permet d'éviter à coup sûr des tentatives d'hameçonnage. Ceci est complémentaire de l'usage de mails au contenu chiffré.

### Signer manuellement

Méthode

Voici comment signer un message :

```
1 # sig.asc est le fichier contenant la signature
2 gpg --detach-sign -o sig.asc msg.txt
```

Voici comment vérifier la signature d'un message :

```
1 gpg --verify sig.asc msg.txt
```

Si la signature est correcte et que le message n'a pas été modifié, on obtiendra un message similaire à :

```
1 gpg: Signature made Fri 15 May 2020 04:10:56 PM CEST
2 gpg:                using RSA key CC15797D1FFEB346F5A87AFE0547178FEEDE7D6B
3 gpg: Good signature from "Quentin Duchemin <quentinduchemin@tuta.io>" [ultimate]
```

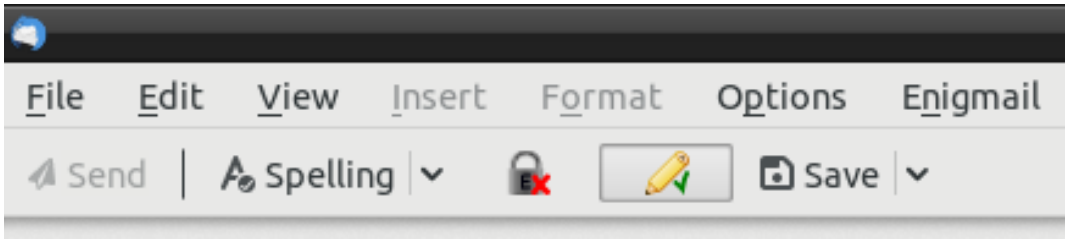
### Enigmail

Méthode

Enigmail est un add-on du client mail Thunderbird. Avec cet add-on, il est possible de générer et gérer ses clés directement avec l'interface graphique. Cet outil a également une mise en place pour débutants qui facilite une première utilisation de clés GPG. Voici le guide pour installer et mettre en place Enigmail : https://enigmail.net/index.php/en/user-manual/quick-start.

Une fois l'add-on bien configuré, lors de chaque envoi de mail, il suffira d'appuyer sur un simple bouton pour chiffrer ou signer le message. Pour chiffrer, il est nécessaire de posséder la clé publique du destinataire.

À chaque réception d'un mail, si Enigmail trouve une clé publique correspondant à l'adresse mail de l'expéditeur, la signature GPG sera automatiquement vérifiée.




Attention

Les versions les plus récentes de Thunderbird (supérieures à la version 78) intègrent directement la gestion de OpenPGP, sans avoir besoin d'installer Enigmail. Le fonctionnement reste très similaire à celui de l'extension.

### Publier sa clé publique

Complément

Pour faciliter les échanges de clés publiques, il est possible de les publier sur des serveurs de clés qui centralisent un grand nombre de clés publiques. Par exemple, celui du MIT est très utilisé : https://pgp.mit.edu/. Si une clé doit être retirée, il est également possible d'envoyer le certificat de révocation à ce type de serveur.

Ces serveurs de clés sont un des moyens (avec le système de certification de clés) qui permet de s'assurer qu'on possède la bonne clé publique de l'expéditeur d'un mail et qu'un attaquant n'a pas transmis une fausse clé publique pour hameçonner sa cible.

### Les sous-clés

Complément

Pour aller plus loin dans l'utilisation de clés GPG, il est très pratique d'utiliser des sous-clés. La première clé primaire est nommée « clé maîtresse » et peut générer des sous-clés dont l'usage peut être restreint : signer uniquement, chiffrer et/ou certifier. Cela facilite les manipulations en cas de fuite de clé.

Si une sous-clé fuite, il suffira de la révoquer et d'en générer une nouvelle.

Si la clé maîtresse fuite, il faut générer une nouvelle paire de clés et transmettre sa nouvelle clé publique à tous ses contacts.

## À retenir

- Signer ses mails permet aux interlocuteurs d'être certain de l'identité de l'expéditeur et de vérifier l'intégrité du document.
- La signature GPG de ses mails peut facilement être mise en place grâce au plugin Enigmail pour Thunderbird.

# 12. Appliquer la notion

On se propose de vérifier la signature d'un document. Utiliser pour ce faire un shell local ou un repl Bash.

## Question 1

[solution n°8 p. 32]

Le fichier `public.gpg` contient une clé publique GPG d'un certain Marc Studi.

*Enregistrer le fichier `public.gpg` sur votre ordinateur.*
(cf. public.gpg)

Importer ce fichier dans l'utilitaire `gpg`.

### Indice :

On peut utiliser l'aide de gpg pour obtenir plus facilement l'information :

```
1 gpg --help
```

### Indice :

L'option `--import` permet d'importer une clé manuellement.

## Question 2

[solution n°9 p. 32]

Vous recevez un message `message.txt` accompagné de sa signature `signature.asc`.
Télécharger les deux fichiers.

*Enregistrer le fichier `signature.asc` sur votre ordinateur.*
(cf. signature.asc)

*Enregistrer le fichier `message.txt` sur votre ordinateur.*
(cf. message.txt)

Vérifier la signature du fichier ci-joint. Quelle est l'adresse mail du signataire ?

### Indice :

Utiliser

```
1 gpg --verify signature.asc message.txt
```

# 13. Essentiel

Le chiffrement symétrique consiste à utiliser une clé qui permet à la fois de coder et de décoder de l'information. C'est une technique qui est utilisée pour des échanges temporaires entre deux personnes ainsi que pour protéger des données que l'on ne souhaite pas partager.

Le chiffrement asymétrique implique de disposer d'une paire de clés. Imaginons qu'Alice possède la clé publique **alice.pub** et la clé privée associée **alice.pri** :

- N'importe qui peut utiliser la clé publique **alice.pub** pour chiffrer des messages confidentiels pour Alice. On est certain que seule Alice pourra les lire, car c'est la seule à posséder la clé privée de déchiffrement **alice.pri**.
- Alice peut aussi utiliser sa clé privée **alice.pri** pour chiffrer des signatures. N'importe qui pourra déchiffrer cette signature avec la clé **alice.pub**, ce qui prouvera que le message vient d'Alice, puisque, encore une fois, elle seule a pu chiffrer une telle signature.

OpenPGP est un standard qui permet de définir une façon robuste de créer des clés et de les utiliser pour chiffrer et signer.

GnuPG est un logiciel libre multi-plateformes qui implémente ce standard.

Enigmail est une extension du logiciel de mail Thunderbird qui permet d'envoyer des mails signés et chiffrés grâce à GnuPG.

# 14. Exercice final

## Quiz - Culture

[solution n°10 p. 32]

### Exercice

Le chiffrement nécessite obligatoirement l'utilisation d'une paire de clés.

- A Vrai
- B Faux

### Exercice

Le chiffrement symétrique est moins sécurisé car il n'utilise qu'une clé ?

- A Vrai
- B Faux

### Exercice

On peut utiliser le chiffrement symétrique dans les cas suivants :

- A Protocoles de communication sécurisé
- B Chiffrer un disque dur
- C Signer des fichiers ou des messages

### Exercice

Le chiffrement asymétrique est utilisé pour les raisons suivantes :

- A Protocole de communication sécurisé
- B Signer des fichiers ou des messages
- C Envoyer des messages confidentiels

## Quiz - Méthode

[solution n°11 p. 34]

### Exercice

Pour transférer une clé secrète AES à un contact, en toute sécurité, je peux :

- A Chiffrer la clé avec la clé RSA publique de mon contact et lui envoyer le cryptogramme.
- B Chiffrer la clé avec la clé GPG privée de mon contact et lui envoyer le cryptogramme.
- C Chiffrer la clé avec la clé GPG publique de mon contact et lui envoyer le cryptogramme.
- D Envoyer la clé dans une archive protégée par un mot de passe.

### Exercice

Laquelle de ces propositions correspond à ce qui est le plus adapté pour prouver son identité ?

- A OpenPGP
- B RSA
- C AES
- D César

## Quiz - Code

[solution n°12 p. 35]

### Exercice

Quelle commande faut-il lancer pour générer une paire de clés RSA utilisables pour SSH ?

### Exercice

Quelle commande faut-il lancer pour importer la clé publique contenue dans le fichier `snowden.gpg` ?

### Exercice

Quelle commande faut-il lancer pour vérifier la signature `sig.asc` du fichier `prism.txt` ?

### Exercice

Quelle commande faut-il lancer déchiffrer le contenu du fichier `swartz.txt` ?

# Solutions des exercices

## Solution n°1

[exercice p. 8]

Le décalage était 2 et le message : « *La machine Enigma n'était pas sécurisée* »

## Solution n°2

[exercice p. 12]

On utilise la commande :

```
1 openssl aes-256-cbc -pbkdf2 -iter 10000 -d -in secret_data.enc -out msg.txt -pass
  file:aes.priv
```

Le message est : « En savoir plus grâce à 'An Introduction to Number Theory with Cryptography' par J. Kraft et L. Washington »

## Solution n°3

[exercice p. 17]

On peut utiliser la commande suivante pour déchiffrer.

```
1 openssl rsautl -decrypt -inkey private.pem -in msg.enc -out message_en_clair.txt
```

Puis avec `cat` pour voir le contenu du message :

```
1 cat message_en_clair.txt
```

Qui est :

```
1 Bravo H4ck3rm4n
```

## Solution n°4

[exercice p. 22]

On utilise :

```
1 gpg --generate-key
```

On entre les informations et on choisit une phrase de passe suffisamment grande.

## Solution n°5

[exercice p. 22]

On utilise :

```
1 gpg --output msg.txt.gpg --encrypt --recipient john@smith.xyz message.txt
```

## Solution n°6

[exercice p. 22]

```
1 cp msg.txt.gpg copy.txt.gpg
2 nano copy.txt.gpg
```

Le fichier étant chiffré, dans l'éditeur de texte, il y aura des caractères illisibles. Il suffit de supprimer un ou deux caractères au milieu du fichier pour l'altérer.

## Solution n°7

[exercice p. 22]

On déchiffre ainsi le premier fichier :

```
gpg --output message_dechiffre.txt --decrypt msg.txt.gpg
```

Le contenu est identique au contenu du fichier original `message.txt`.

Pour la copie altérée, le déchiffrement est impossible.

```
gpg --output copy.txt --decrypt copy.txt.gpg
gpg: no valid OpenPGP data found.
gpg: decrypt_message failed: Unknown system error
```

Dans le cas où l'altération (légère) est réalisée en début ou en fin de fichier, il est possible que le déchiffrement reste réalisable car seul l'entête ou la signature est légèrement touchée

## Solution n°8

[exercice p. 26]

On utilise :

```
gpg --import public.gpg
```

On obtient en sortie :

```
gpg: key 0x613572E5E3D19D73: "Grace Hopper <grace.hopper@ibm.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1
```

## Solution n°9

[exercice p. 26]

L'adresse est celle de Grace Hopper : `grace.hopper@ibm.com`

**Attention aux usurpations d'identité !** ⊕ Complément

Comme le montre cet exemple il est possible de générer une clé avec n'importe quelle adresse e-mail.

Il est donc important de vérifier la clé de l'émetteur avant de lui faire confiance.

GPG introduit un mécanisme de **certifications**, qui consiste grossièrement à dire « cette clé appartient bien à cette personne » grâce à des mécanismes cryptographiques. Si une clé est suffisamment certifiée par plusieurs personnes, on pourra lui faire confiance.

## Solution n°10

[exercice p. 28]

**Exercice**

Le chiffrement nécessite obligatoirement l'utilisation d'une paire de clés.

A Vrai

Faux

B

Le chiffrement symétrique utilise la même clé pour chiffrer et déchiffrer. ✓

**Exercice**

Le chiffrement symétrique est moins sécurisé car il n'utilise qu'une clé ?

A Vrai

Faux

B

D'un point de vue mathématique et technique, les chiffrements symétrique et asymétrique font appel aux mêmes propriétés, ils répondent à des besoins différents. ✓

**Exercice**

On peut utiliser le chiffrement symétrique dans les cas suivants :

Protocoles de communication sécurisé

A

Par exemple : SSL/TLS pour le Web ✓

B Chiffrer un disque dur ✓

C Signer des fichiers ou des messages

**Exercice**

Le chiffrement asymétrique est utilisé pour les raisons suivantes :

Protocole de communication sécurisé

A

Par exemple : SSH pour accéder à un serveur distant ✓

B Signer des fichiers ou des messages ✓

C Envoyer des messages confidentiels ✓

# Solution n°11

[exercice p. 29]

**Exercice**

Pour transférer une clé secrète AES à un contact, en toute sécurité, je peux :

A Chiffrer la clé avec la clé RSA publique de mon contact et lui envoyer le cryptogramme. ✓

B Chiffrer la clé avec la clé GPG privée de mon contact et lui envoyer le cryptogramme.

Il est très dangereux de partager ses clés privées RSA ou GPG avec d'autres personnes. De plus, n'importe qui avec la clé publique pourra déchiffrer le message.

C Chiffrer la clé avec la clé GPG publique de mon contact et lui envoyer le cryptogramme. ✓

D Envoyer la clé dans une archive protégée par un mot de passe.

Il est possible de « casser » le mot de passe des archives et le transfert du mot de passe lui même demandera de sécuriser la communication.

**Exercice**

Laquelle de ces propositions correspond à ce qui est le plus adapté pour prouver son identité ?

A OpenPGP ✓

B RSA

C AES

D César

Une clé OpenPGP est toujours rattachée à une identité (nom, adresse e-mail), qui fait l'objet de certifications d'autres personnes, ce qui augmente la confiance dans l'identité de la personne.

# Solution n°12

[exercice p. 29]

**Exercice**

Quelle commande faut-il lancer pour générer une paire de clés RSA utilisables pour SSH ?

ssh-keygen

**Exercice**

Quelle commande faut-il lancer pour importer la clé publique contenue dans le fichier `snowden.gpg` ?

gpg --import snowden.gpg

**Exercice**

Quelle commande faut-il lancer pour vérifier la signature `sig.asc` du fichier `prism.txt` ?

gpg --verify sig.asc prism.txt

**Exercice**

Quelle commande faut-il lancer déchiffrer le contenu du fichier `swartz.txt` ?

gpg --decrypt swartz.txt

# Crédits des ressources

**Chiffre de César** p. 6
*Attribution - Partage dans les Mêmes Conditions - Patricia.fidi Wikipédia*