http://WWW.CABARE.NET ©

# Sécurisation Cryptage Windows – sys 30 – Cours T.P. -

## Cryptage AxCrypt Veracrypt Windows

**Michel Cabaré – Ver 1.0 –Mai 2017-**

**Sécurisation Cryptage Windows**

**Cours et T.P.**

**Michel Cabaré – Ver 1.0 – Mai 2017**

www.cabare.net ©

# TABLE DES MATIERES

# BITLOCKER VS EFS

## Cryptage Logiciel

**BitLocker** est conçu pour protéger la totalité des fichiers personnels et des fichiers système d'un lecteur logique. Vous pouvez utiliser **BitLocker** pour chiffrer tous les fichiers sur des lecteurs de données fixes (tels que des disques durs internes) et utiliser **BitLocker To Go** pour chiffrer les fichiers sur des lecteurs de données amovibles (disques durs externes ou disques flash ou USB).

Le **système de fichiers EFS** permet de protéger des fichiers individuels sur tout lecteur, en fonction de l'utilisateur.

**N.B**: Le chiffrement est perdu en cas de transmission par mail ou via le réseau.

**BitLocker** (logiciel et materiel)

- BitLocker chiffre tous les fichiers personnels et système sur un lecteur de données fixe et sur des lecteurs de données amovibles.
- BitLocker ne dépend pas des comptes d'utilisateurs individuels associés aux fichiers. BitLocker est activé ou désactivé, pour tous les utilisateurs ou groupes.
- BitLocker peut utiliser le module de plateforme sécurisée (TPM), une puce spéciale présente sur de nombreux ordinateurs et qui prend en charge des fonctionnalités de stockage et de chiffrement.
- Vous devez être un administrateur pour activer ou désactiver le chiffrement BitLocker

**EFS, Système de fichiers EFS** (logiciel)

- EFS chiffre les fichiers personnels et système un par un et ne chiffre pas le contenu complet d'un lecteur.
- Le système de fichiers EFS chiffre des fichiers en fonction du compte d'utilisateur associé. Si un ordinateur comporte plusieurs utilisateurs ou groupes, chacun peut chiffrer ses propres fichiers individuellement.
- Le système de fichiers EFS ne requiert ou n'utilise pas de matériel spécial.
- Il n'est pas nécessaire que vous soyez un administrateur pour pouvoir utiliser le système de fichiers EFS.

Si cette clef USB n'est pas présente dans un des ports USB de votre ordinateur, Windows ne démarrera pas, à moins que vous saisissiez la clef de recouvrement Bitlocker(numéro super long de secours). La clef USB devient en quelque sorte votre puce TPM, mais avec moins de fonctionnalités. La puce TPM possède des fonctionnalités avancées permettant de ne pas démarrer Windows du tout, si le disque dur n'est pas dans le même ordinateur, que le BIOS, le MBR, ou le Gestionnaire de démarrage ont changés.

## Fonctionnement d'EFS

Le système **EFS** inclus dans windows permet à un utilisateur de chiffrer un fichier de manière à ce que celui-ci ne puisse pas être lu par quelqu'un d'autre que lui (saut un compte particulier, appelé **agent de récupération**, généralement un administrateur…)

Sans la « clé », le fichier est véritablement indécryptable, même par l'administrateur. Seul un agent de récupération peut être alors utilisé pour relire le fichier. POUR LIRE UN FICHIER EFS, même si on dispose de tous les droits NTFS, même si on s'est approprié le fichier , IL EST NECESSAIRE DE DISPOSER DE LA CLE PRIVEE (ou d'être AGENT DE RECUPERATION). Ainsi un disque « démonté et remonté sur un autre système » ou un portable reste protégé.

EFS fonctionne en arrière plan, et stocke dans le fichier lui-même un certain nombre d'informations (qui a chiffré, avec quelle clé, liste des agents de récupération…).

**N.B**: certaines applications peuvent sauvegarder des fichiers temporaires, et donc risquent de laisser des traces lisibles de vos fichiers, il vaut mieux pour éviter cela **demander de crypter tout un dossier, plutôt que uniquement un fichier**…. (dans ce cas, chaque fichier posé dans ce dossier sera automatiquement chiffré avec la clé de son propriétaire…)

**Pour qu'EFS fonctionne il est nécessaire que :**

- Vous soyez sur un volume NTFS
- L'utilisateur ait un certificat valide d'utilisateur EFS
- Un compte d'agent de récupération d'EFS au moins ait un certificat valide de récupération d'EFS. (une autorité de certification n'est pas nécessaire, EFS produit automatiquement ses propres certificats pour les utilisateurs et les comptes d'agent de récupération…)

**Lorsque EFS chiffre un fichier :**

Lorsqu'un utilisateur chiffre un fichier, les opérations suivantes sont exécutées :

- Production d'une clé de chiffrement
- Chiffrage du fichier avec cette clé de chiffrement
- Chiffrage de cette clé de chiffrement avec la clé publique de l'utilisateur (si besoin création de cette clé et du certificat associé)
- Stockage de cette « clé de chiffrage chiffrée » dans le fichier dans une zone nommée DDF (une par fichier)

**Résultat** : lorsque l'on essayera de lire un fichier chiffré, seul le possesseur de la clé privée de cet utilisateur, peut déchiffrer la clé de chiffrage employée lors du codage. Ce ne peut être que l'utilisateur

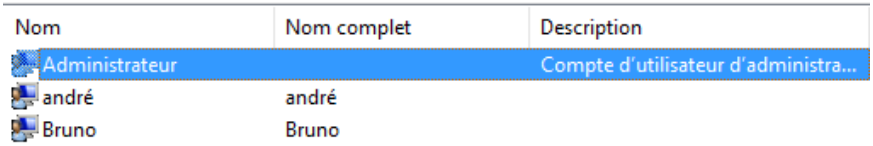
Mais n'oublions pas que EFS permet à des comptes définis d'agent de récupération de déchiffrer et de récupérer le fichier, au cas où la clé privée serait détruire (exemple, un utilisateur renvoyé crypte tous ses fichiers avent de partir…) donc lorsqu'un utilisateur chiffre un fichier les opérations suivantes sont aussi exécutées :

- Chiffrage de cette clé de chiffrement avec la clé publique de chaque agent de récupération
- Stockage de cette « clé de chiffrage chiffrée » dans le fichier dans une zone nommée DRF (une ou plusieurs par fichier)
- Chaque fois que l'on manipule ce fichier, copie, ouverture, modification, la zone DRF est mise à jour selon les besoins

Résultat: seul le possesseur de la clé privée des agents de récupération peu lire le fichier

# Voir les Certificats Utilisateur

Soit 2 utilisateurs locaux sur un poste Windows 10, andré et bruno, respectivement avec comme mot de passe Local10

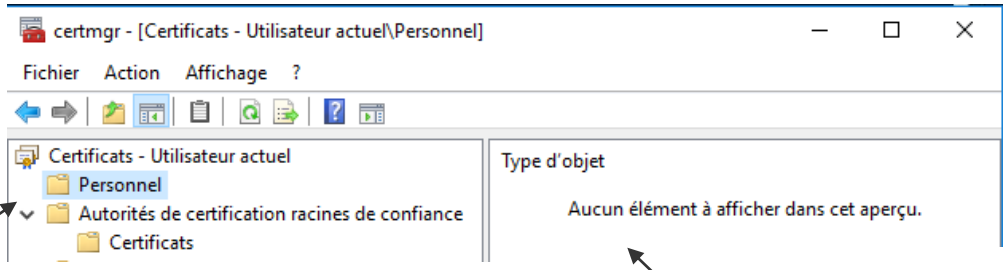



Vérifions d'abords par exemple pour andré que celui-ci ne possède pas de certificat permettant de stocker une clé publique (il n'a jamais chiffré de fichier, et n'a donc jamais eut besoin d'une clé privée+clé publique…)

Plusieurs méthodes existent

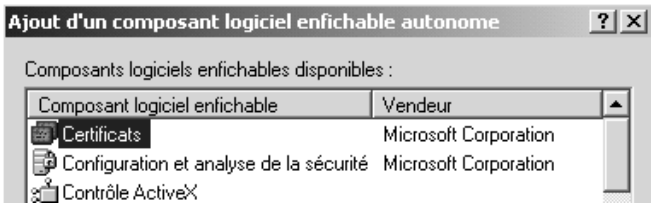
En invite de commande **certmgr.msc**, lance la gestion du **certificat utilisateur** pour l'utilisateur courant

Dans la console en se plaçant sur **Personnel**

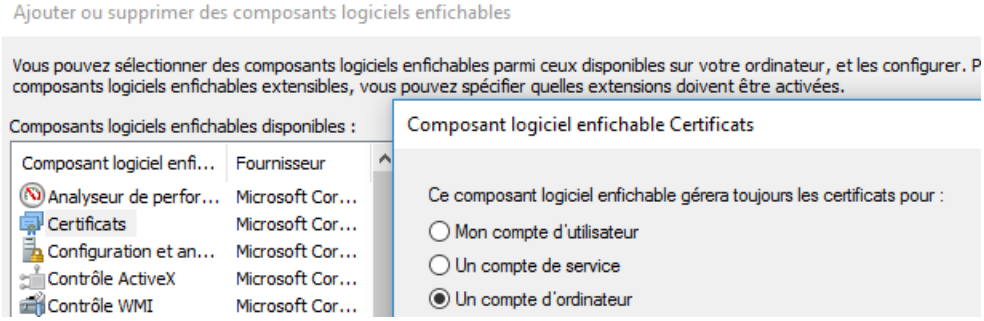



On voit bien l'absence de certificats…

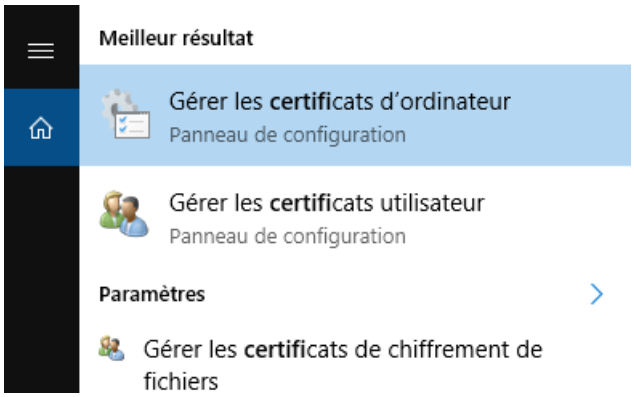
On peut créer une **mmc** avec le composant **Certificats**



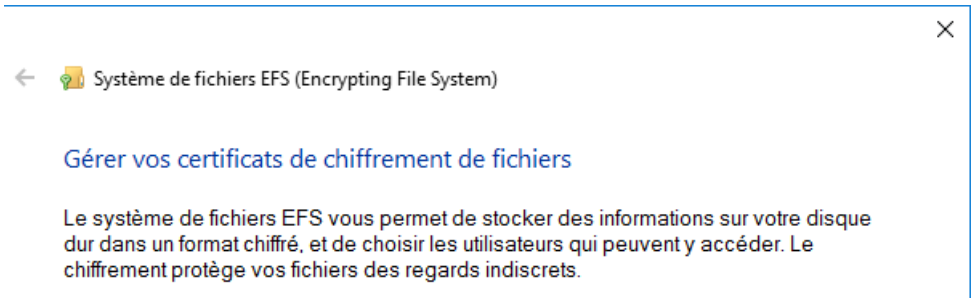

Si on est simple utilisateur, on ne pourra que voir les du **certificat utilisateur** pour l'utilisateur courant , mais si on est administrateur, on pourrait choisir



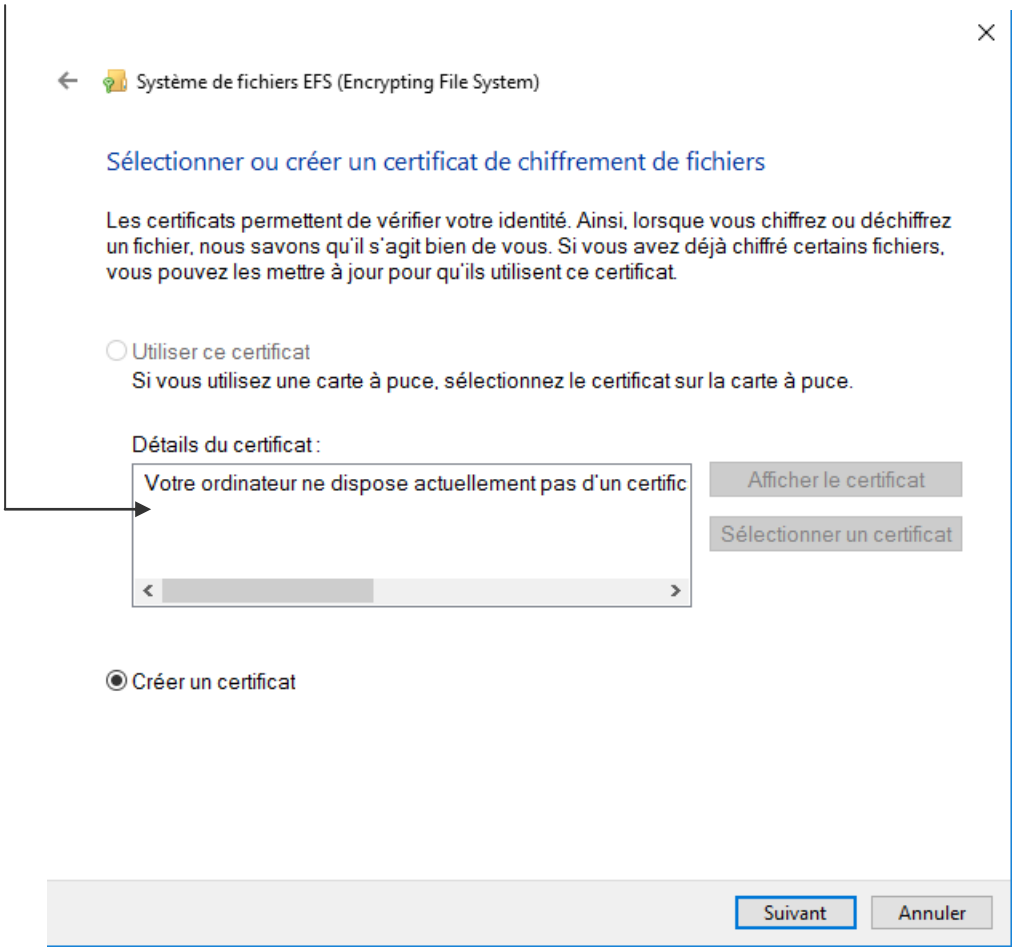
Sous windows 10, en tant qu'administrateur, une recherche **Cortana** avec le mot ***certificat*** propose assez rapidement **Gérer les certificats de chiffrement de fichiers**




Cela déclenche un petit assistant



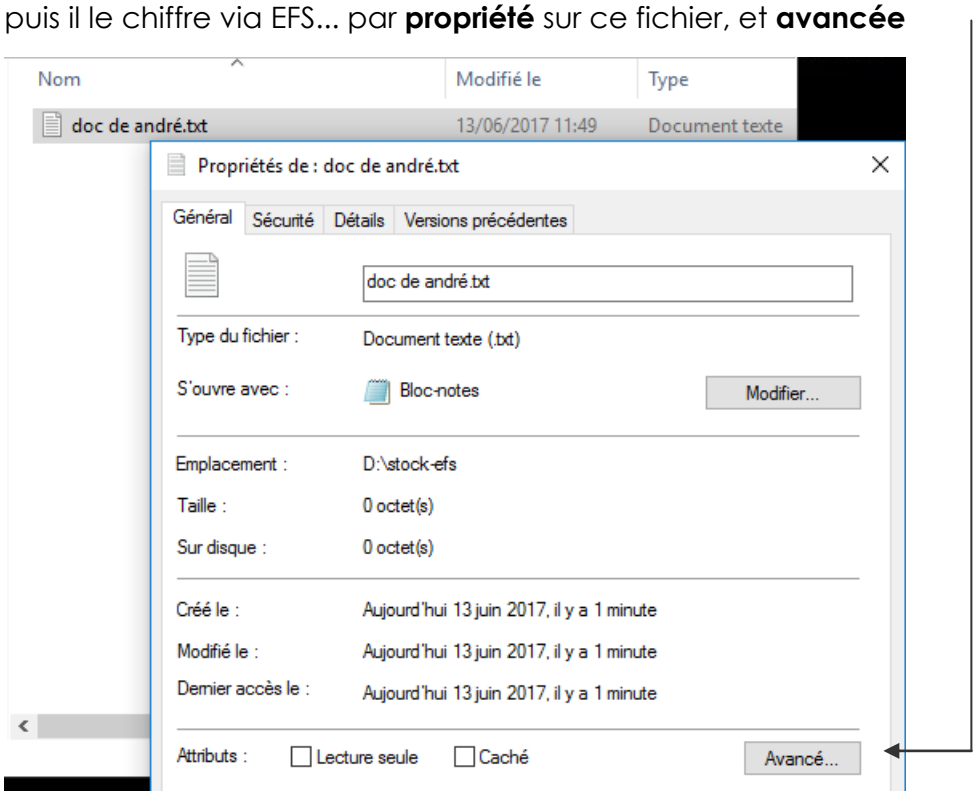

Qui indique bien l'absence de certificat



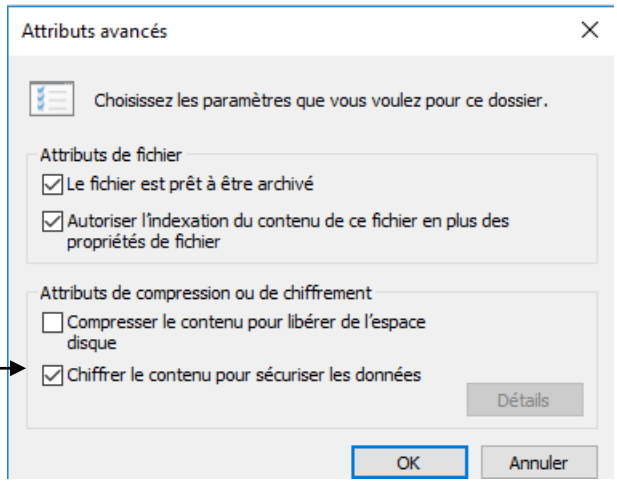
# Cryptage de fichier

Dans un dossier nommé **stock-efs**, andré crée un fichier **doc de andré**

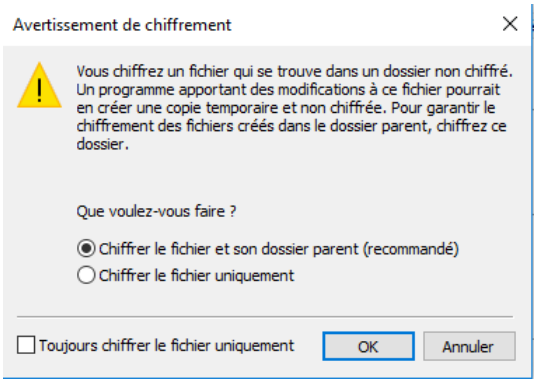

puis il le chiffre via EFS... par **propriété** sur ce fichier, et **avancée**




On demande un cryptage (via EFS) en cochant cette case




Une mise ne garde concernant le cryptage fichier ou dossier apparaît




et voilà !

A partir de maintenant, indépendamment des droits NTFS, Bruno ne peut plus ouvrir le fichier **doc de andré**

Si on crypte que le fichier, les conséquences sont claires

Si on crypte le dossier alors on aura :

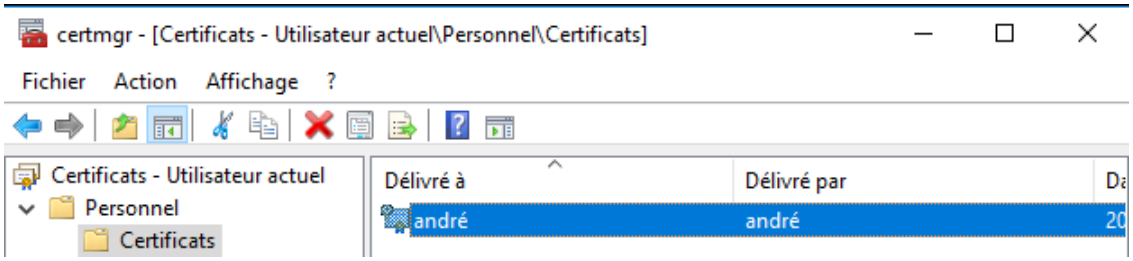
- Tous les fichiers existants sont protégés par notre clé (mais que si on peut écrire dedans au niveau des ACL )
- Tout ce qui sera crée ou copié plus tard dans ce dossier par nous sera protégé par notre clé
- Tout ce qui sera crée par un autre utilisateur sera protégé par sa clé
- Tout ce qui sera déplacé restera inchangé (au niveau cryptage)

IL VAUT MIEUX POSER UN CRYPTAGE PAR DOSSIER VIDE AVANT TOUTE UTILISATION

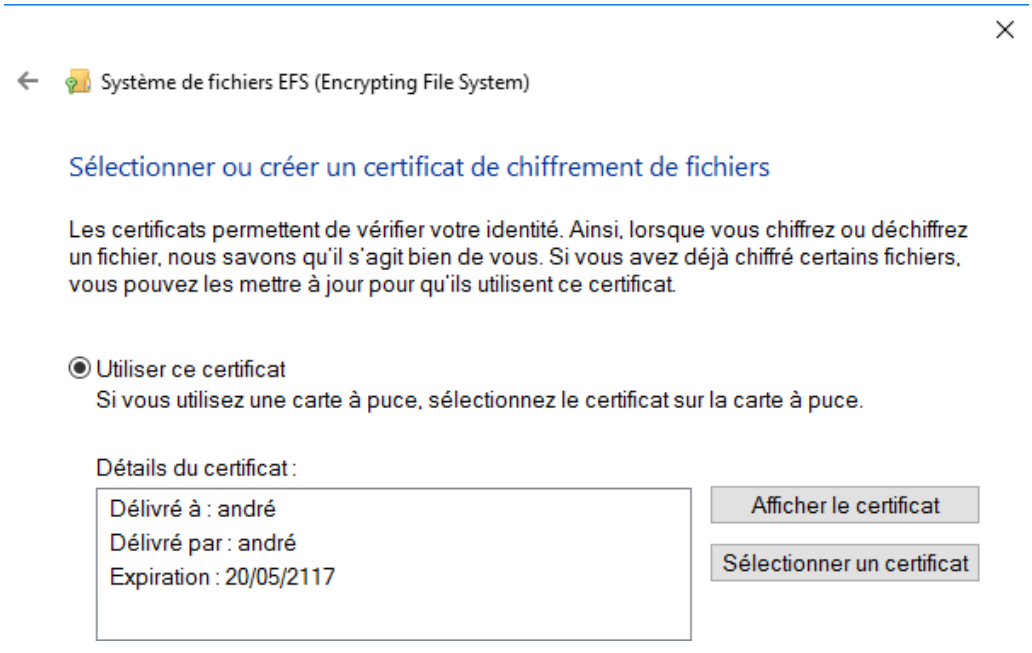
## Vérification Certificats Utilisateur

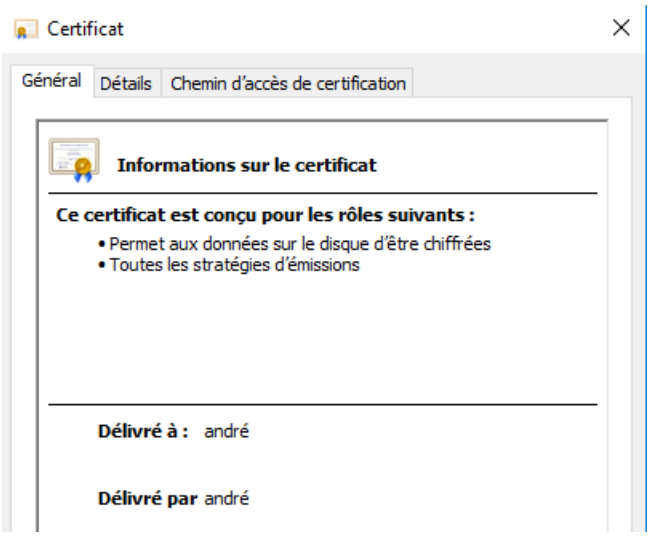
Automatiquement, lors du premier appel à EFS de la part de André, une création de clé a été effectuée, et un certificat a été délivré…

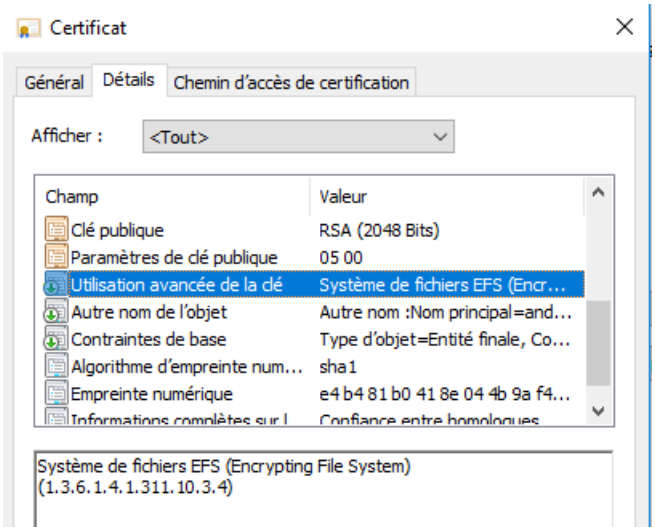
Vérifions nos méthodes, En invite de commande **certmgr.msc**, lance la gestion du **certificat utilisateur** pour l'utilisateur courant




Via Cortana, avec la recherche…



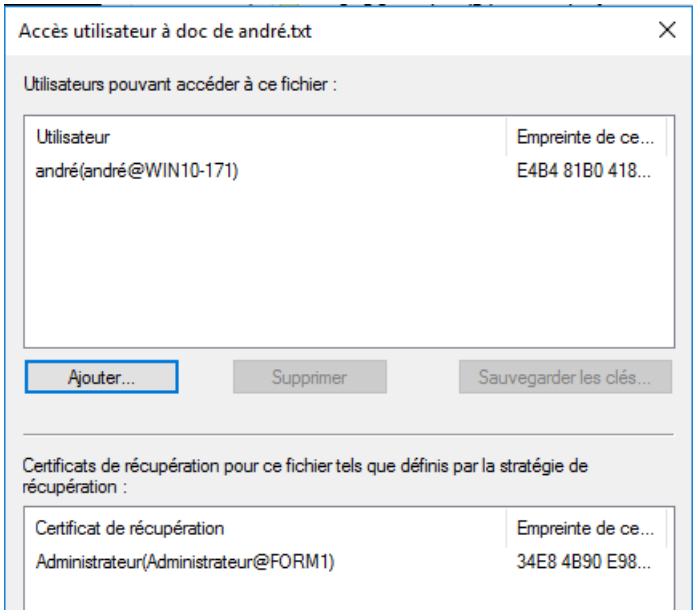



# Agent Récupérateur

Par ailleurs il y a forcément un agent de récupération, on peut le connaître en via les **propriétés** puis **avancées** du fichier, où l'on voit

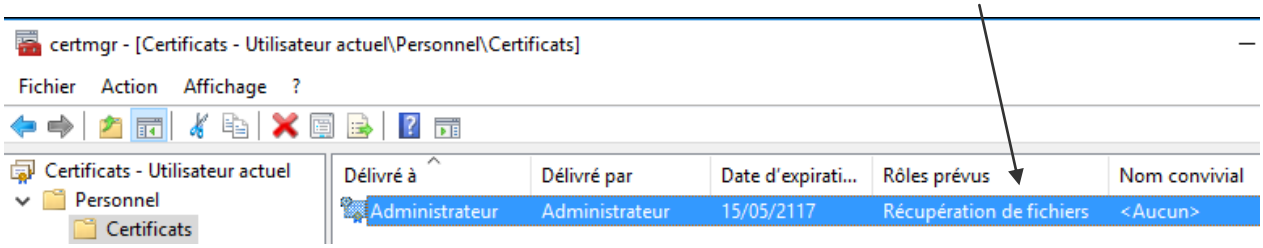
**Certificats de récupération pour ce fichier tels que définies par la stratégie de récupération**




Dans un Domaine, c'est l'Administrateur de domaine qui possède ce rôle,

Sur une machine en Workgroup, c'est l'Administrateur du poste qui possède ce rôle

on peut par conséquent afficher les certificats personnels de l'administrateur, et on devrait voir




**N.B** : Si une machine de domaine sort du domaine, les fichiers crytpés se retrouvent sans agent récupérateur.

# Export clé et certificat EFS de l'utilisateur

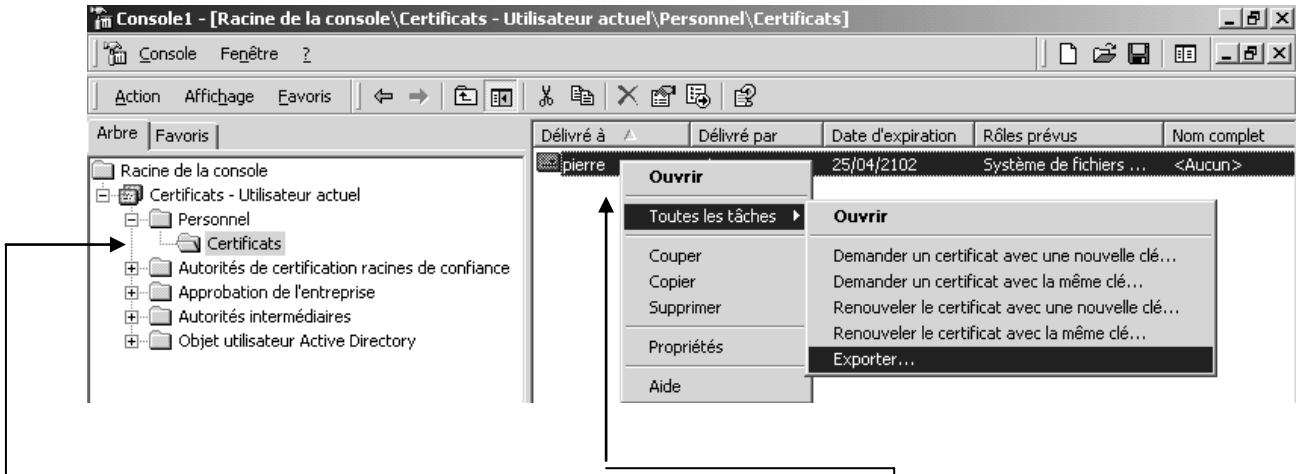
Plusieurs cas pour lesquels il peut être intéressant d'exporter la clé privée de l'utilisateur

- Avoir une copie de sauvegarde de la clé
- Si l'on souhaite qu'un autre utilisateur (à part l'agent de récupération) puisse modifier le fichier, il faut exporter la clé privée de l'utilisateur qui a chiffré le fichier, et l'importer pour l'utilisateur à qui on veut fournir l'accès

Par exemple, pour pouvoir faire en sorte que ***paul*** puisse modifier le fichiers cryptés par ***pierre***, il faut exporter le certificat de ***pierre***, et l'importer pour ***paul***

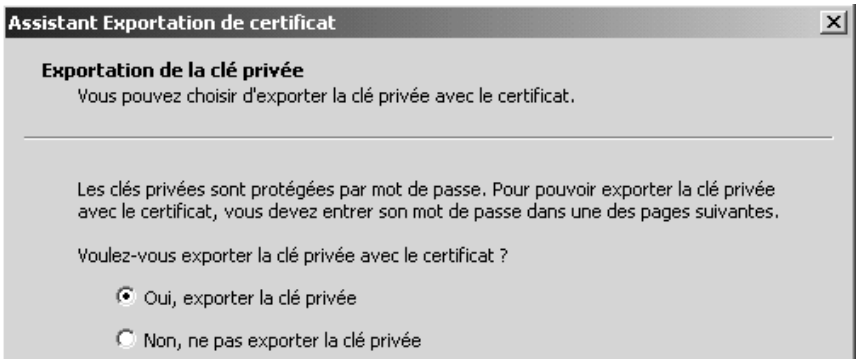
### Exportation du certificat de pierre sur disquette

Ayant une session pour pierre, on crée une mmc certificats

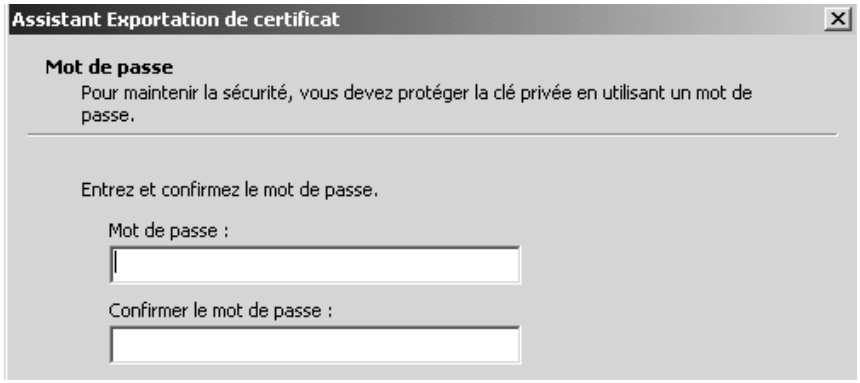



On se place sur certificats à gauche, puis menu contextuel sur le certificat de pierre sur lequel on va enclencher l'exportation de certificat avec la clé privée de Pierre

Les étapes sont claires




Avec exportation de la clé privée



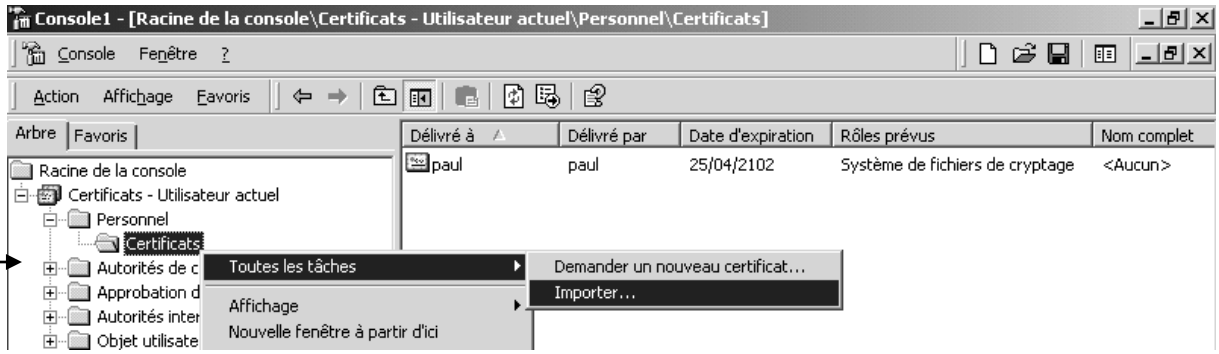
A partir de là, on dispose du certificat de pierre (et de sa clé privée) sur une clé ou tout autre emplacement. Il reste à les donner à paul…

### Importation du certificat de pierre pour paul

Ayant accès à l'emplacement contenant le certificat de pierre,

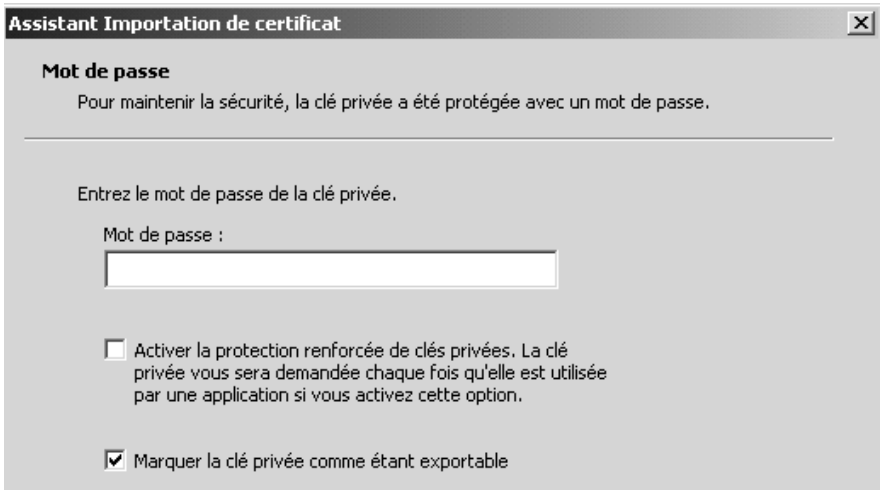


et ayant ouvert une session pour paul, on crée une mmc certificats

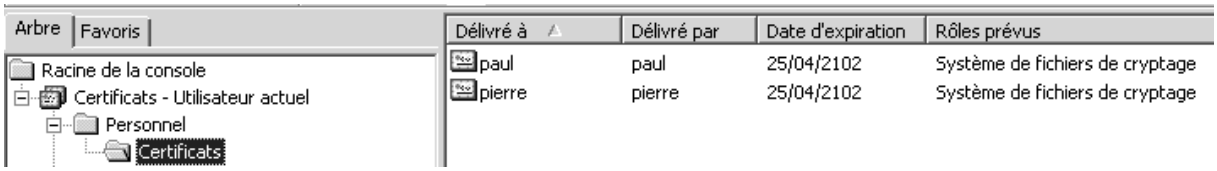



On se place sur certificats à gauche, puis avec le menu contextuel

on va enclencher l'importation de certificat avec la clé privée de Pierre…

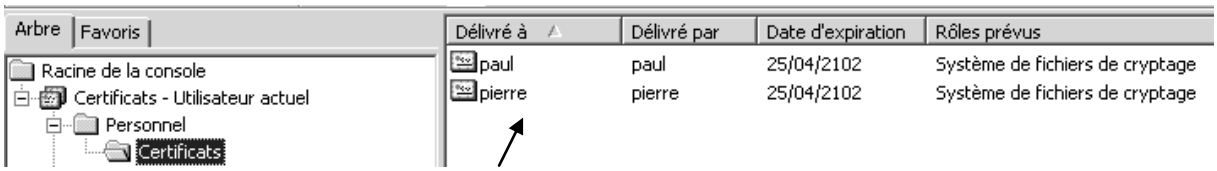

Lorsque l'assistant se termine on devrait avoir




et paul peut désormais modifier les documents de pierre

### Annulation non rétro-active de l'import de certificat

Le temps passe, et paul se fâche avec pierre, comment peut on empêcher paul d'accéder aux fichiers de pierre ? Il faut ouvrir une session en tant que Paul, aller dans les certificats…




et supprimer le certificat de pierre dans sa liste de certificats…

**N.B** : par défaut Pierre peut toujours modifier les documents déjà existant de pierre ? s'il les a déjà ouvert, leur champs liste DRF (liste des agents récupérateurs) inclus l'utilisateur Pierre comme personne habilité à les gérer.

Par contre, si maintenant Paul crée d'autres documents, ceux-ci sont inutilisables à Pierre

# Decryptage de Fichier EFS

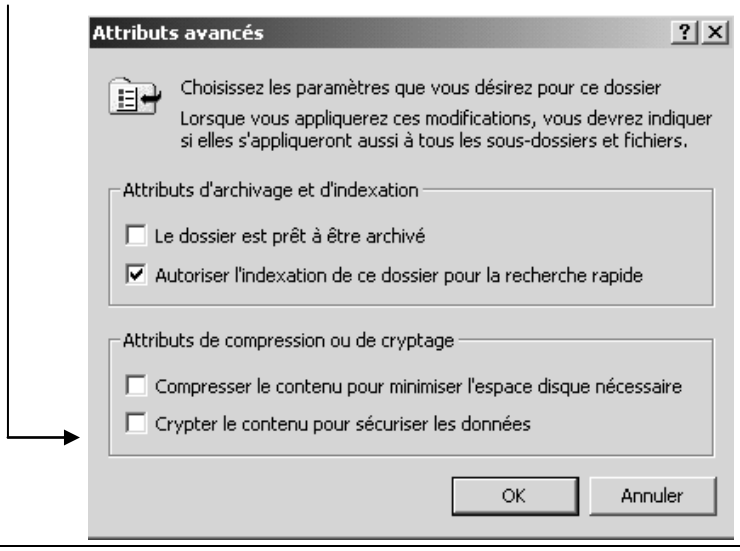
### Décryptage EFS et gestion des fichiers:

Donc un fichier crypté dans un domaine, pourra être "lu" sur n'importe quelle machine, à partir du moment où l'on ouvre une session du domaine… Si on se transfère sur une autre machine (autre domaine ou chez soi) il faut emporter aussi le certificat avec la clé privée, ou bien être repéré comme agent de récupération !

SI la clé privée est inutilisable, (utilisateur inexistant) l'agent de récupération peut ouvrir le fichier en utilisant sa propre clé cryptée. Si l'agent de récupération se trouve sur un autre ordinateur, il faut lui envoyer le fichier à décrypter, plutôt que l'agent vous envois sa propre clé cryptée…

### Décryptage d'un fichier

De manière générale, lors d'une manipulation, pour décrypter un fichier il suffit de décocher la case **crypter le contenu**… dans avancée



**Manipuler un fichier crypté**

Lorsque l'on effectue les manipulations sur les fichiers cryptés, de manière générale, le fichier est décrypté, modifié, puis recrypté sous sa nouvelle forme ou destination si cela est possible

| Ainsi lorsque l'on : | le chiffrage est |
|---|---|
| • Change le nom | maintenu |
| • Déplace Copie le fichier | maintenu si cible sur NTFS2000 |
| • Déplace vers un autre PC | si la cible accepte EFS, maintenu avec la clé publique de l'expediteur. L'ordinateur cible doit être approuvé pour la délégation (voir dans Utilisateur et Ordinateur Active Directory, dossier Computer, propriété de l'ordinateur sur lequel on veut effectuer le transfert, onglet Général, la case à cocher Approuver l'ordinateur pour la délégation |
| • Sauvegarde | avec l'outil backup maintenu |

## Commande en ligne cipher

On peut aussi utiliser une commande en ligne **cipher** dont l'aide est complète…

Sous sa forme la plus simple, cipher indique si les fichiers – dopssiers sont cryptés ou non

```
    Utilisé sans paramètres, CIPHER affiche l'état de chiffrement du répert.
actuel et des fichiers qu'il contient. Vous pouvez utiliser plusieurs noms de répert.
et des caractères génériques. Vous devez insérer des espaces entre plusieurs paramètres.
```

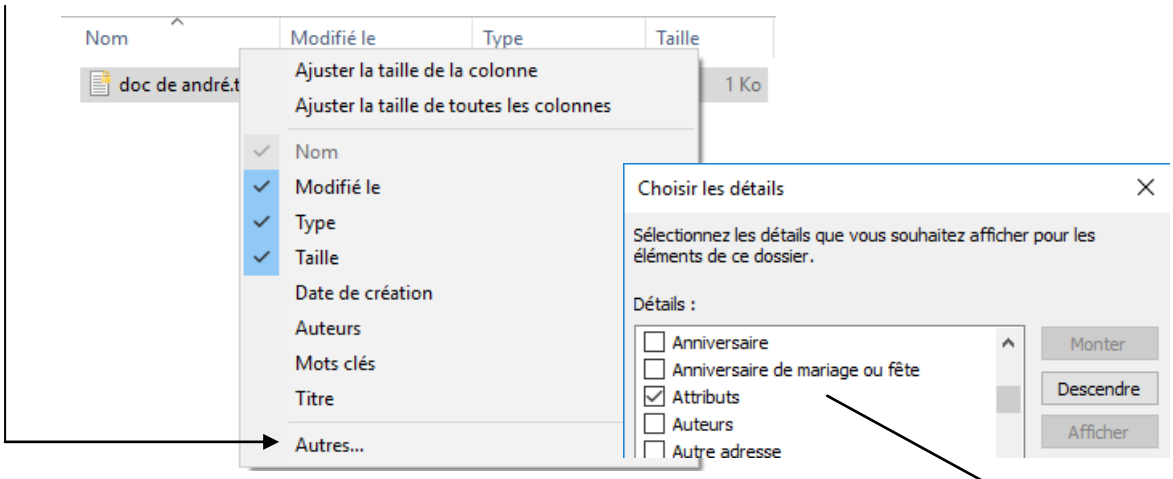
ainsi **cipher** donnerait:

**E** pour encrypté →

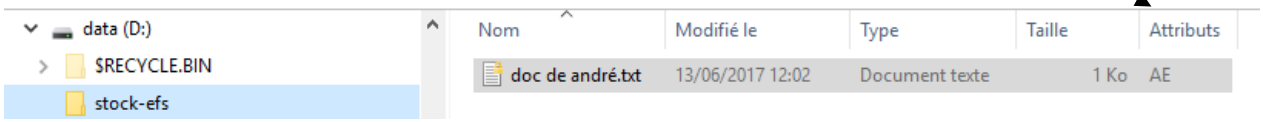
```
D:\stock-efs>cipher

 Liste de D:\stock-efs\
 Les nouveaux fichiers ajoutés à ce répertoire ne seront pas chiffrés.

E doc de andré.txt
```

**N.B** : **l'attribut E** peut apparaître aussi dans l'explorateur de fichier

Il faut demander dans l'explorateur sur les colonnes via le menu contextuel **Autres**, et demander **Attributs** :




Pour obtenir




# EFS – Sécuriser un poste autonome

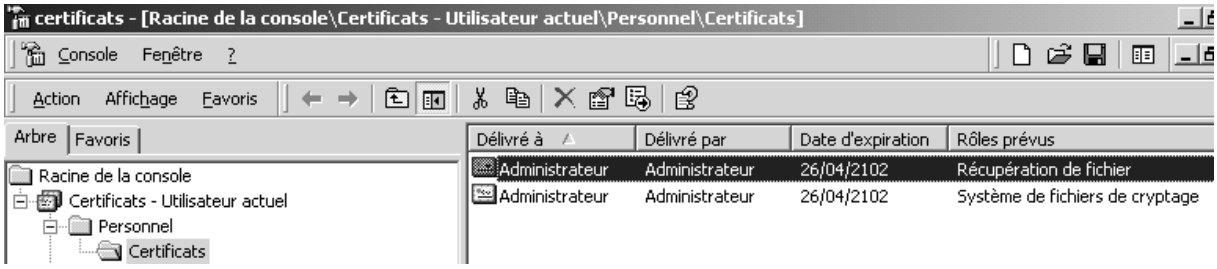
**Vol d'un poste (portable ?) :**

Si on a compris le principe de la sécurisation, on a compris que sur une machine, par défaut le Compte de l'Agent de récupération par défaut est celui de l'Administrateur de Domaine (dans le cas d'une machine faisant partie d'un domaine) ou le compte de l'Administrateur Local…

Pour sécuriser au maximum une machine autonome, il suffit d'exporter le certificat du compte de l'agent de récupération, en exportant sa clé privée en un endroit sûr ! A partir de la, l'agent de récupération n'existe plus !!!!! il faut pour le « reconstruire » que l'on importe le certificat et sa clé privée, ce qui suppose d'avoir accès aux fichiers ….

**Export du certificat de l'agent de récupération d'un poste autonome :**

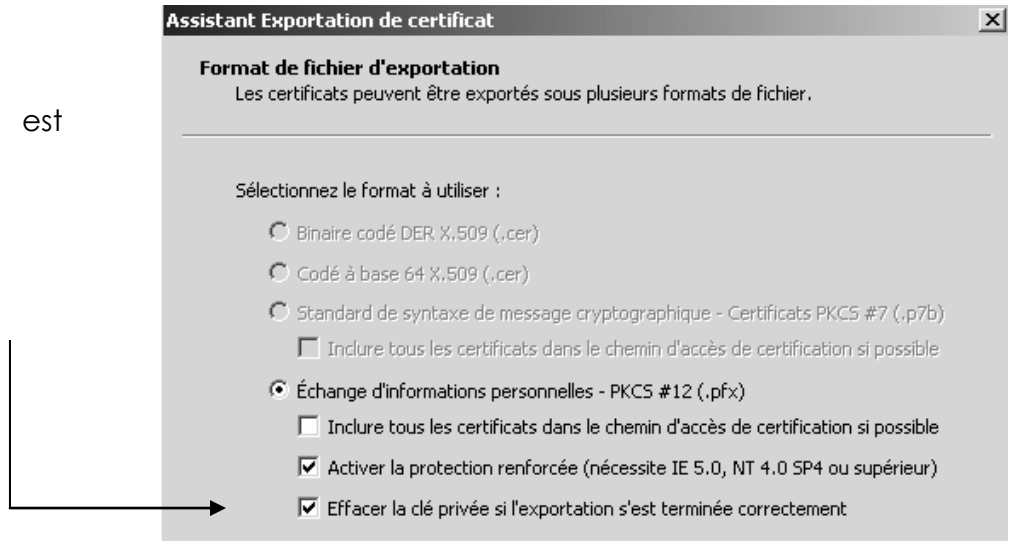
Le poste étant autonome (hors Domaine), il faut ouvrir une session en tant qu'administrateur local…




et demander d'exporter, dans l'assistant il faut indiquer alors


et surtout effacer la clé localement

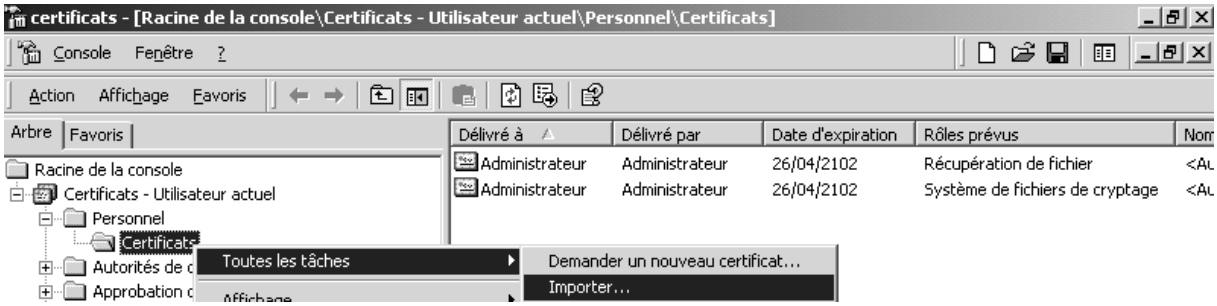
Sinon la clé est laissée localement




Vérifier que désormais l'administrateur n'est plus agent de récupération (ou plutôt il est **toujours agent de récupération**, mais ne **dispose plus de la clé privée**…)

**Import du certificat de l'agent de récupération d'un poste autonome :**

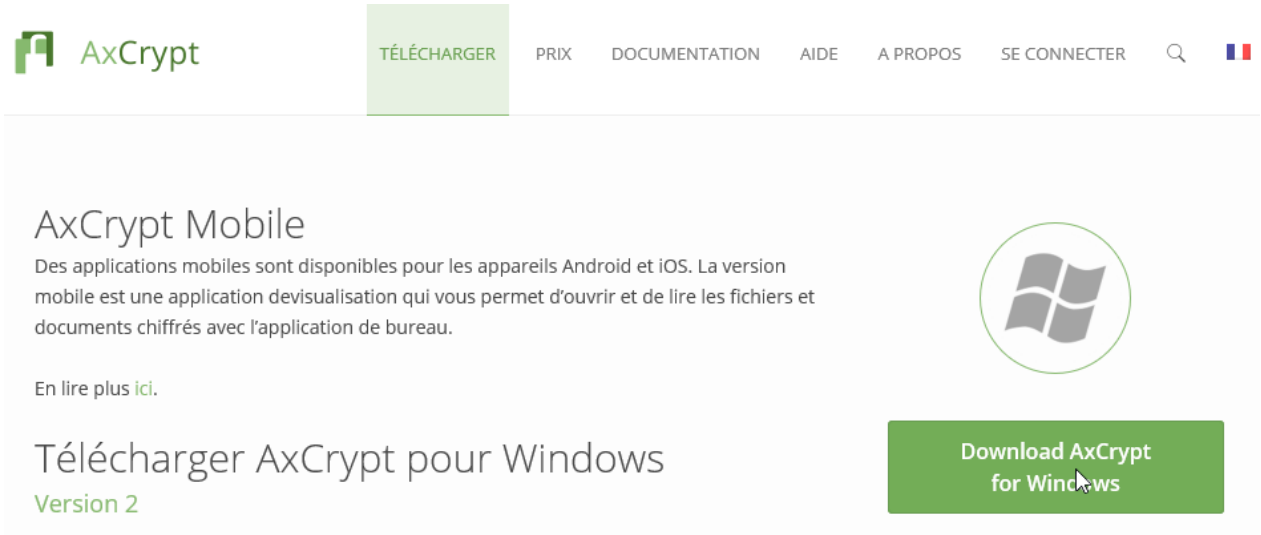
En cas de problème, on ouvre une session en tant qu'administrateur et on demande d'importer la clé



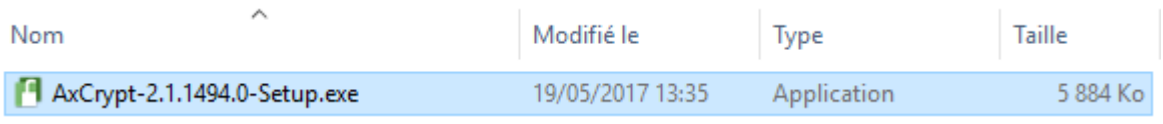

**N.B** : pour que cela marche, la clé ne suffit pas, il faut la réimporter dans le compte qui est agent de récupération par défaut… (ici administrateur local).

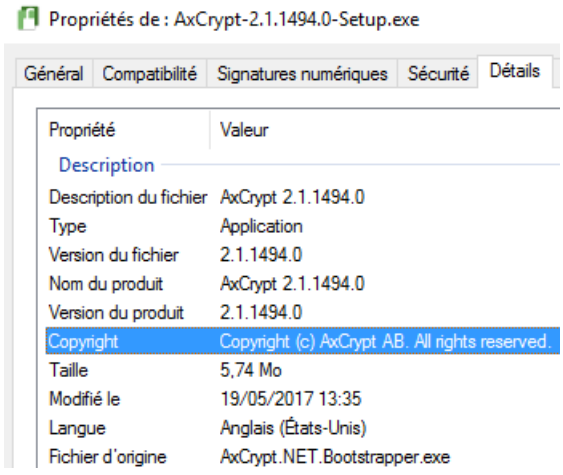
# Téléchargement de AxCrypt.net

Depuis le site **https://www.axcrypt.net/fr/download/**



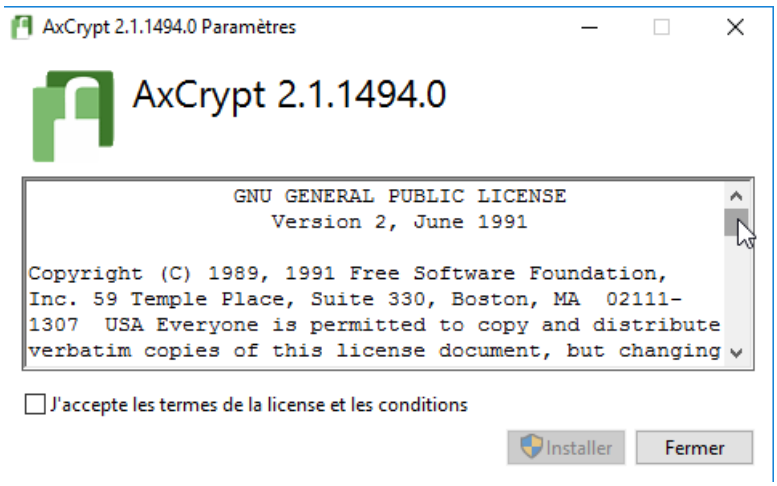

On récupère un executable



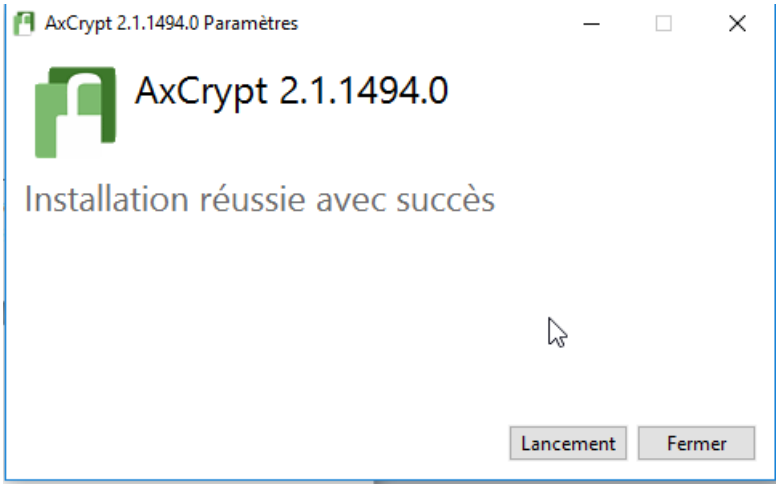



# Installer AxCrypt 2.0

On accepte la licence

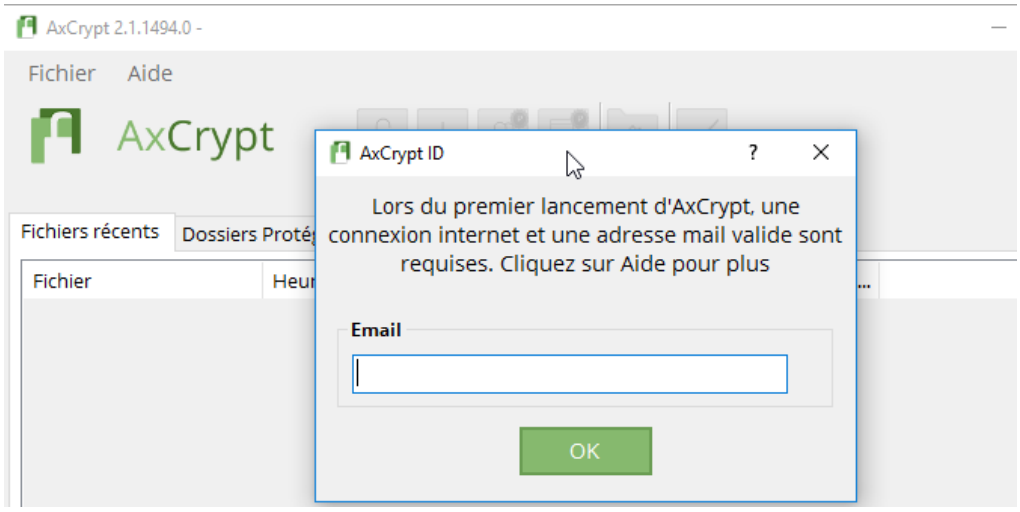



Pour obtenir

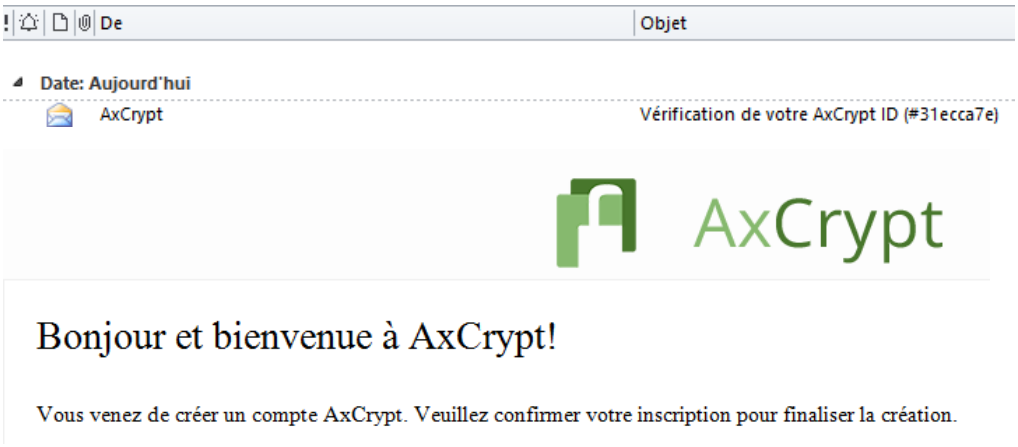



# 1° lancement - paramétrage

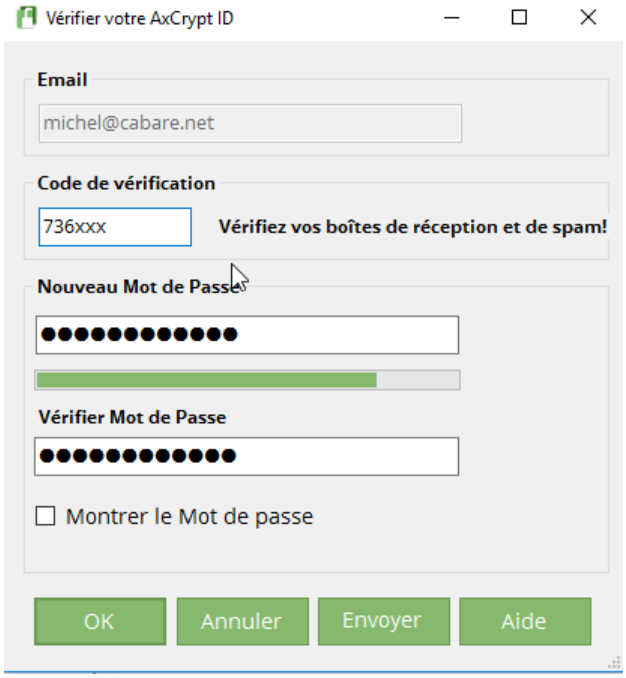
Au premier lancement une adresse mail est requise



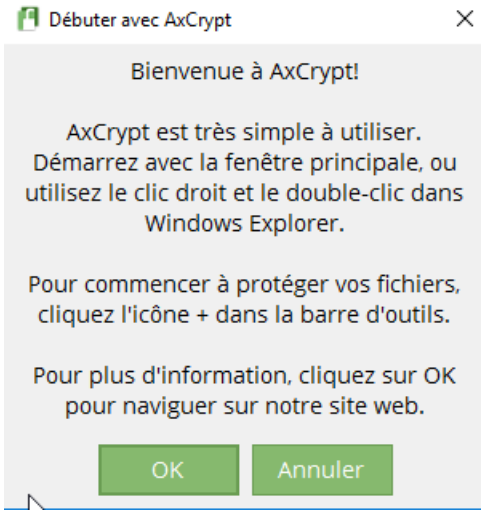
Cela permet de recevoir par email un code de vérification




Ce code permettra de réponde a la boite de dialogque suivante

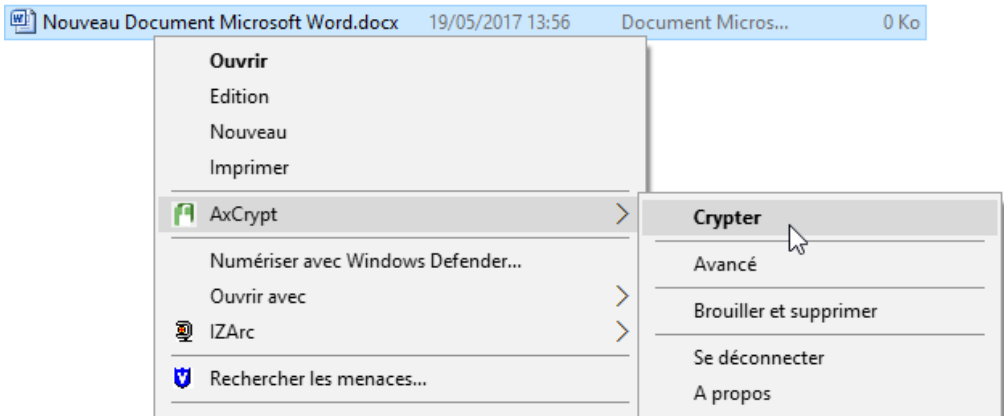



et de démarrer AxCrypt

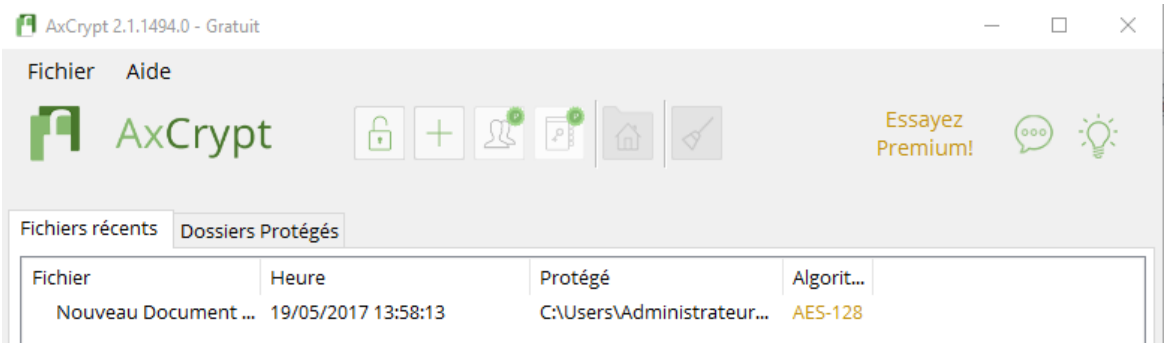


## lancer AxCrypt 2.0 via Clic Droit

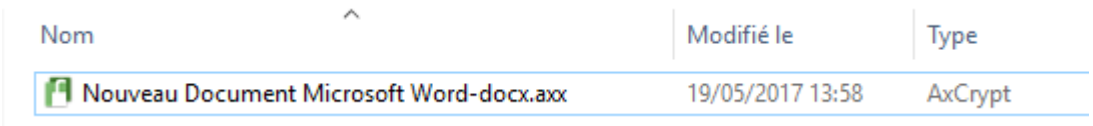
On se met sur un fichier à crypter




On demande Crypter, on rentre le mot de passe AxCrypt et on obtient



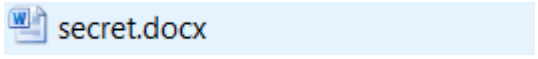

Et si on regarde le fichier, on retrouve en fait maintenant à la place de notre fichier Word un fichier crypté **.axx**




## Démonter le cryptage

Si on veut rendre de nouveau un fichier accessible, sans besoin de décryptage, alors il faut demander sur le fichier un clic contextuel, puis **AxCrypt / Décrypter**


Et il faut faire attention car on crée alors un nouveau document de toutes pièce (avec le nom du document d'origine)




**N.B** : attention si on a renommé le fichier crypté entre temps, cela peut provoquer quelques surprises..

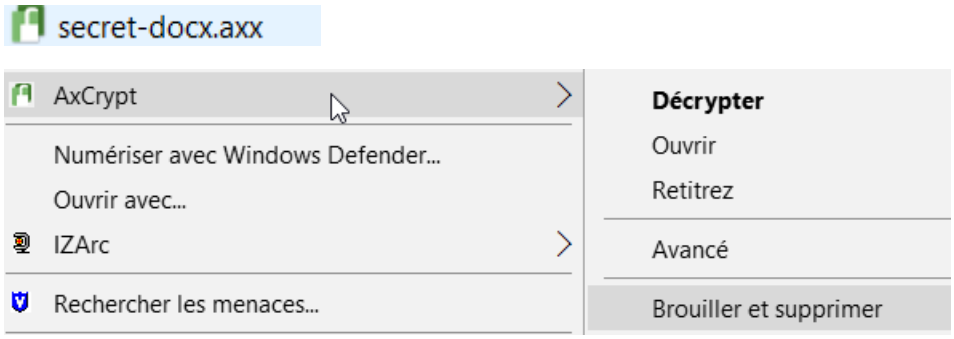
## Ouvrir le fichier sur une autre machine

Si le logiciel AxCrypt n'est pas installé (avec le même mot de passe paramétré) alors le fichier est inutilisable.

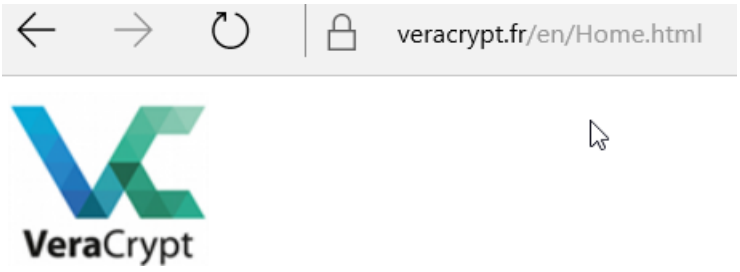
On ne peut l'ouvrir car aucune application ne reconnait le format de fichier **axx.**

## Supprimer définitivement un fichier crypté

Si on veut effacer définitivement un fichier alors il faut demander sur le fichier un clic contextuel, puis **AxCrypt / Brouiller et supprimer**



## Téléchargement de Veracrypt




Veracrypt s'installe de 2 façon :

- Installation complète sur une machine

  Toujours possible en environnement Windows, si on a les droits. On procède dans ce cas à une vrai installation du soft sur la machine, et ensuite on s'en sert

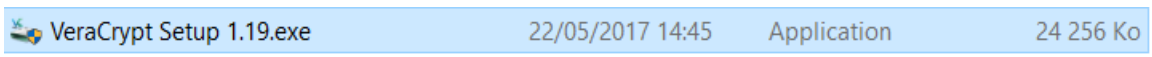
- Installation « portable mode»

  On procède dans ce cas à une unitlisation « à la volée » du fichier executable **veracrypt.exe**. Ce qui peut déclancher à chaque fois l'UAC, et parfois  poser quelques problemes de compatibilité car le driver utilisé n'est pas toujours compatible.

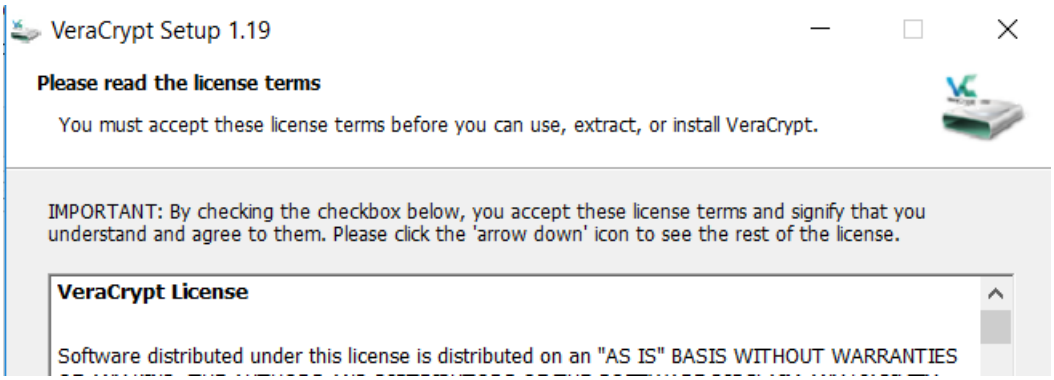
Veracrypt peut s'utiliser de 3 manières

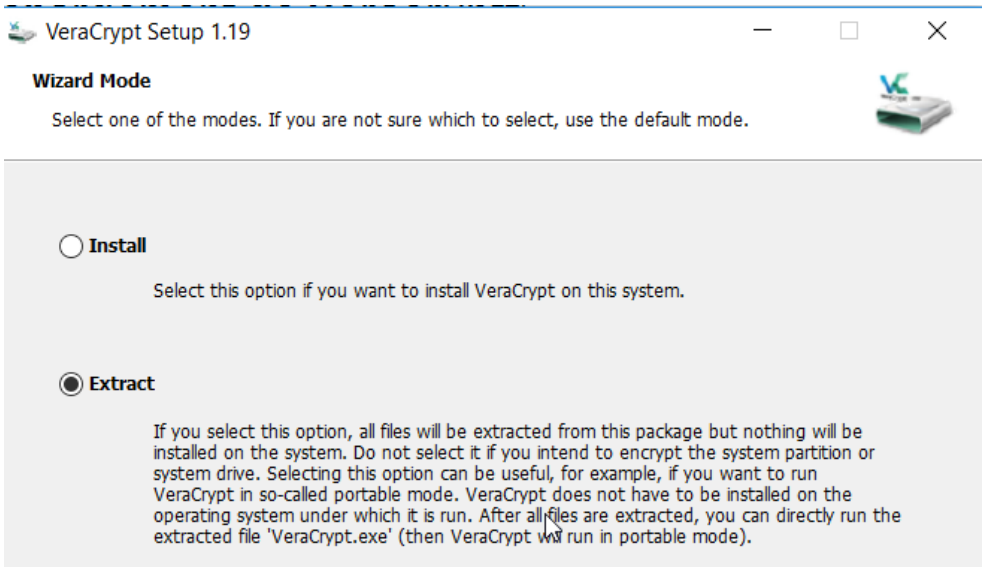
- Encryptage de la prtition système
- Encryptage d'une partition de données
- Création dans un fichier d'un disque dur virtuels cryptés (caché ou non)

## décompacter Veracrypt 1.19

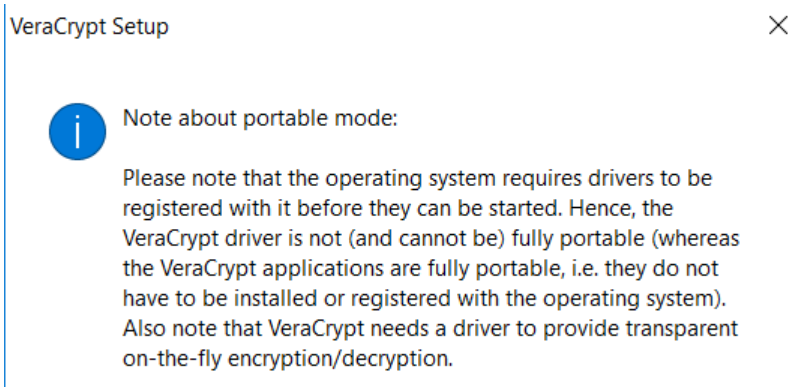



On accepte la licence

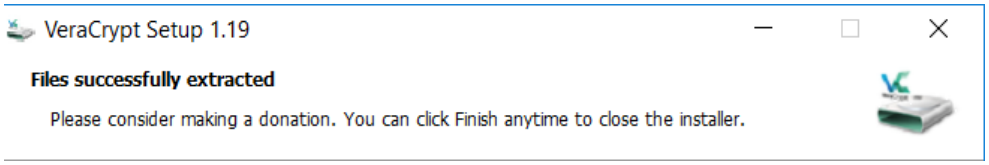



On a un warning



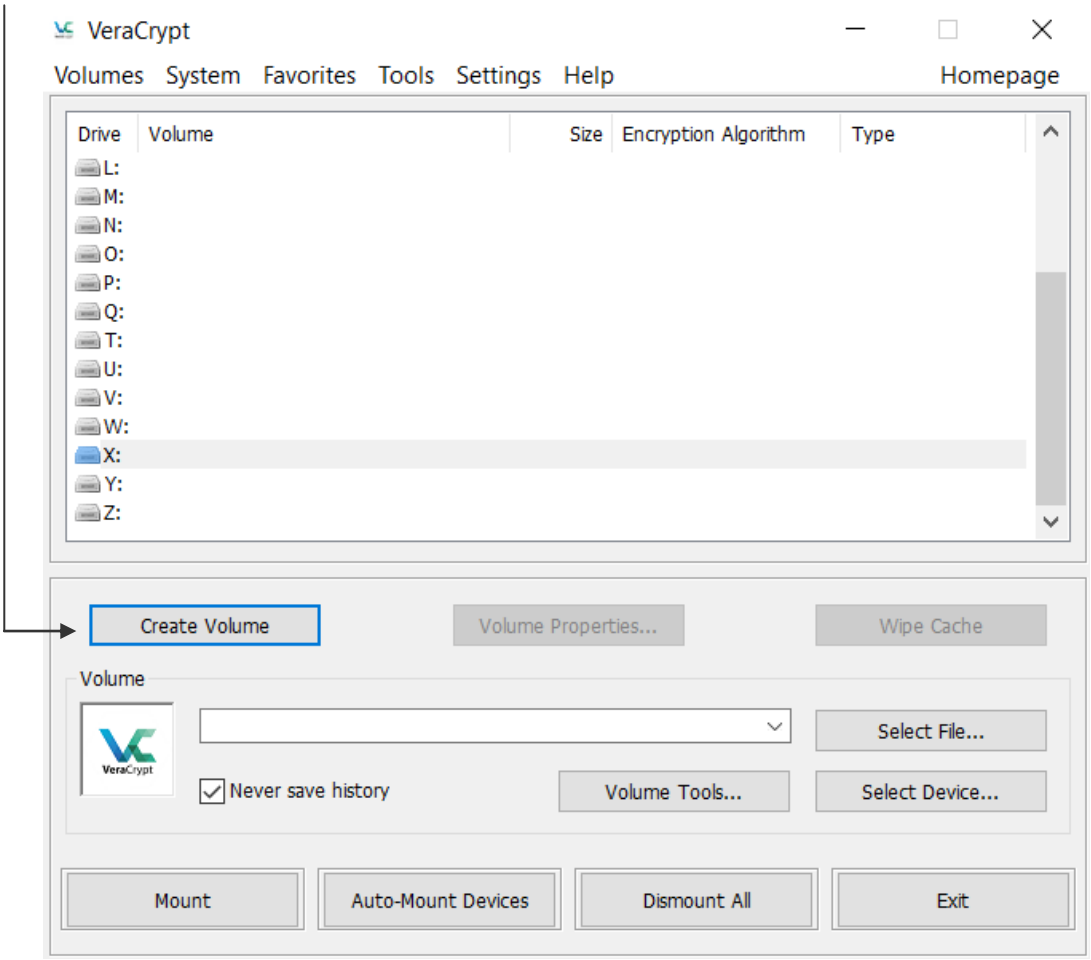

Et cela se termine



# Création container crypté – fichier disque virtuel

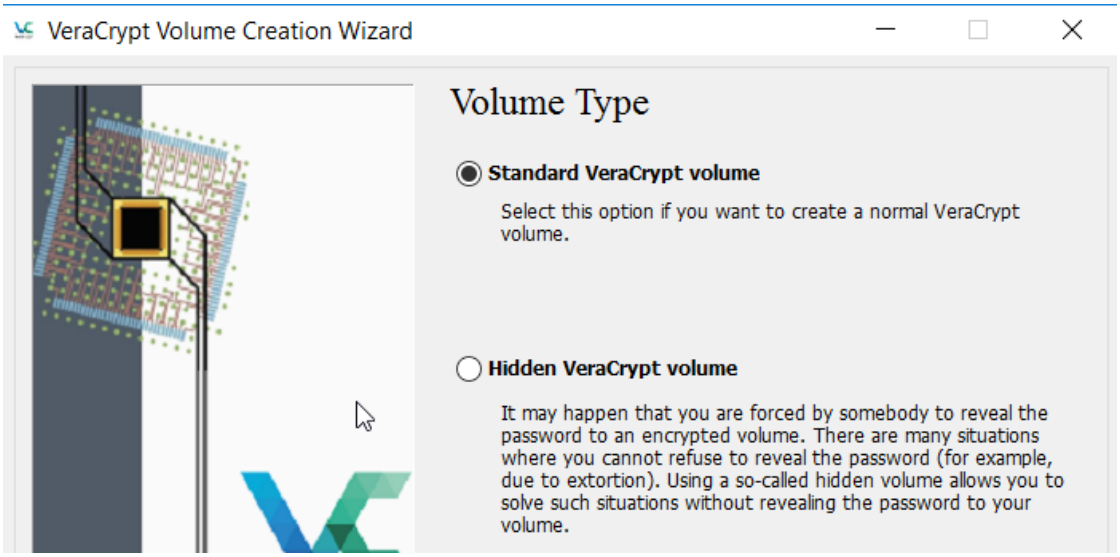
Utilisons Veracrypt pour créer un fichier « disque dur virtuels cryptés »

On lance veracrypt via **veracrypt.exe** et on demande **Create Volume**

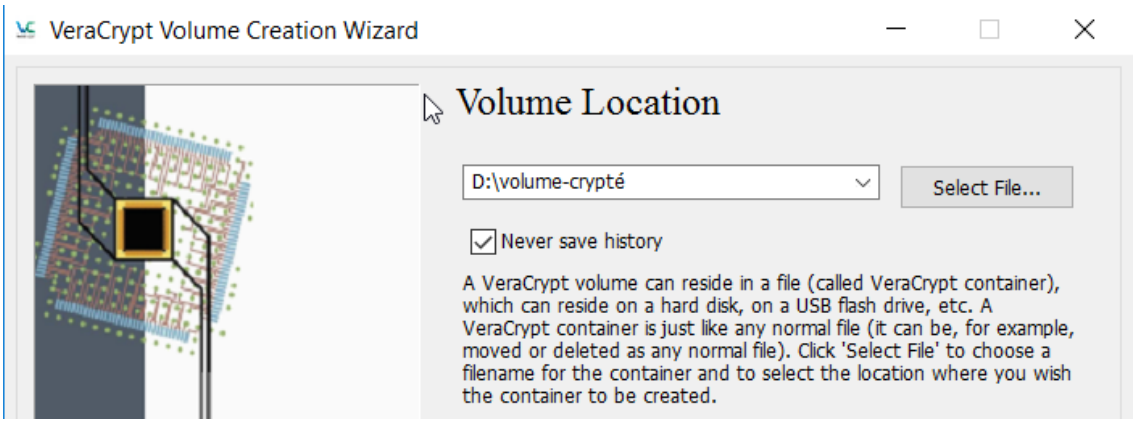



Le 1° choix est le bon « **create an ecrypted file container** »

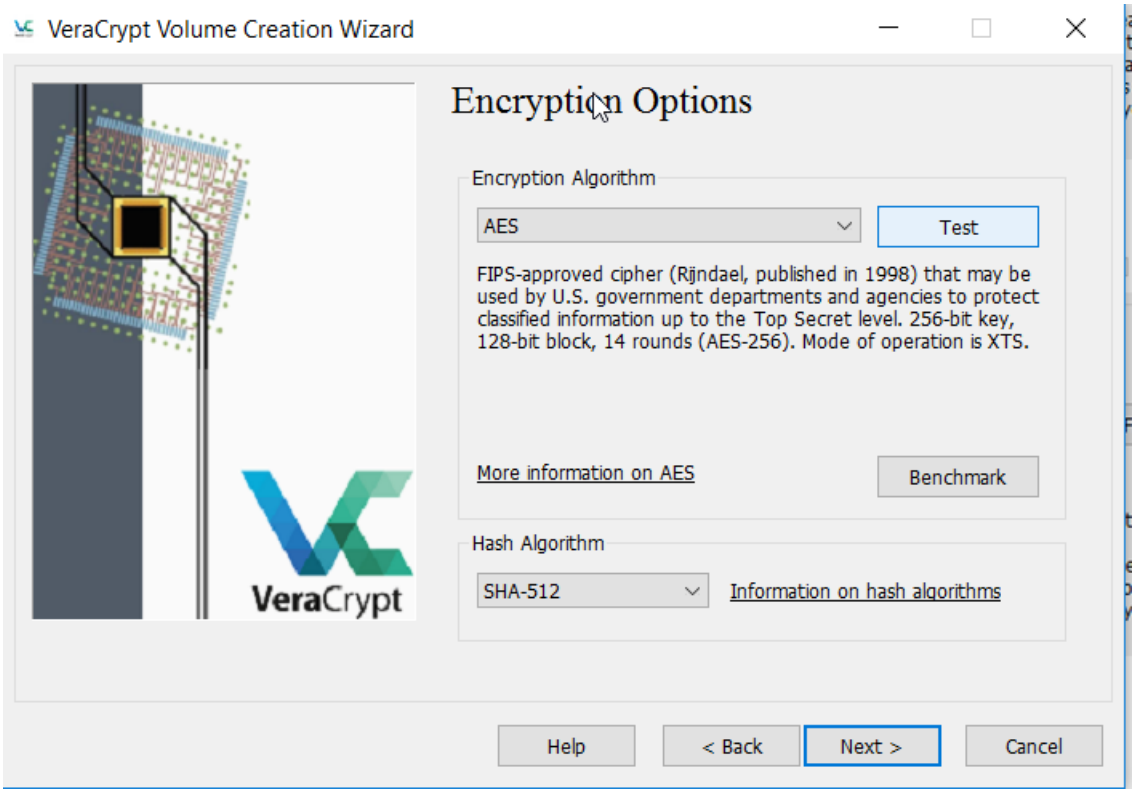
Non caché



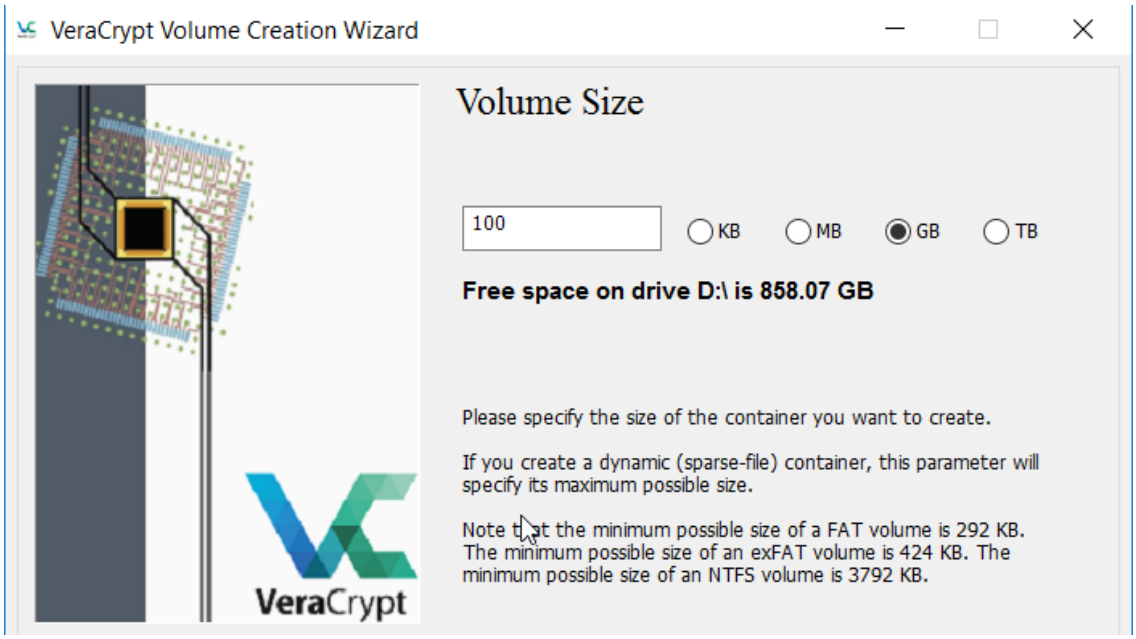

On donne un emplacement (ici ***D:***) et un nom au fichier (ici ***volume-crypté***)




On choisit un niveau de cryptage

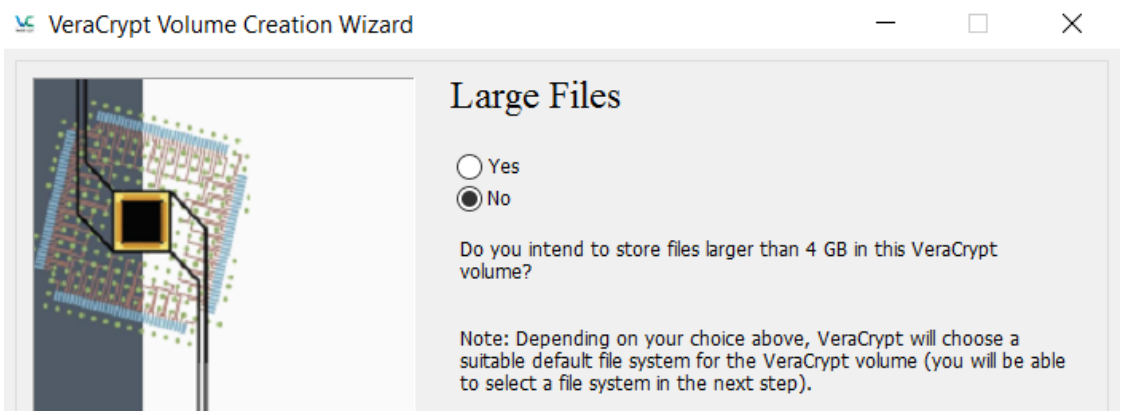


La taille du disque dynamique crypté



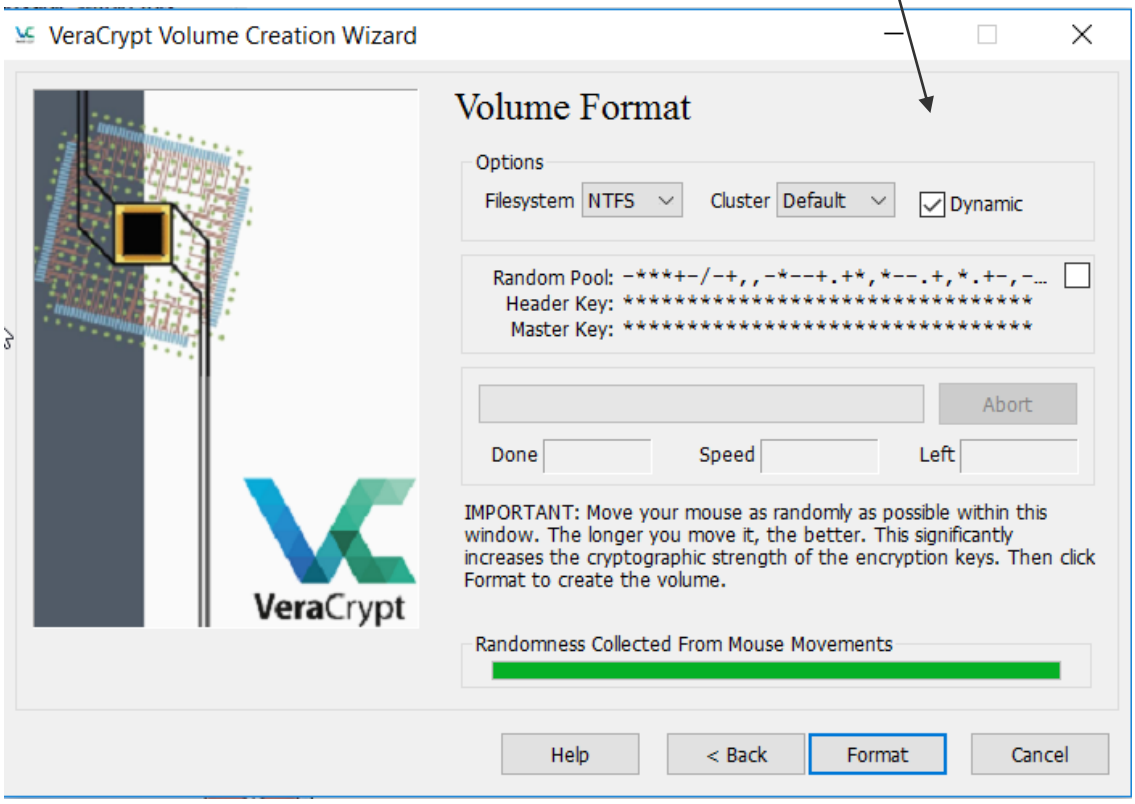

Le choix du mot de passe de 20 caractères minimum est CAPITAL ! (Min,Maj, lettre ,chiffre, caractère spéciaux)


On répond

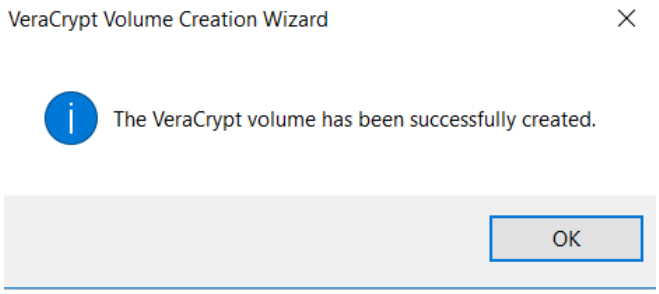


On demande un disque dynamique

On clique **Format** et on attend



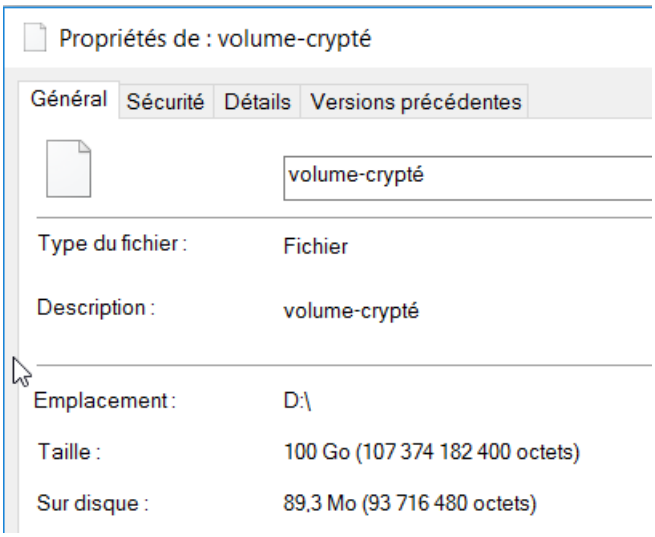

On à la confirmation



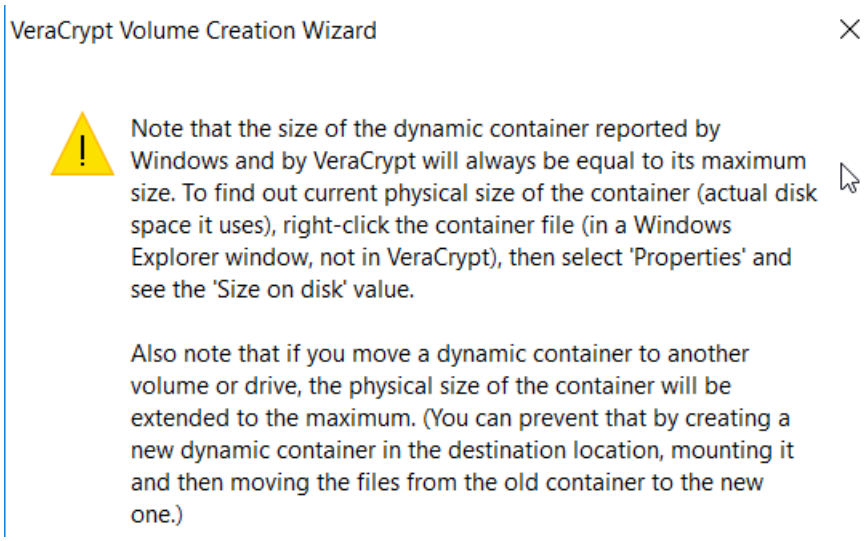

Puis une information importante sur la gestion de ces fichiers

- La taille réelle d'un volume crypté ne s'obtient que par ses propriétés, et non pas par son affichage direct dans Windows

  Ainsi un fichier qui annonce 100 Go Peut faire réellement 89 Mo



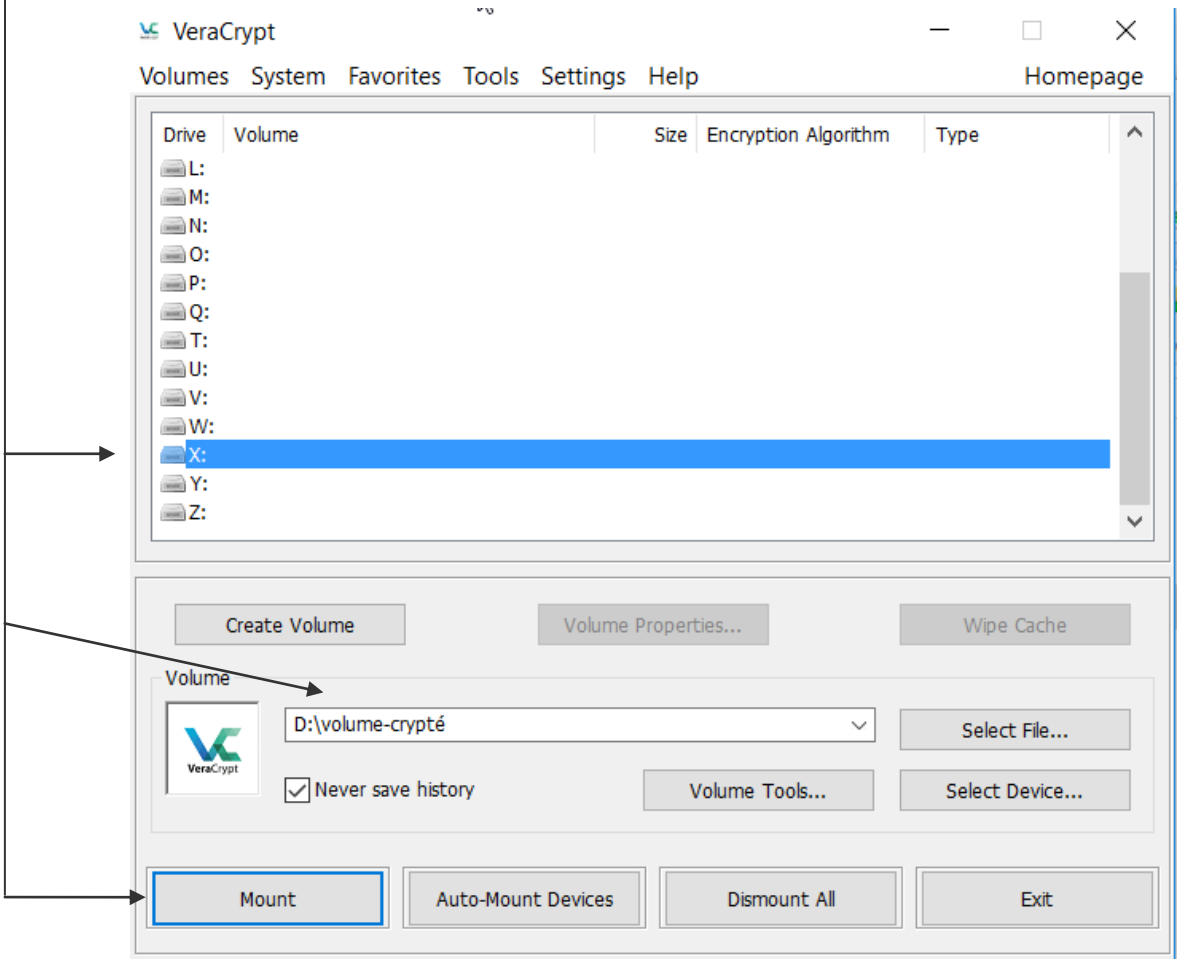
- Si on déplace un disque crypté sur un autre volume, il peut prendre sa taille réelle ! Si nécessaire on crée plutôt un autre volume Dynamique crypté, et ensuite on recopie les fichiers dedans !!!



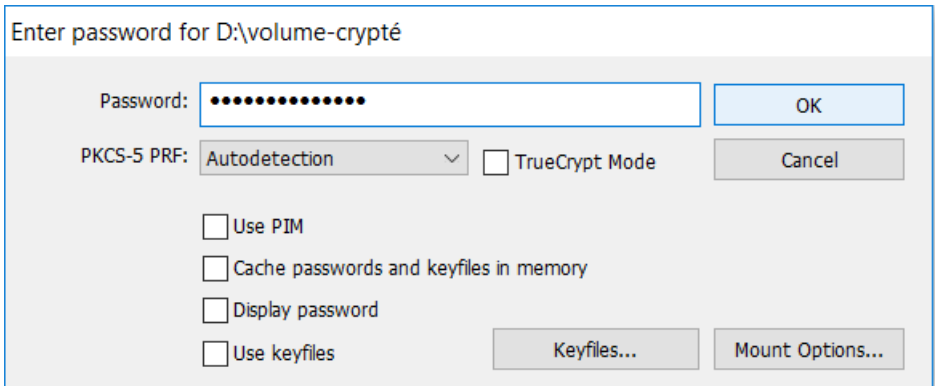

## Montage Volume sur container Veracrypt

Il faut maintenant monter un volume logique sur notre fichier-container précédamment construit. (c.a.d. le fichier contenant notre disque crypté)

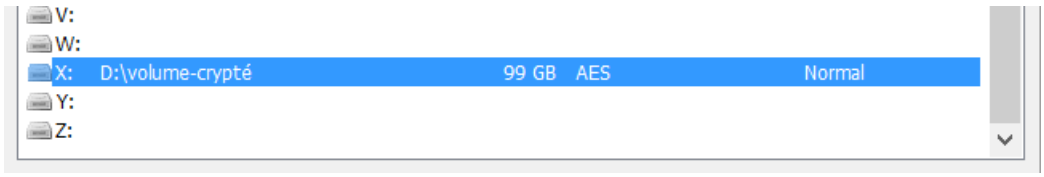
On choisit un lecteur (ici ***X*** : ) , le fichier-container (ici ***volume-crypté***), et on demande **Mount**



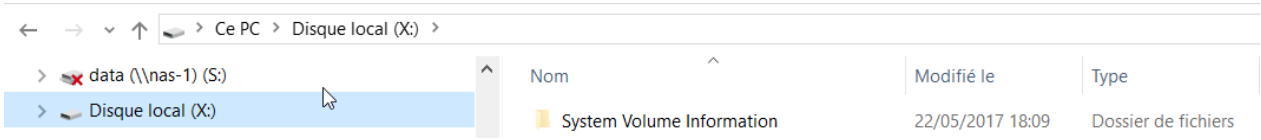
On donne le mot de passe




Et c'est fini



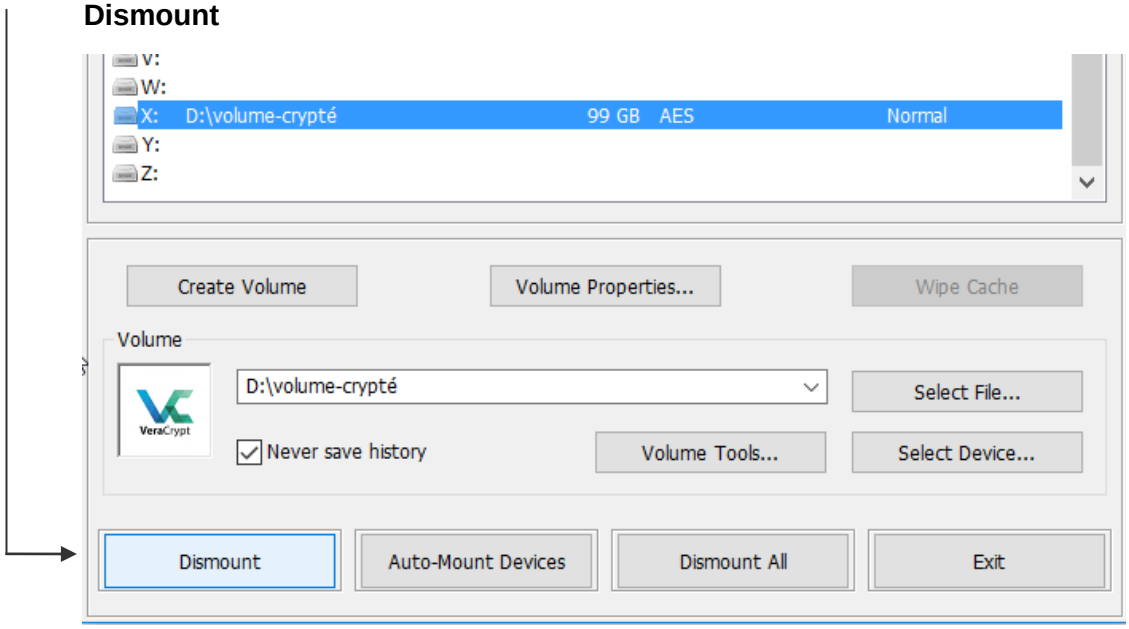

C'est un disque X : utilisable normalement




## Démontage Volume sur container Veracrypt

Le démontage se fait

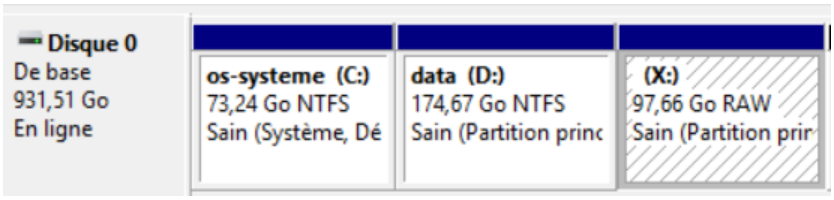
- automatiquement lors de l'arrêt machine,
- mais peut aussi être fait via le lacement de Veracrypt.exe, puis **Dismount**



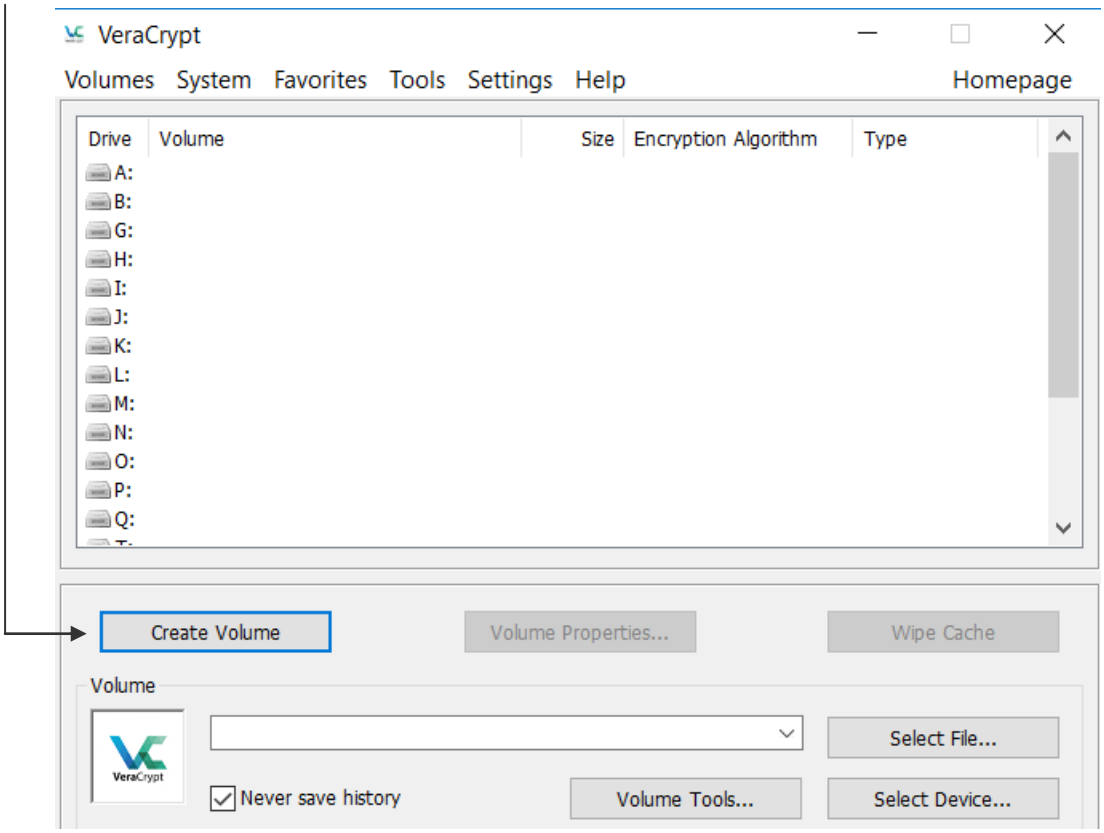
# Création Volume crypté

Il est possible de crypter un volume complet sur la machine

Par exemple un volume ici ***X*** : pré-existant

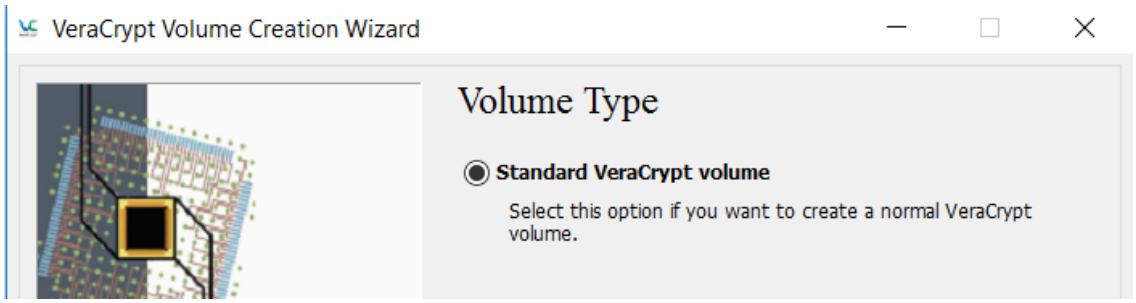



Il suffit dans l'assistant de demander de créer un volume **Create Volume**

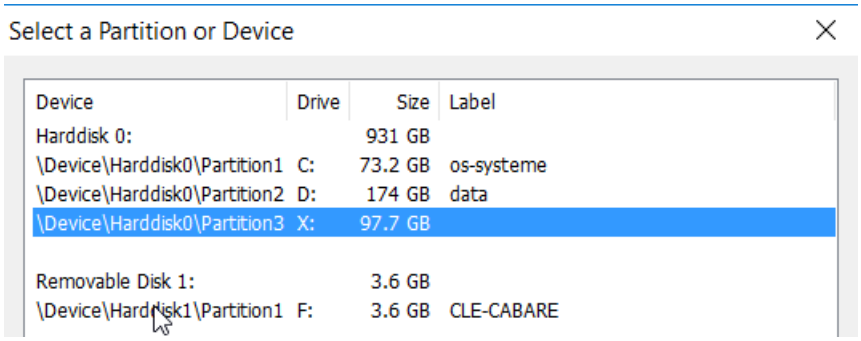


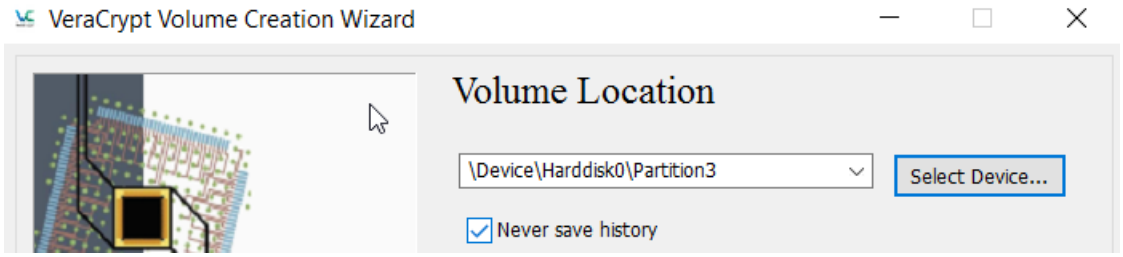

Puis **Encrypt a non system partition drive**

L'assistant ensuite est similaire…

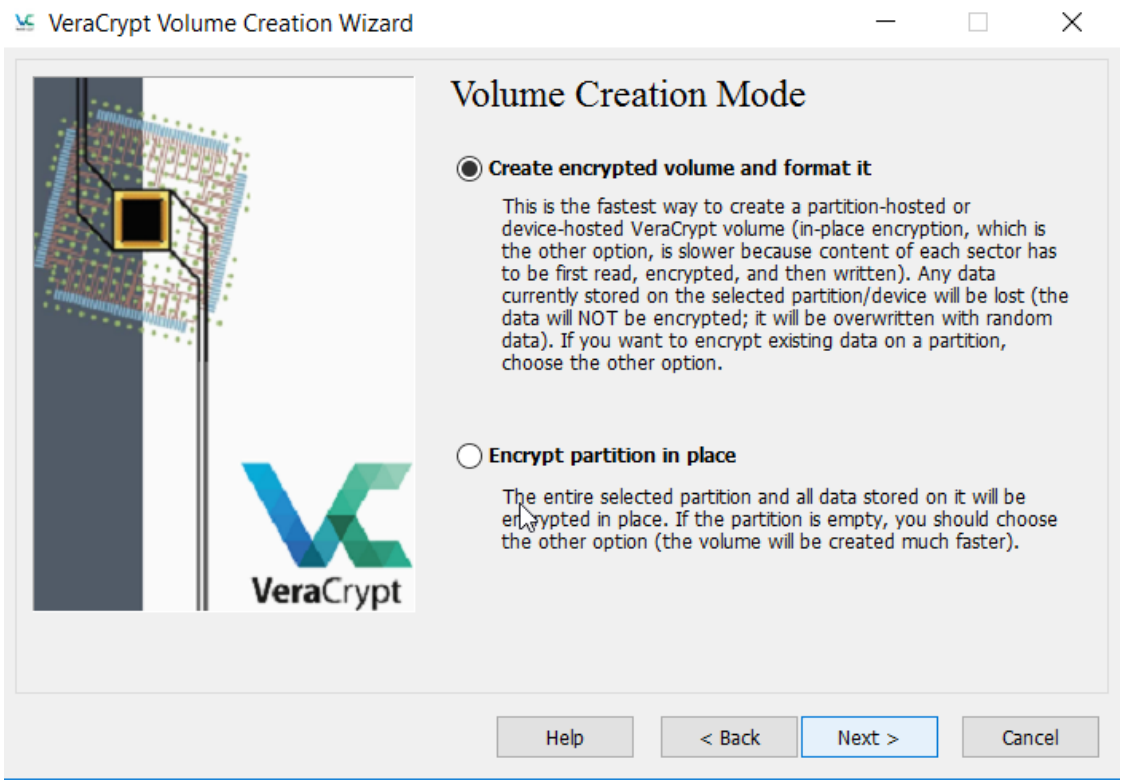



Encore faut il choisir la partiion – lecteur à crypter

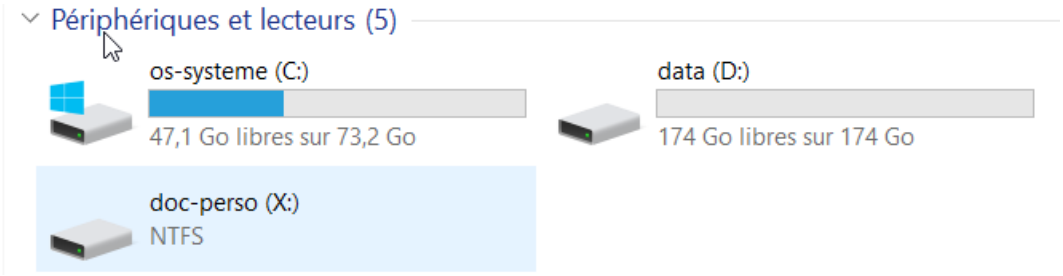


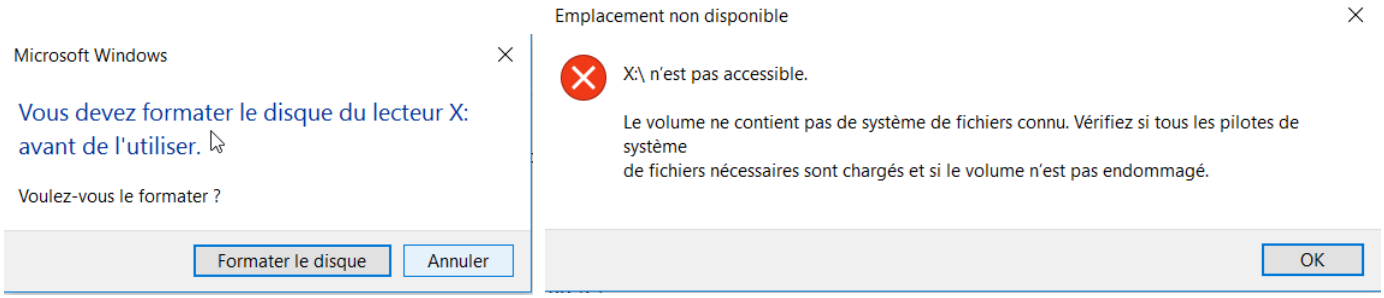




Et savoir si on veut la reformater ou convertir



Lorsque le lecteur est crée, tant qu'il n'est pas « monté », il apparaît comme non formaté…

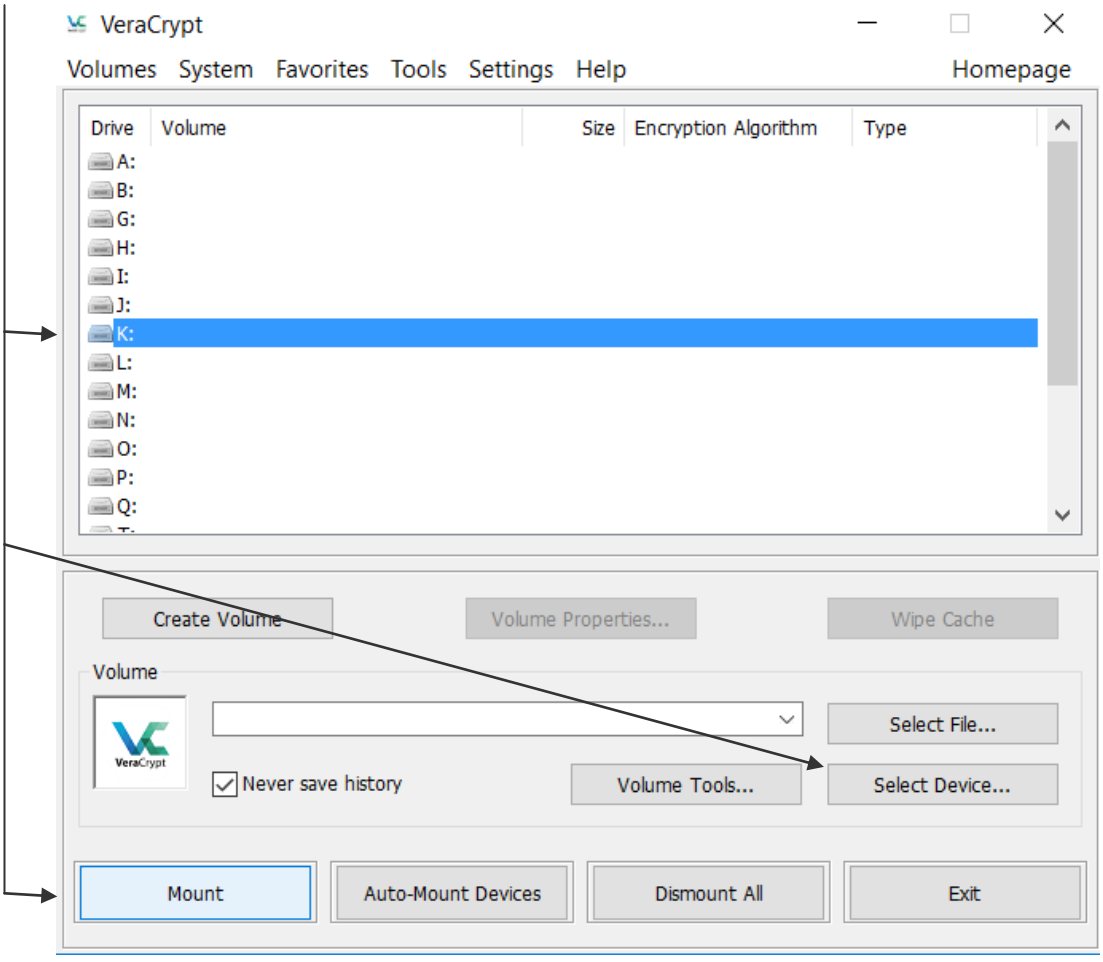


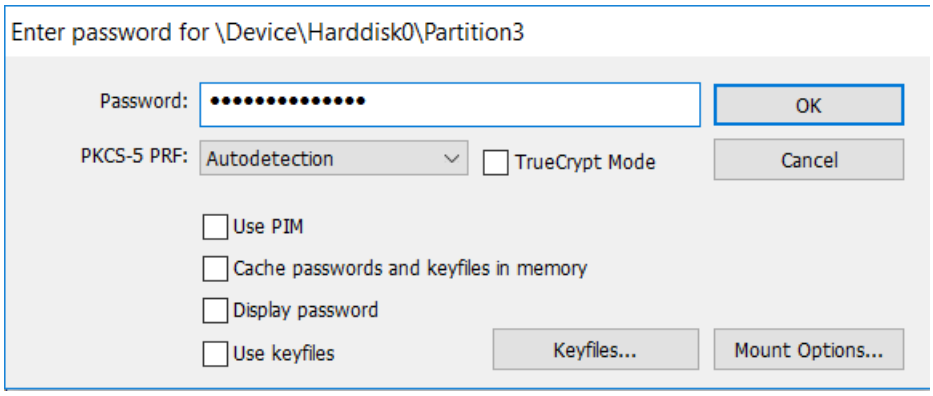

## Montage Volume crypté

**N.B** : on ne peut plus utiliser la lette X : qui est désormais plus disponible

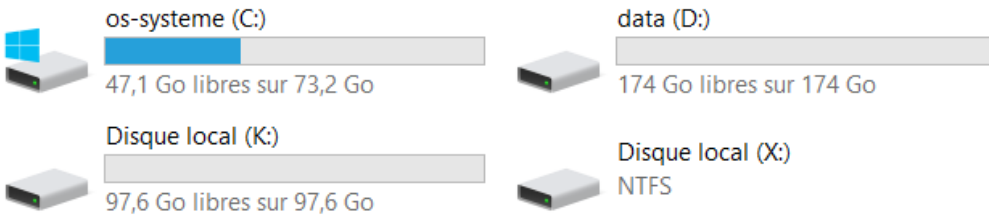
Donc par exemple **K :** puis **Select Device** puis **Mount**



Evidemment à ce moment là le mot de passe d'encryptage est demandé

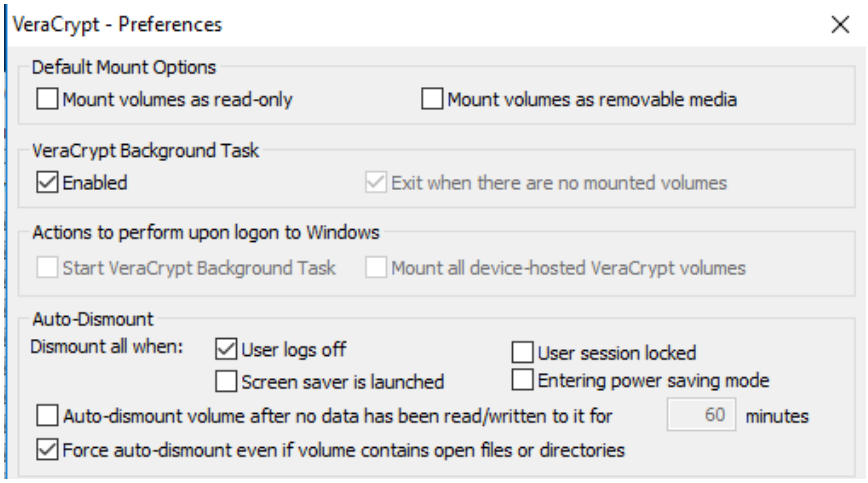



Et le disque K : devient utilisable




**N.B** : il sera démontable comme l'est un « fichier – container », c'est-à-dire soit via Veracrypt, soit automatiquement lors de l'arrêt du poste. Ou lors de l'arrêt de la session..

Les conditions de démontages peuvent être affinées via le menu veracrypt **Setting / preferences**



# SIGNATURE ET ENCRYPTAGE

## Différence entre une signature, et l'encryptage :

Une signature numérique, c'est un système qui assure que l'identité de l'expéditeur est bien celle supposée, est c'est un système qui vérifie que l'intégrité du message a été respectée (autrement dit on sait que le message nous vient bien de "untel" est qu'il n'a pas été modifié " en route")

Mais un message signé numériquement reste lisible par un éventuel pirate… (non modifiable, mais lisible…). Le cryptage, permet d'éviter à une tierce personne de lire le message au passage.

## Mécanisme de clé secrète (symétrique) :

La même clé est utilisée pour coder – décoder le fichier. Elle doit rester forcément secrète pour assurer la fiabilité. Environ 1000 fois plus rapide que la clé symétrique. EFS crypte les fichiers avec une clé symétrique, amis crypte cette clé symétrique avec une clé asymétrique dans les certificats…

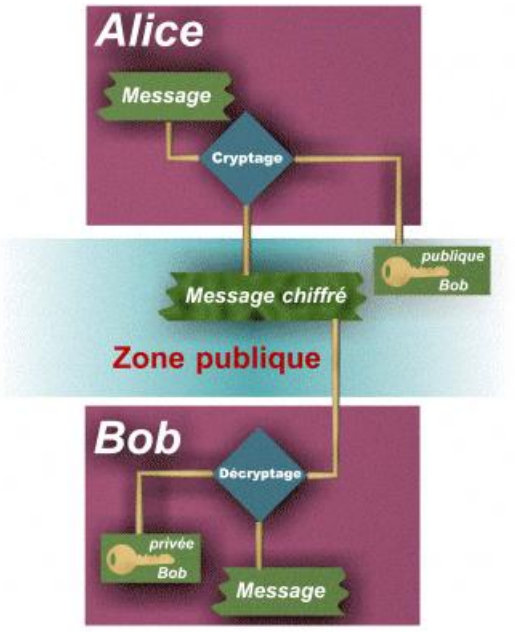
## Mécanisme de clé publique – clé privée (asymétrique):

En effet comme il est impossible de prévenir d'une interception frauduleuse des données il faut donc rendre ces informations illisibles par son intercepteur. Pour cela RSA a développé en 1977 un système de cryptage dit "a clé publique" qui répond parfaitement à ce besoin

Il est important de bien comprendre le fonctionnement d'un tel algorithme de cryptage : Cet algorithme fonctionne a l'aide de 2 clés une publique et une privée, tout phrase cryptée par la clé publique ne peut être décryptée que par la clé privée et vice-versa.

Grâce a ce principe, nous pouvons établir une transmission sécurisée (**voir schéma 1**) de Alice a Bob.

- Bob diffuse sa clé publique
- Alice crypte le message qu'elle désire envoyer à Bob avec cette clé.
- Bob pourra décrypter ce message. (et que lui)

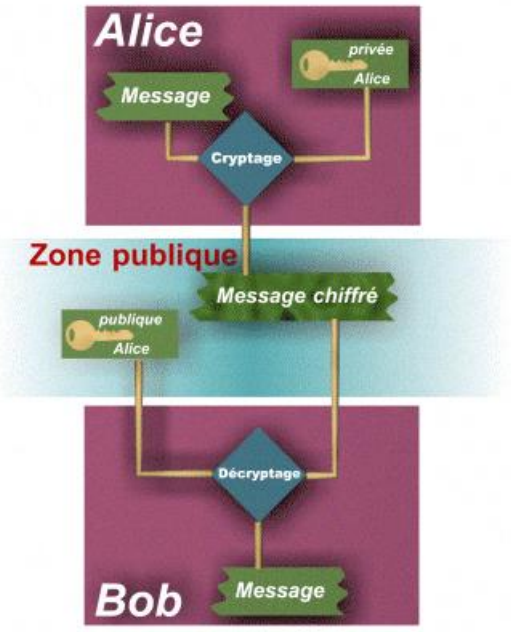


Transmission de Alice a Bob sécurisée

**N.B**: Bien sur il faut s'assurer que la clé présente dans la zone publique est bien celle de Bob pour cela il existe un **mécanisme de certificat**

Il est aussi possible grâce a ce principe de signer électroniquement (**voir schéma 2**) .

- En effet si Alice crypte un message avec sa clé privée
- seulement sa clé publique pourra le décrypter…
- Bob est donc sûr que c'est Alice qui a signé ce message.



Signature électronique d'Alice

# Mécanisme de Certificat :

Comme nous avons vu , il n'est pas possible de garantir qu'une clé présente dans la zone publique appartient bien à la personne que vous désirez contacter de manière sécurisée. Pour cela nous avons besoin d'un tiers de confiance qui va lui assurer l'appartenance des clés publiques

Donc il vous faut impérativement un certificat. Le format des certificats est défini par la norme **X509**.

Le procédé de certification est assez simple : vous contactez **un tiers** certificateur, vous lui transmettez vos coordonnées et votre clé publique et celui-ci, après s'être assuré de la validité de ces informations, vous donne une chaîne de caractère qui est en fait le certificat crypté par la clé privée de ce **tiers** ( sa signature électronique lisible avec sa clé publique…)

Donc pour récupérer votre clé publique, il faut que votre « contact » montre patte blanche auprès de **ce tiers**, afin que celui-ci le reconnaisse et lui délivre votre clé publique, c'est la rançon à payer pour être sûr de votre clé publique

Donc en fait les deux interlocuteurs doivent passer par un tiers auprès duquel il se sont enregistrés, de manière à être surs des provenances des clé publiques respectives. Ces deux interlocuteurs s'assurent de l'identité du tiers car toutes les informations qu'ils reçoivent de ce tiers (et notamment leurs clés publiques respectives) sont cryptée et décodables uniquement avec la … clé publique du tiers !