Sécuriser et authentifier les échanges d’information

Tamara Crétard - 14.05.2025

Sommaire:

1. Contexte

Dans ce TP, Alice et Bob doivent échanger des messages de manière sécurisée. Ils doivent s'assurer que les messages qu'ils reçoivent ont bien été envoyés par la bonne personne (authentification) et que personne d'autre ne peut accéder au contenu (confidentialité).

Pour ce travail nous utiliserons le message suivant, envoyé par Bob à Alice pour l'inviter à un rendez-vous secret :

« Bonjour Alice, rendez-vous ce soir à 19h34 devant la porte du 26 bis rue de la République. Assure-toi de ne pas être suivie. Bob »

Également, différentes méthodes seront utilisées: le chiffrement symétrique et asymétrique, la signature numérique, ainsi que la méthode hybride PGP pour le transfert de fichiers volumineux.

Chaque étape sera détaillée avec des explications claires et des captures d'écran, afin de montrer la compréhension des principes de sécurité et d'authentification dans les échanges d'information.

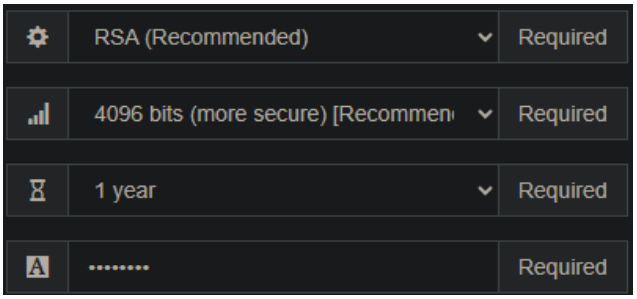
2. Etape 0 : génération des clés et mise à disposition des clés publiques

Afin de générer les clés publiques et privées d'Alice et Bob, il faut d'abord se rendre sur le site https://pgp.craeckor.ch/.

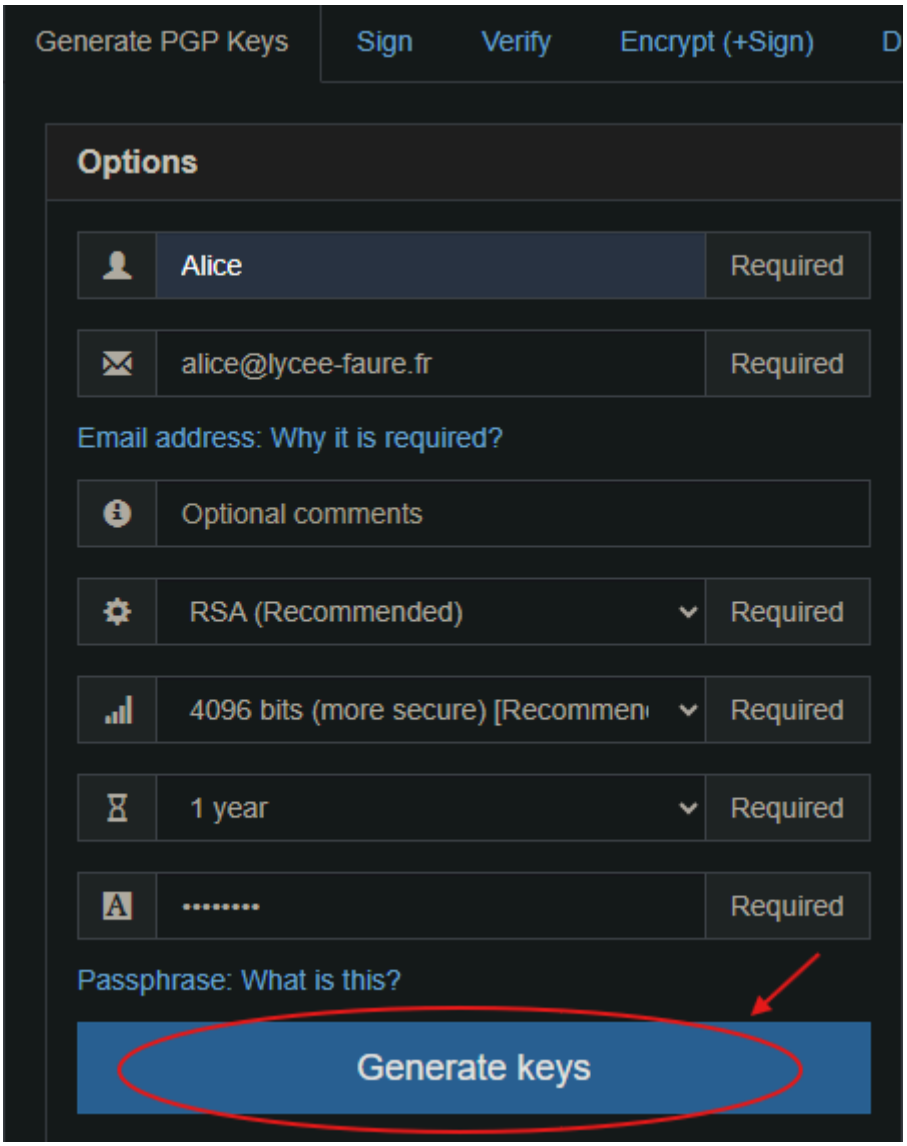
2.1. Génération des clés pour Alice

D'abord, nous générons les clés d'Alice. Pour cela, dans l'onglet "Generate PGP Keys", il faut entrer les informations concernant le créateur:

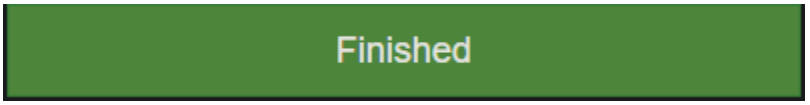
Puis, il faut définir l'algorithme de chiffrement asymétrique utilisé, la longueur des clés, la durée de validité des clés ainsi qu'une passphrase permettant d'ajouter une couche supplémentaire de sécurité aux clés:



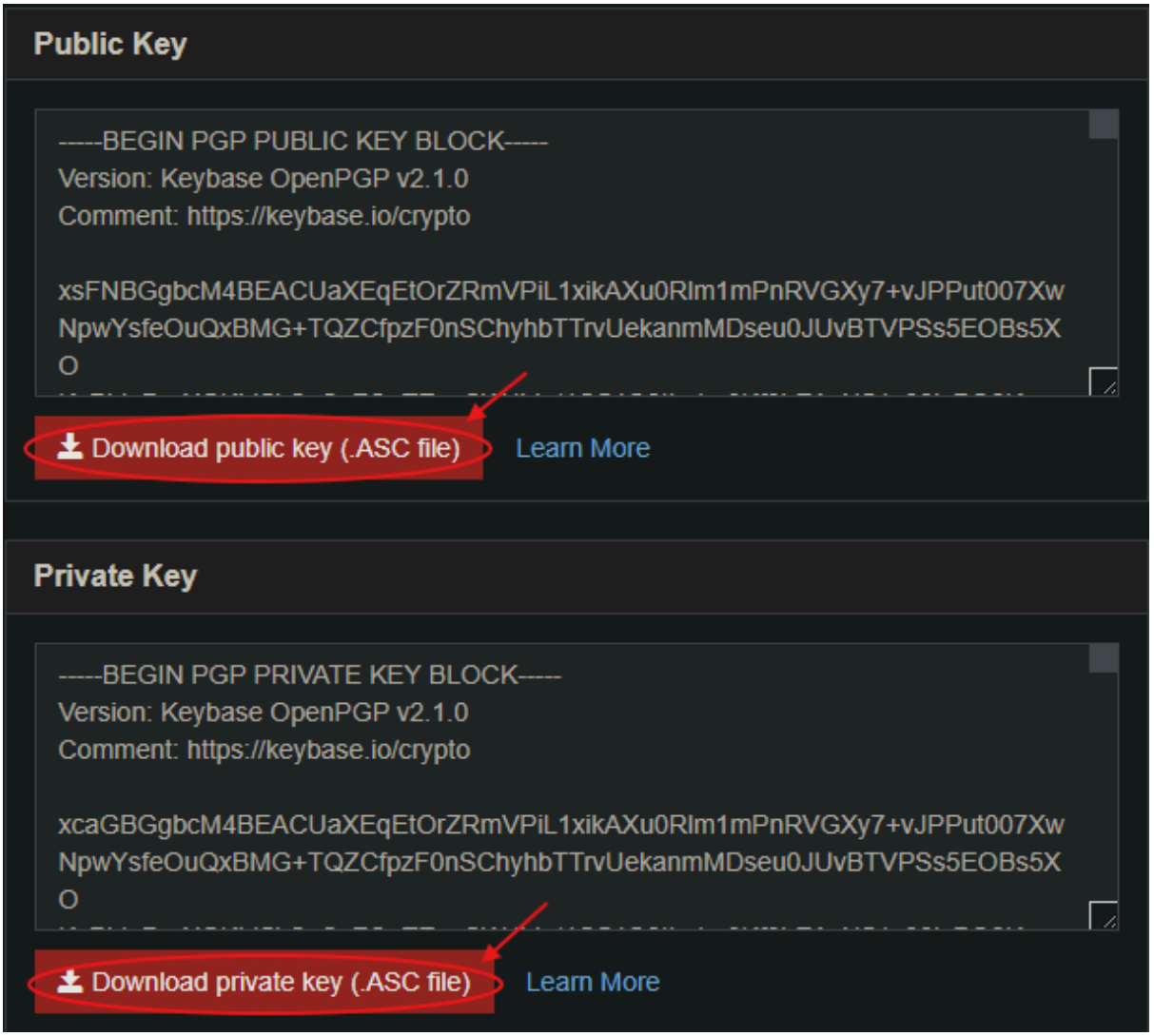
Une fois toutes ces informations entrées, il suffit de cliquer sur le bouton "Generate keys":



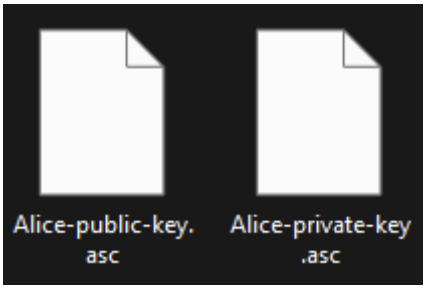
Une fois la génération des clés faite, le bouton change de couleur et affiche “Finished”:



Dans la partie droite de l’écran, on peut maintenant télécharger les clés publiques et privées en cliquant sur les boutons “Download public key” et “Download private key”:



Enfin, on renomme ces fichier pour les reconnaître plus facilement:



2.2. Génération des clés pour Bob

Pour régénérer les clés publique et privée de Bob, il suffit de suivre les mêmes étapes que pour celles d'Alice.

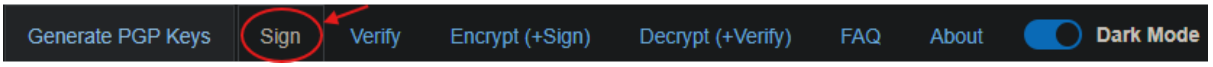
3. Etape 1 : signer et vérifier un message

Lors de cette étape, Bob va signer son message et Alice va pouvoir le vérifier.

3.1. Signature du message par Bob

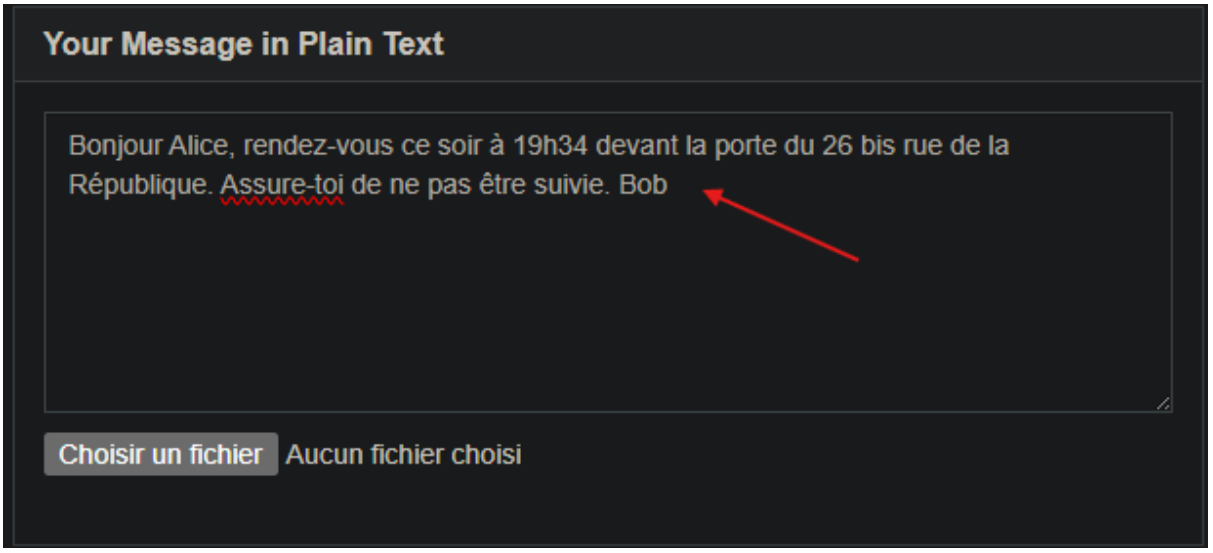
Pour signer un message et qu'Alice puisse bien reconnaître que c'est Bob qui lui a envoyé ce message, ce dernier doit utiliser sa propre clé privée.

Pour effectuer cette signature, il faut dans un premier temps se rendre dans l'onglet "Sign":

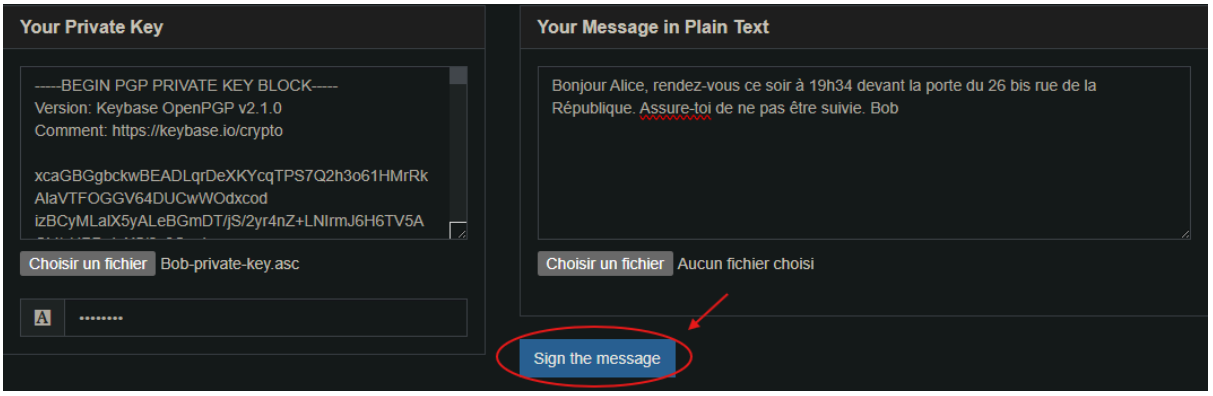


Dans la partie gauche de l'écran, on charge le fichier contenant la clé privée de Bob et on entre la passphrase choisie lors de la génération des clés:

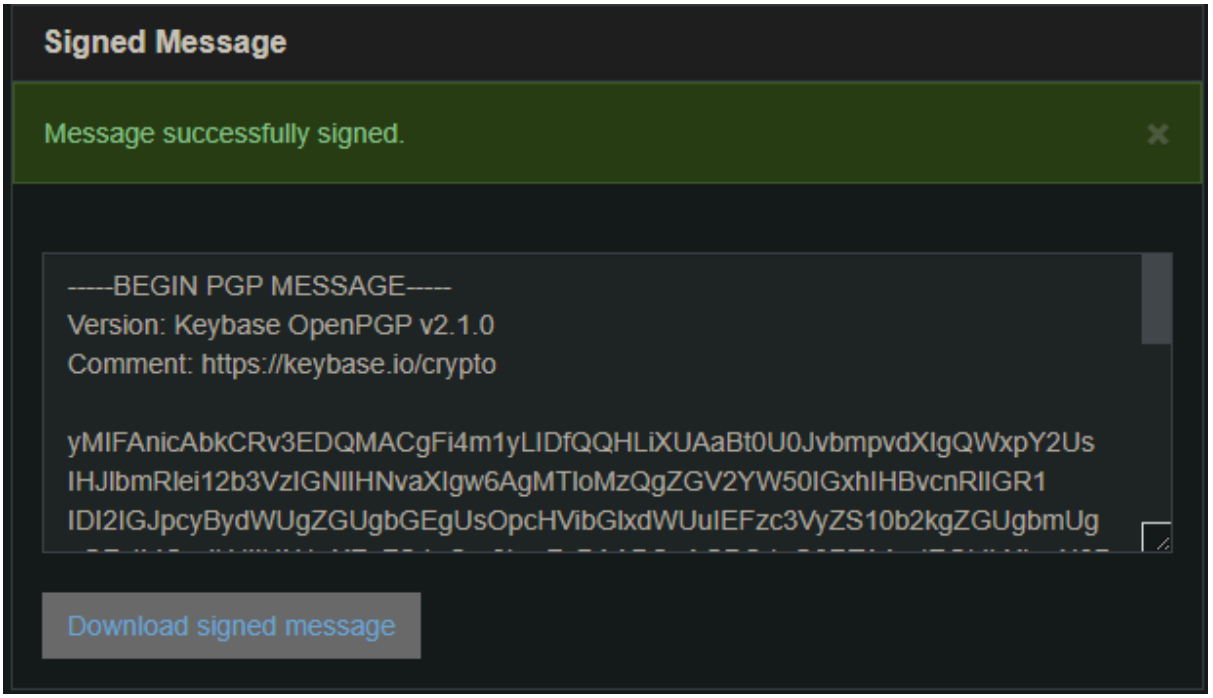
Ensuite, dans la partie droite, on entre le message à signer:



Enfin, on clique sur le bouton “Sign the message”:

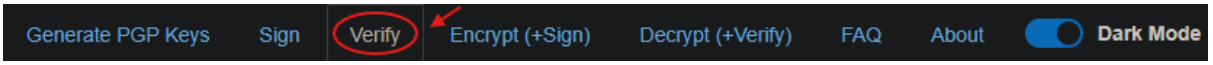


On obtient alors le message signé que l'on peut télécharger:



3.2. Vérification de la signature par Alice

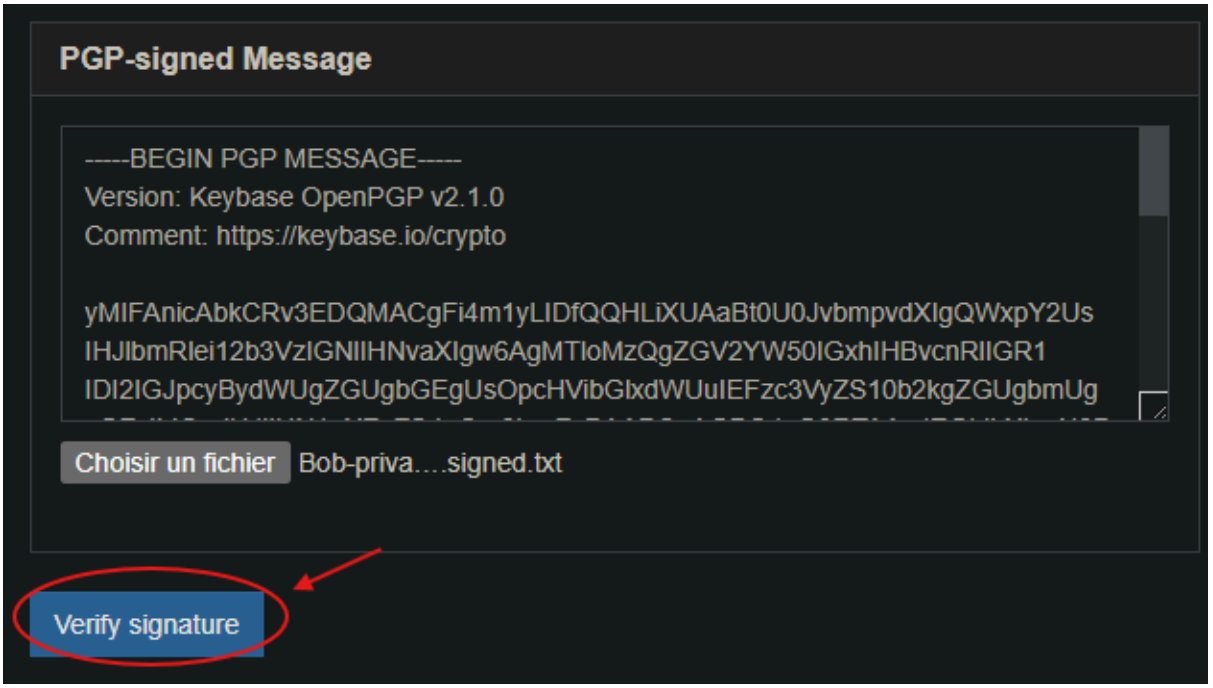
Pour qu'Alice puisse prouver que c'est bien Bob qui lui a écrit, on va dans l'onglet "Verify":



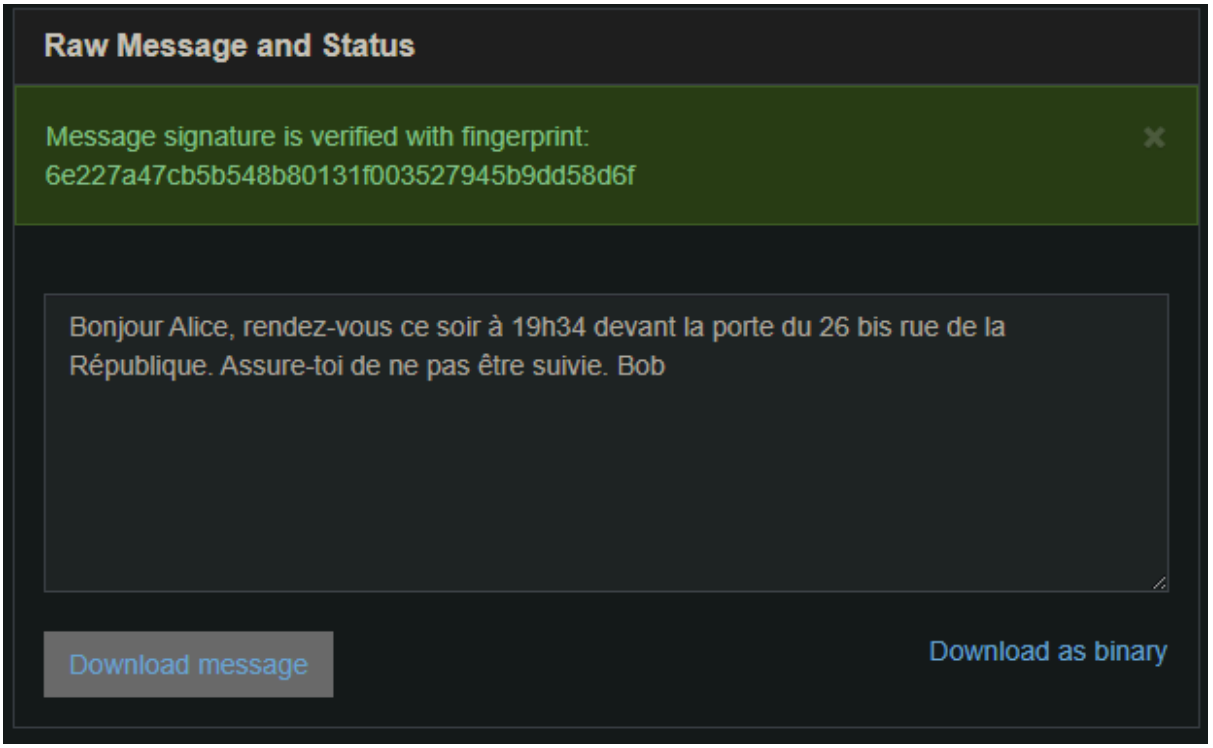
Dans la partie gauche de l'écran, on charge le fichier contenant la clé publique de Bob, considérant qu'il lui a envoyé auparavant:

Ensuite, dans la partie droite, on charge le message signé par Bob:

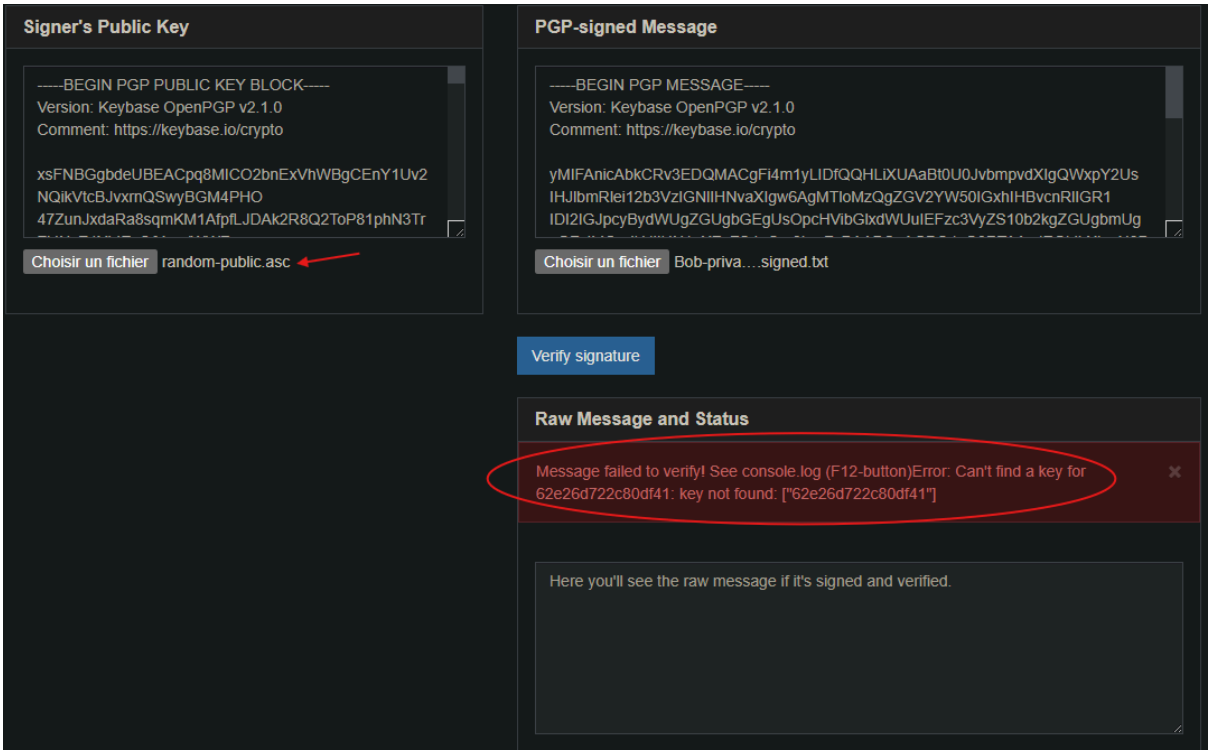
Enfin, on clique sur le bouton “Verify signature”:



On obtient alors le message avec la vérification de la signature:



Si l'on essaie de vérifier ce même message avec une clé publique autre que celle de Bob, un message d'erreur s'affiche:



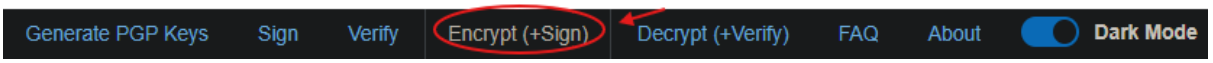
Cela montre que la vérification de la signature ne peut fonctionner qu'avec la clé publique correspondant à la clé privée qui a servi à signer le message. De cette manière, seule la clé publique de Bob permet de valider que c'est bien lui l'auteur du message. Si on utilise une autre clé publique, la vérification ne fonctionne pas, ce qui garantit l'authenticité de l'expéditeur et protège contre les usurpations d'identité.

4. Etape 2 : chiffrer un message

Lors de cette étape, Bob va simplement chiffrer un message et Alice va le déchiffrer.

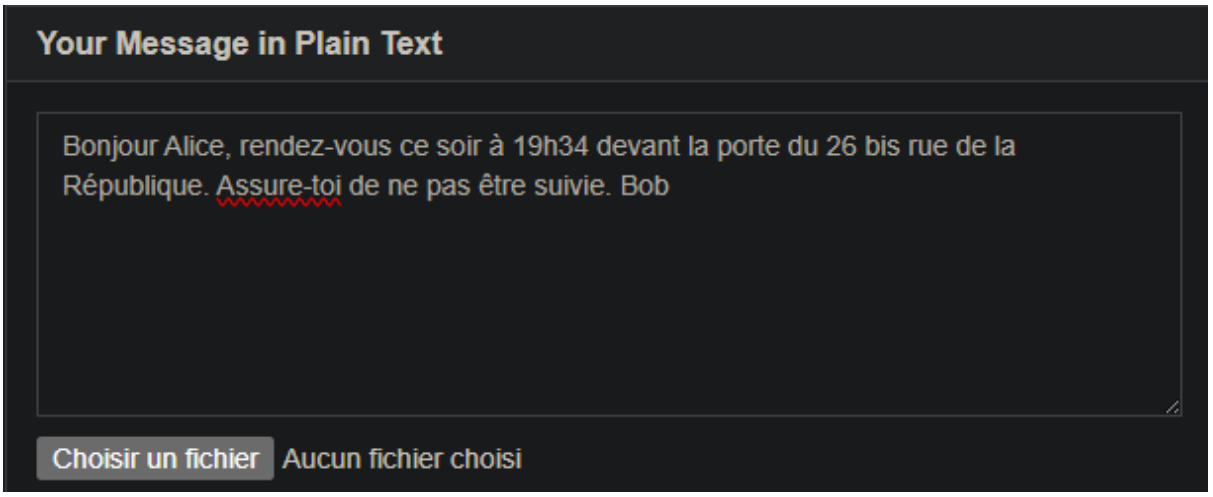
4.1. Chiffrement du message par Bob

Pour simplement chiffrer un message cette fois-ci, on se rend dans l'onglet "Encrypt(+Sign)":

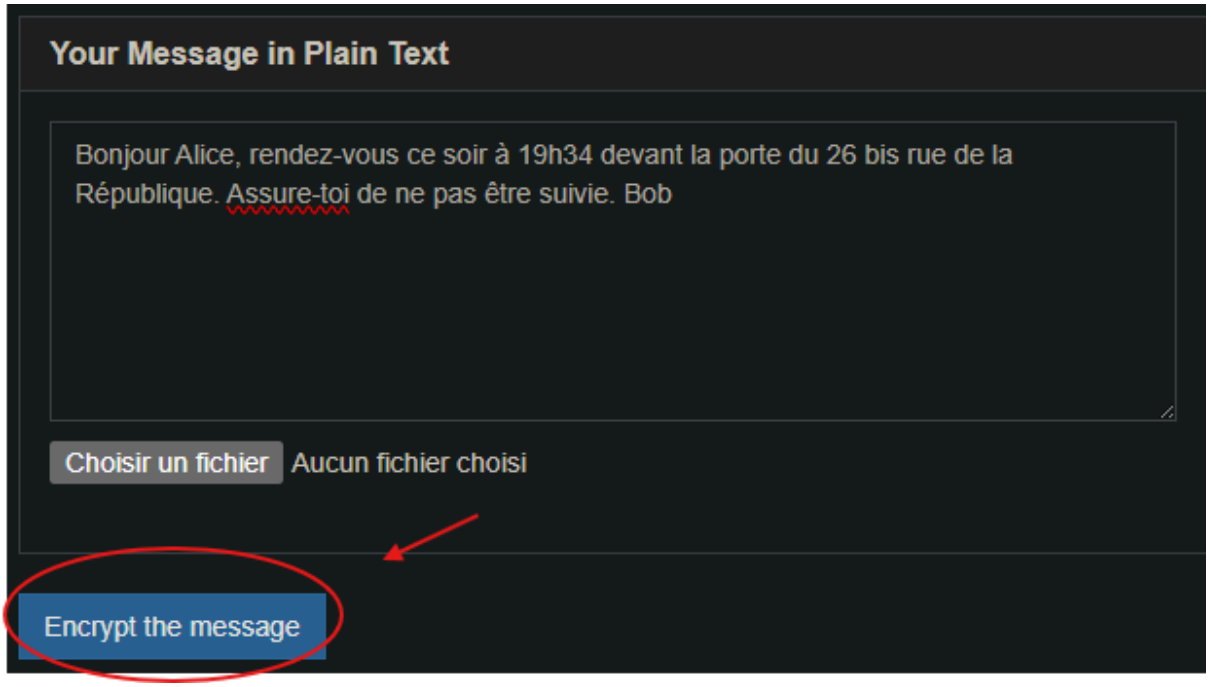


Pour envoyer un message chiffré à Alice, Bob doit utiliser la clé publique d'Alice, son destinataire. Dans la partie gauche de l'écran on charge donc le fichier contenant la clé publique d'Alice:

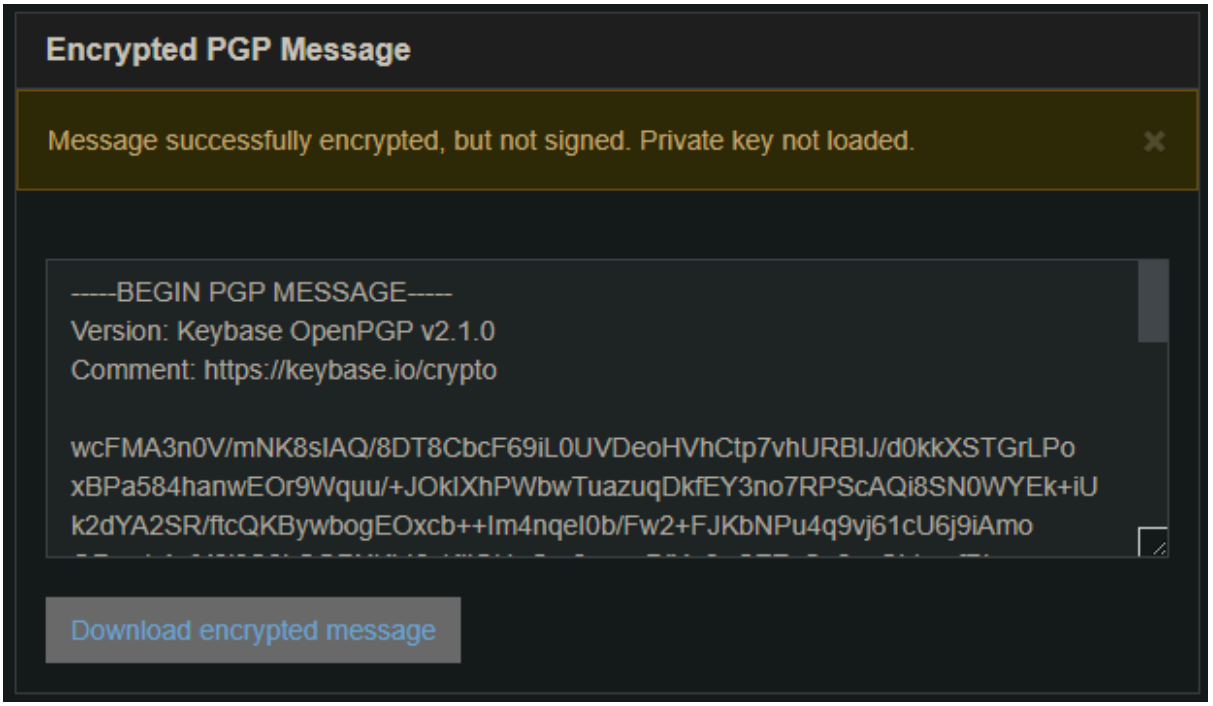
Ensuite, dans la partie droite de l'écran, on entre le message à envoyer:



Puis, on clique sur le bouton "Encrypt the message":



On obtient alors le message chiffré que l'on peut télécharger:

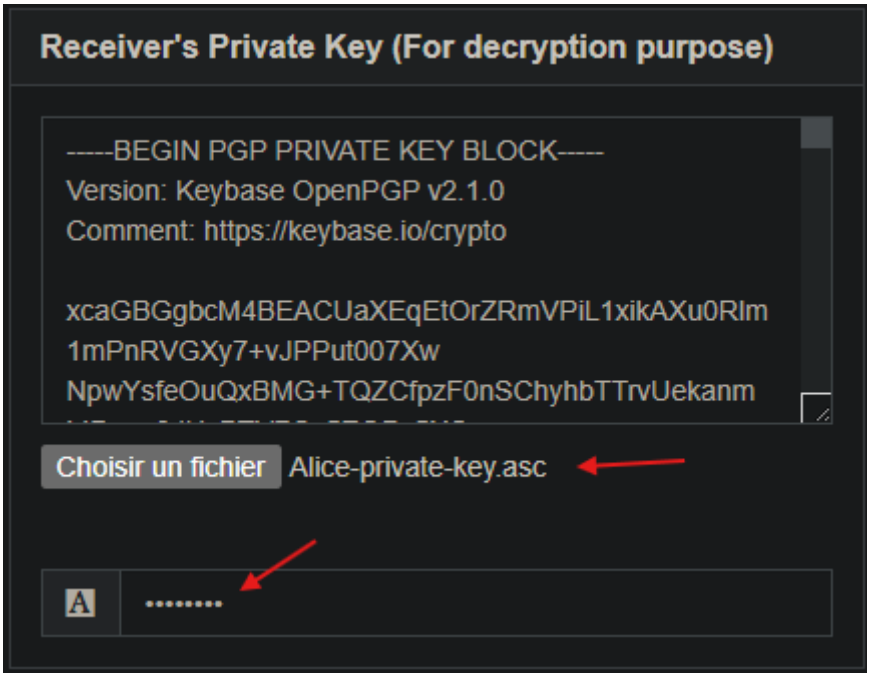


On observe un message d'attention: le message a bien été chiffré, mais pas signé. Cela est normal, car Bob a fait le choix de ne pas signer son message, mais de simplement le chiffrer.

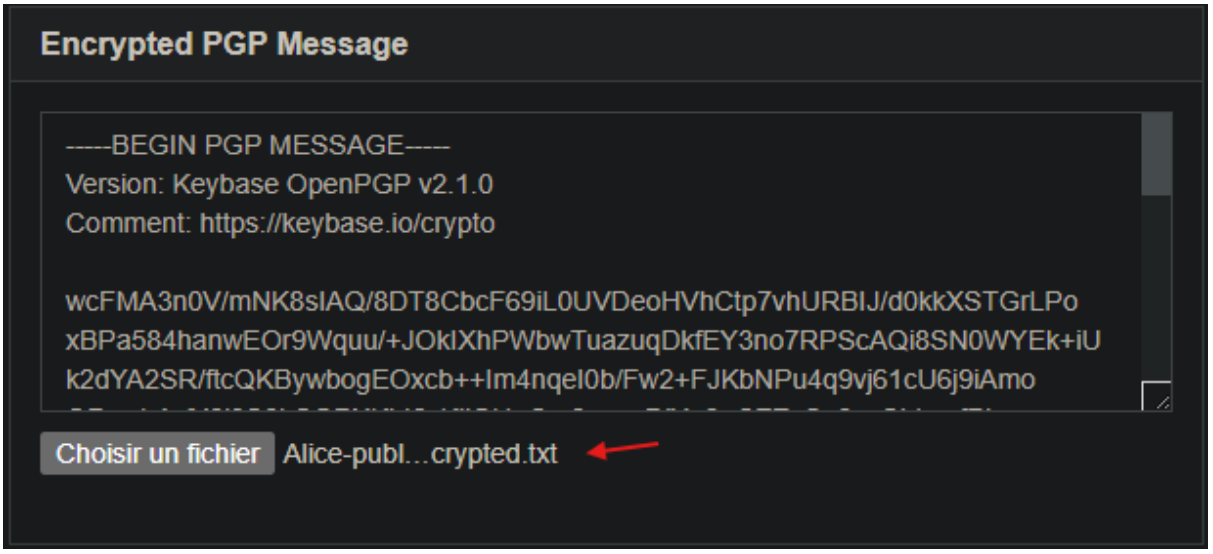
4.2. Déchiffrement du message par Alice

Pour qu'Alice puisse bien déchiffrer le message envoyé par Bob, on se rend dans l'onglet "Decrypt(+Sign)":

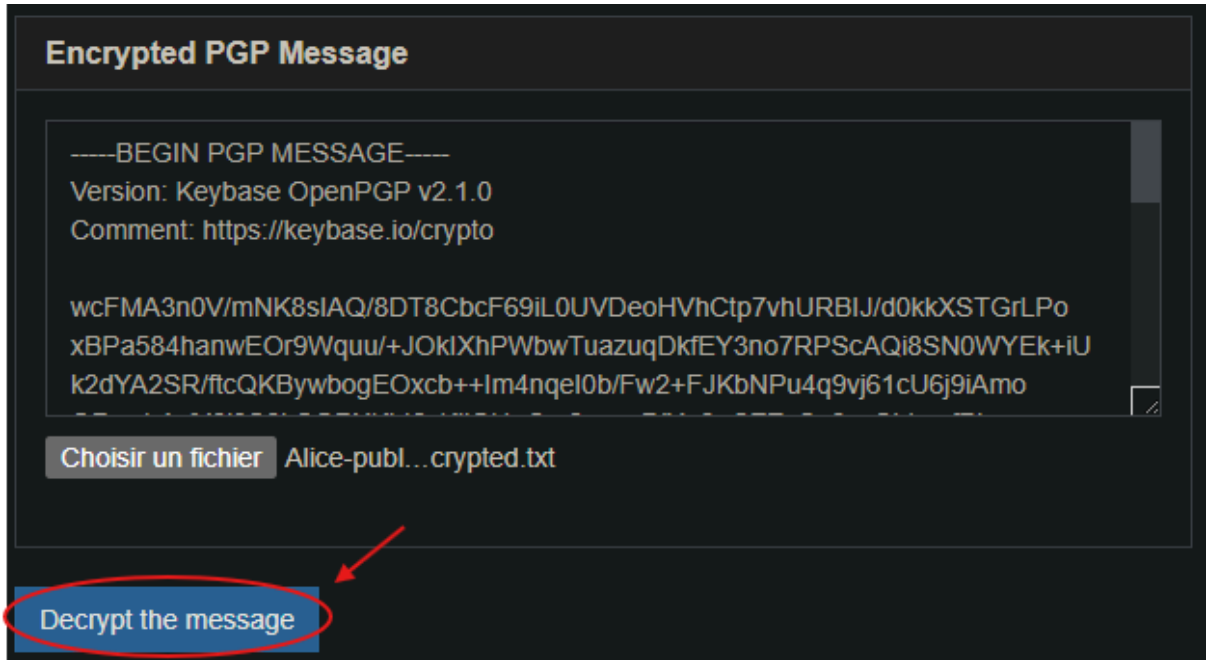
Pour déchiffrer le message envoyé par Bob, Alice doit utiliser sa propre clé privée. Dans la partie gauche de l'écran on charge donc le fichier contenant la clé privée d'Alice et on entre sa passphrase:



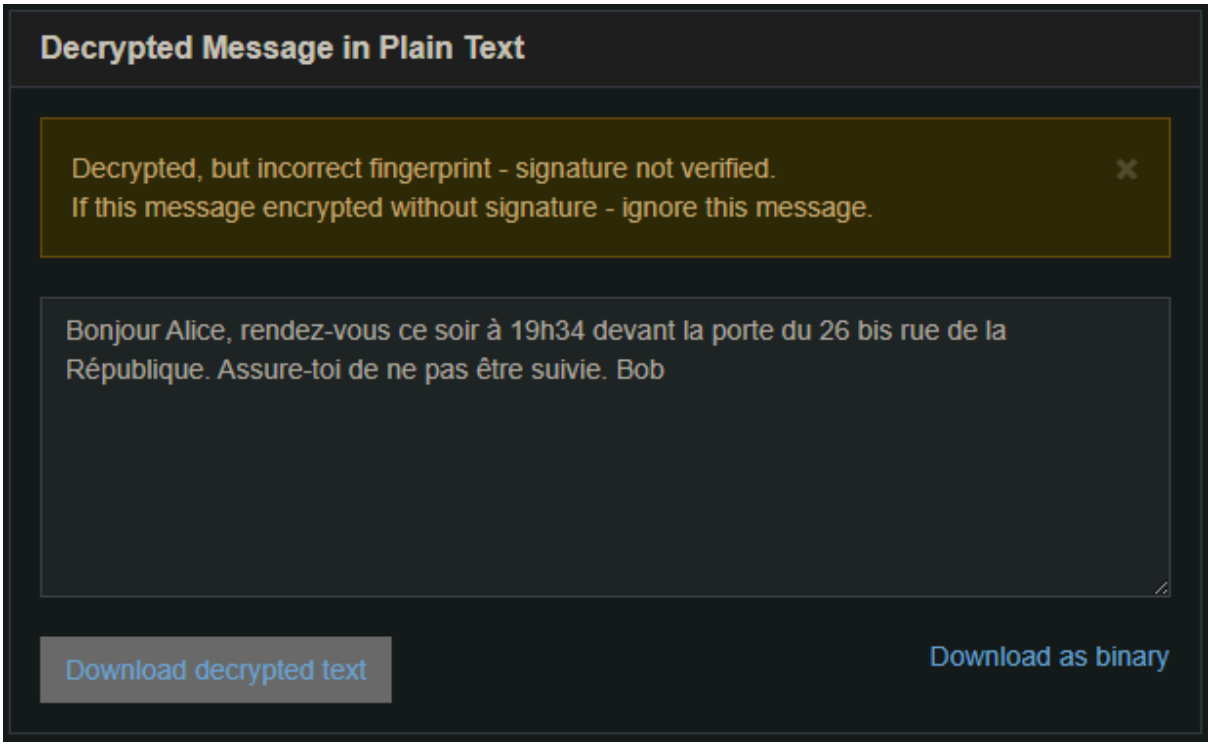
Ensuite, dans la partie droite, on charge le message chiffré par Bob:



Puis, on clique sur le bouton “Decrypt the message”:

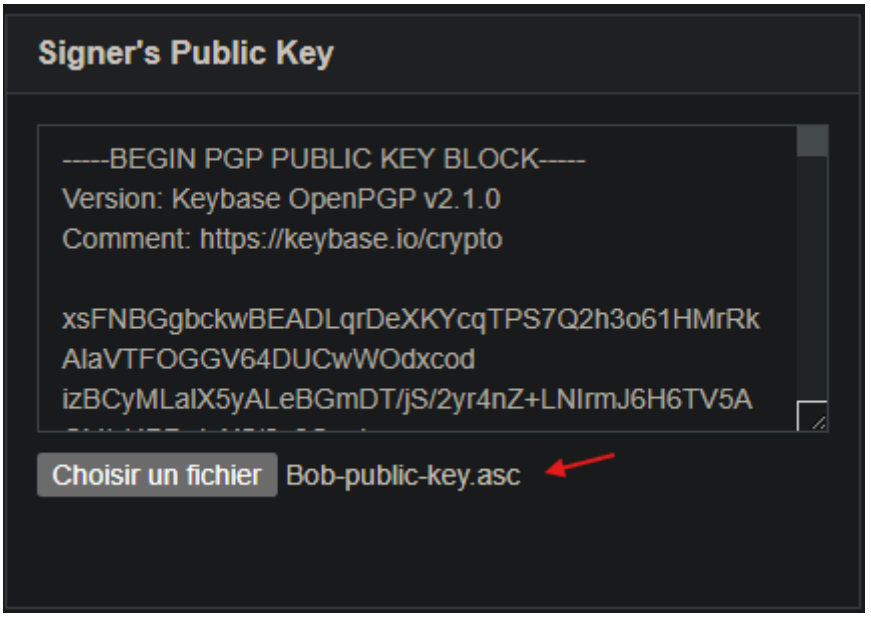


On obtient alors le message déchiffré que l’on peut télécharger:

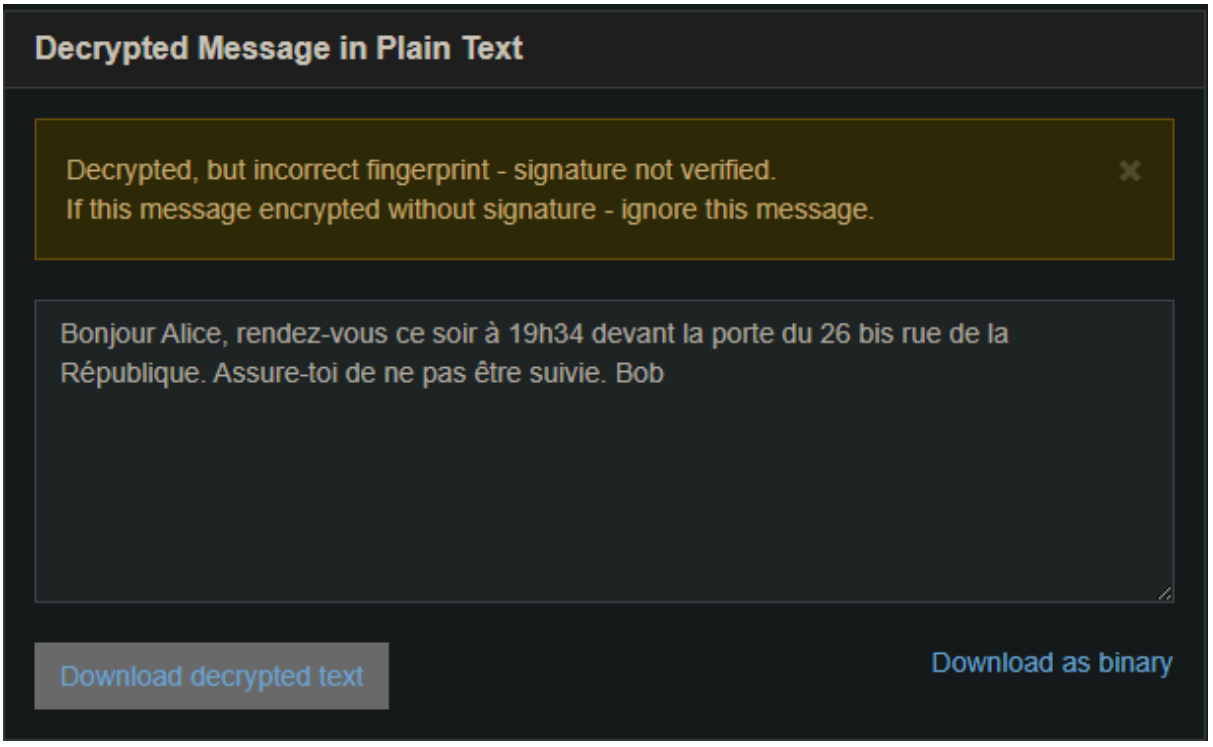


On constate cependant comme ci-dessus que l’on ne peut pas prouver que c’est Bob qui a envoyé ce message. Cela signifie que dans cette étape, le chiffrement du message avec la clé publique d’Alice garantit la confidentialité, cependant, l’authenticité du message n’est pas garantie.

En suivant les étapes précédentes, on peut dans la partie gauche de l'écran, ajouter la clé publique de Bob pour vérifier la signature:



Après cela, si l'on clique sur le bouton "Decrypt the message", on obtient:



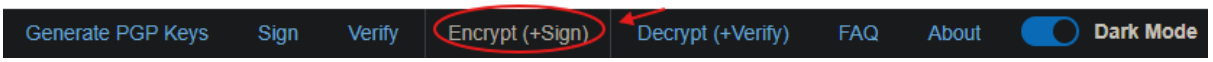
On constate cependant que l'on ne peut toujours pas prouver que c'est Bob qui a envoyé ce message. En effet, Bob n'a pas signé son message avant de l'envoyer à Alice.

5. Etape 3 : chiffrer et signer un message

Cette fois-ci, Bob souhaite chiffrer et signer son message destiné à Alice.

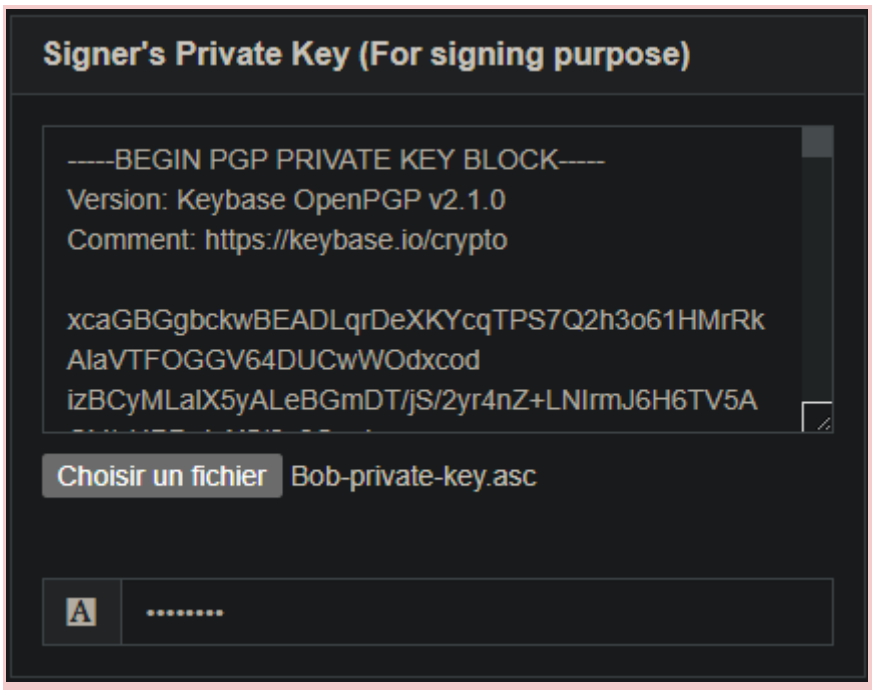
5.1. Chiffrement et signature par Bob

Pour pouvoir à la fois chiffrer et signer un message cette fois-ci, on se rend dans l'onglet "Encrypt(+Sign)":

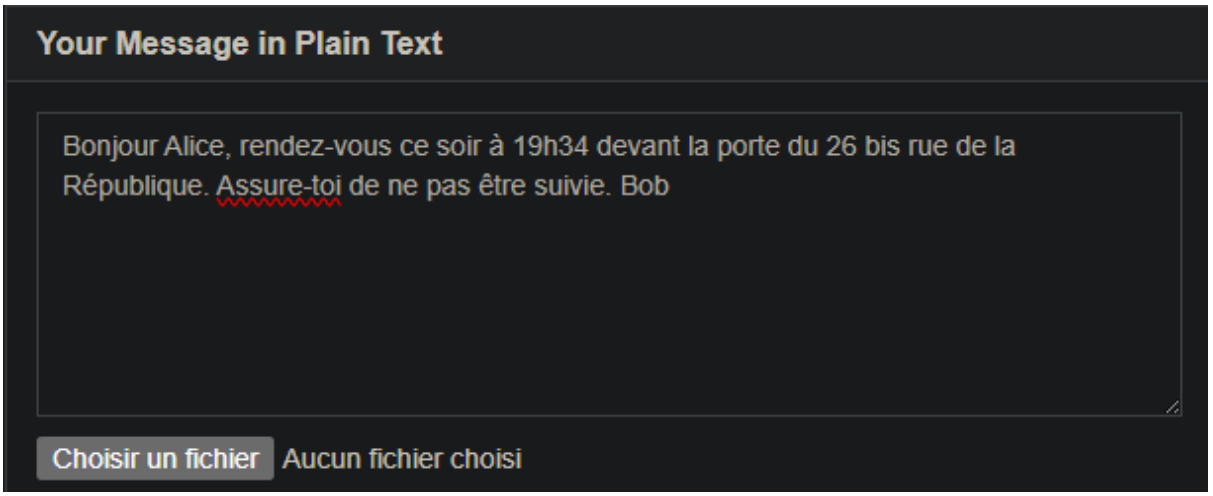


Pour envoyer un message chiffré et signé à Alice, Bob doit utiliser la clé publique d'Alice, son destinataire. Dans la partie gauche de l'écran on charge donc le fichier contenant la clé publique d'Alice:

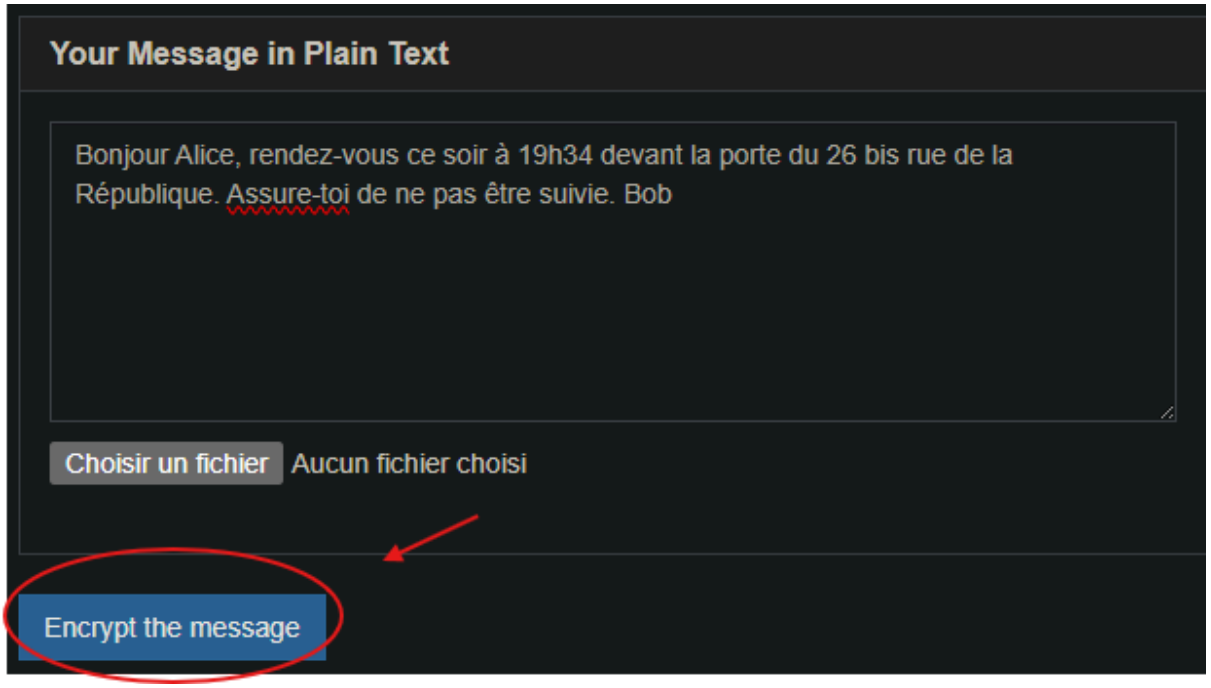
Pour signer son message, Bob doit utiliser sa propre clé privée. Toujours dans la partie gauche de l'écran, on ajoute la clé privée de Bob pour signer le message et la passphrase qu'il a défini:



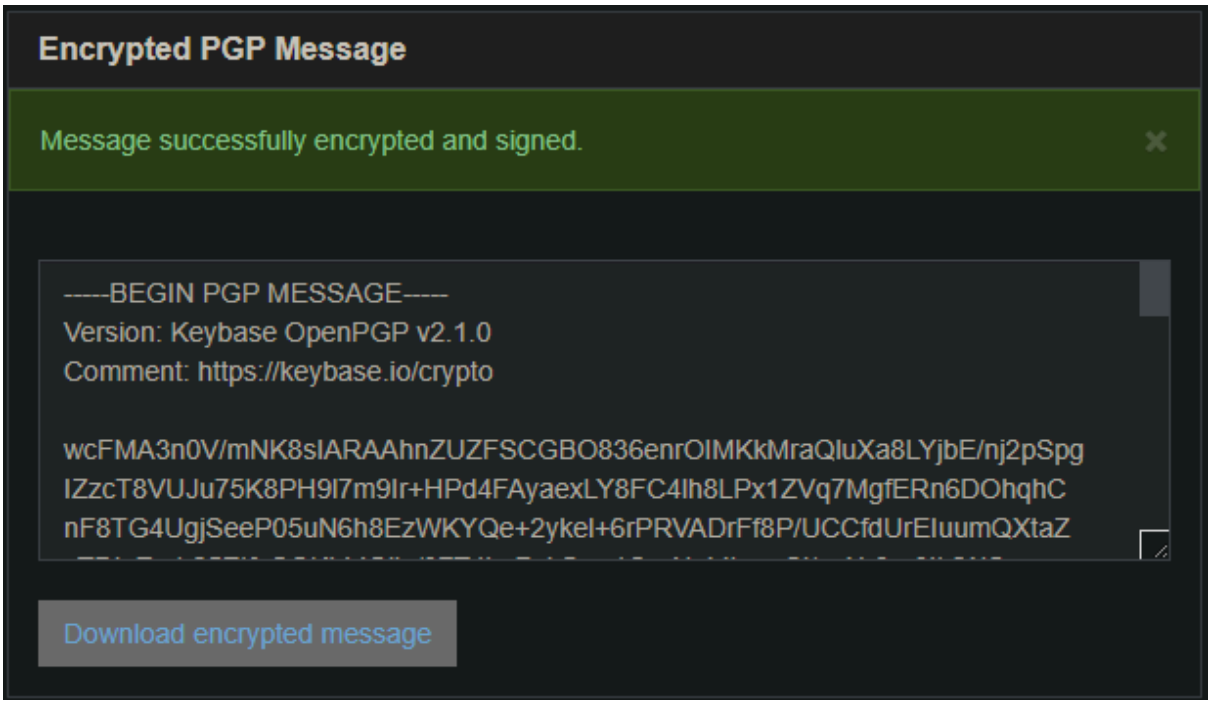
Ensuite, dans la partie droite de l'écran, on entre le message à envoyer:



Puis, on clique sur le bouton “Encrypt the message”:

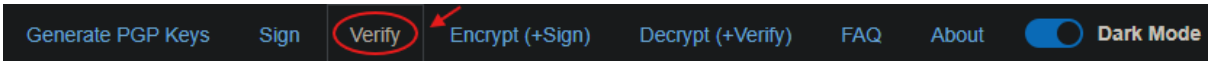


On obtient alors le message chiffré et signé que l’on peut télécharger:

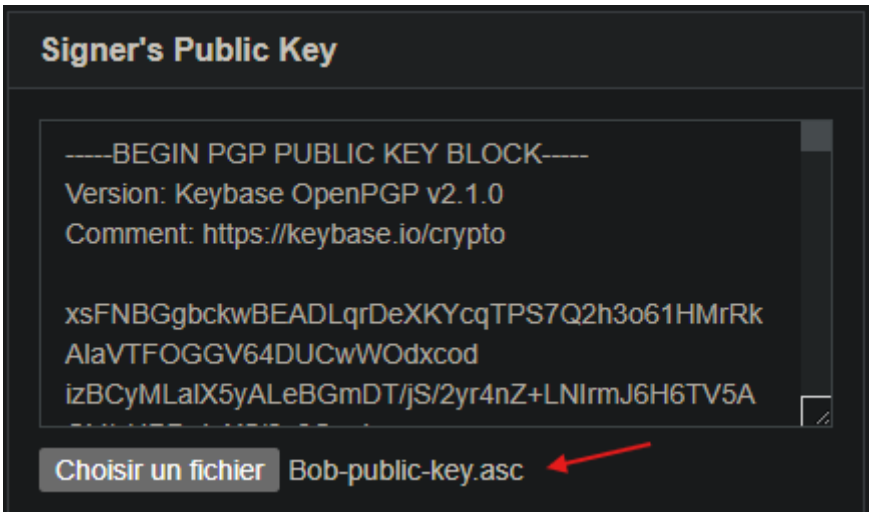


5.2. Vérification et déchiffrement par Alice

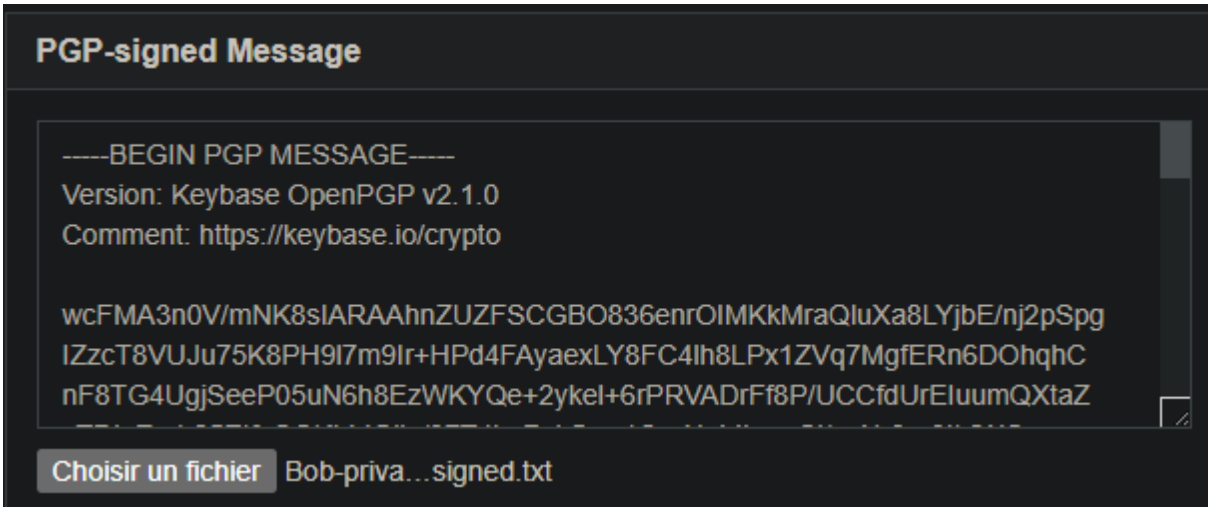
Pour qu'Alice puisse vérifier que c'est bien Bob qui lui a envoyé un message, on va dans l'onglet "Verify":



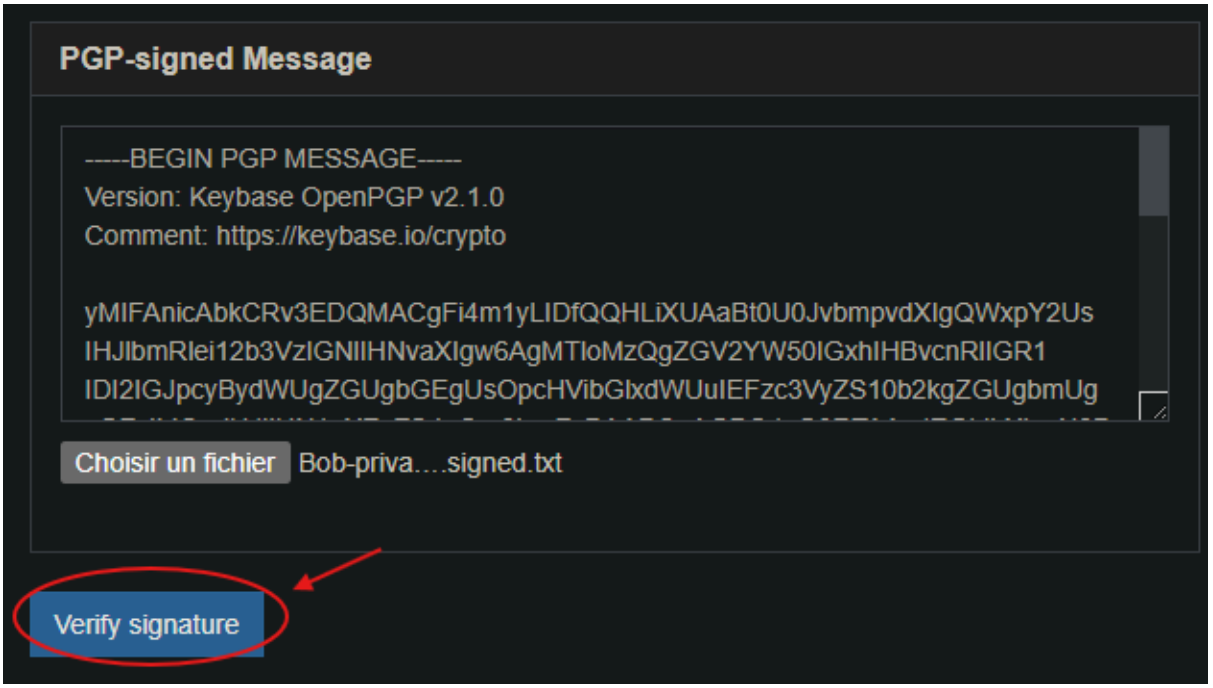
Dans la partie gauche de l'écran, on charge le fichier contenant la clé publique de Bob, considérant qu'il lui a envoyé auparavant:



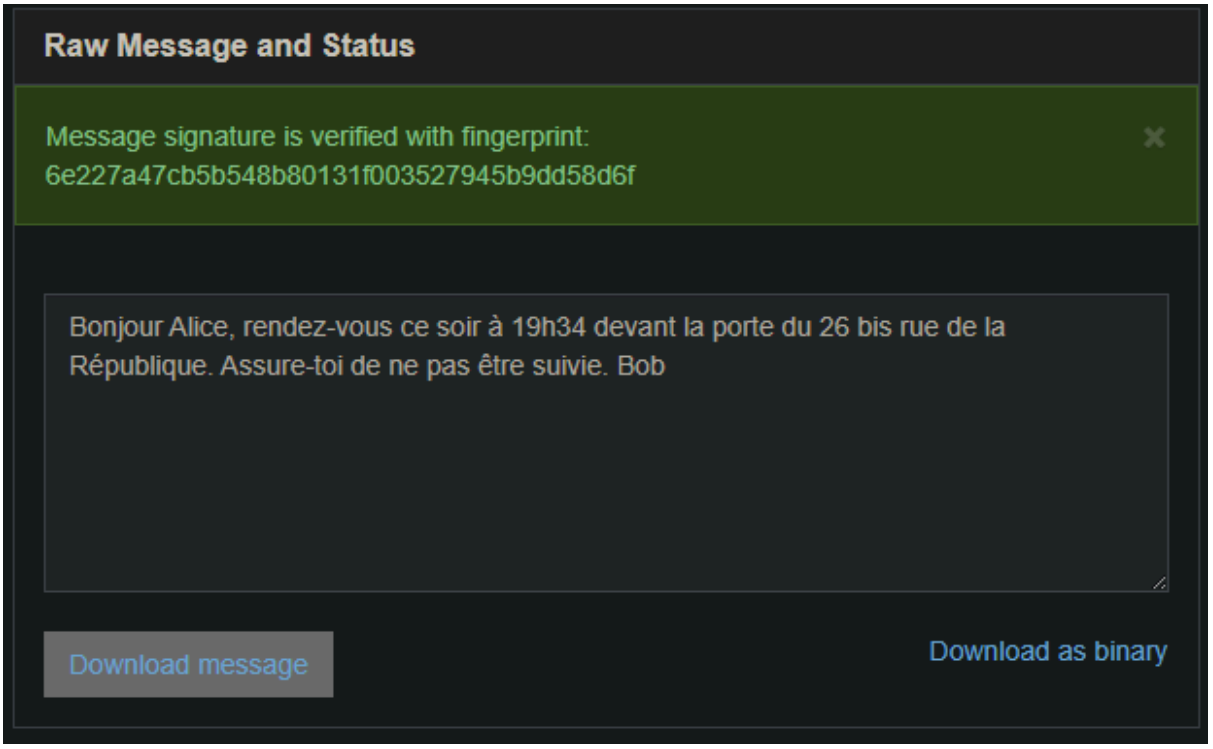
Ensuite, dans la partie droite, on charge le message chiffré et signé par Bob:



Enfin, on clique sur le bouton “Verify signature”:

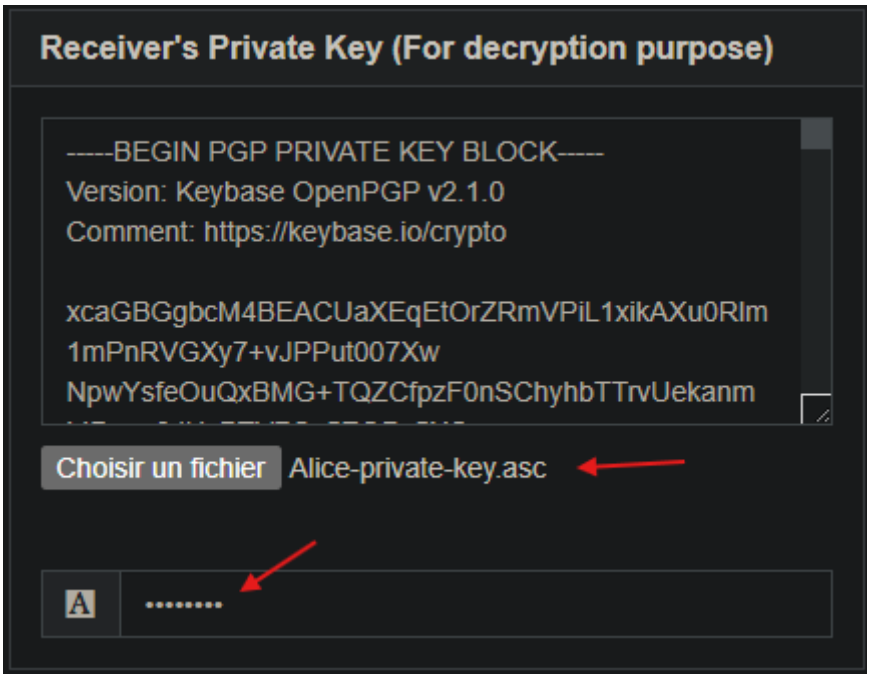


On obtient alors le message avec la vérification de la signature:



Pour qu’Alice puisse bien déchiffrer le message envoyé par Bob, on se rend dans l’onglet “Decrypt(+Verify)”:

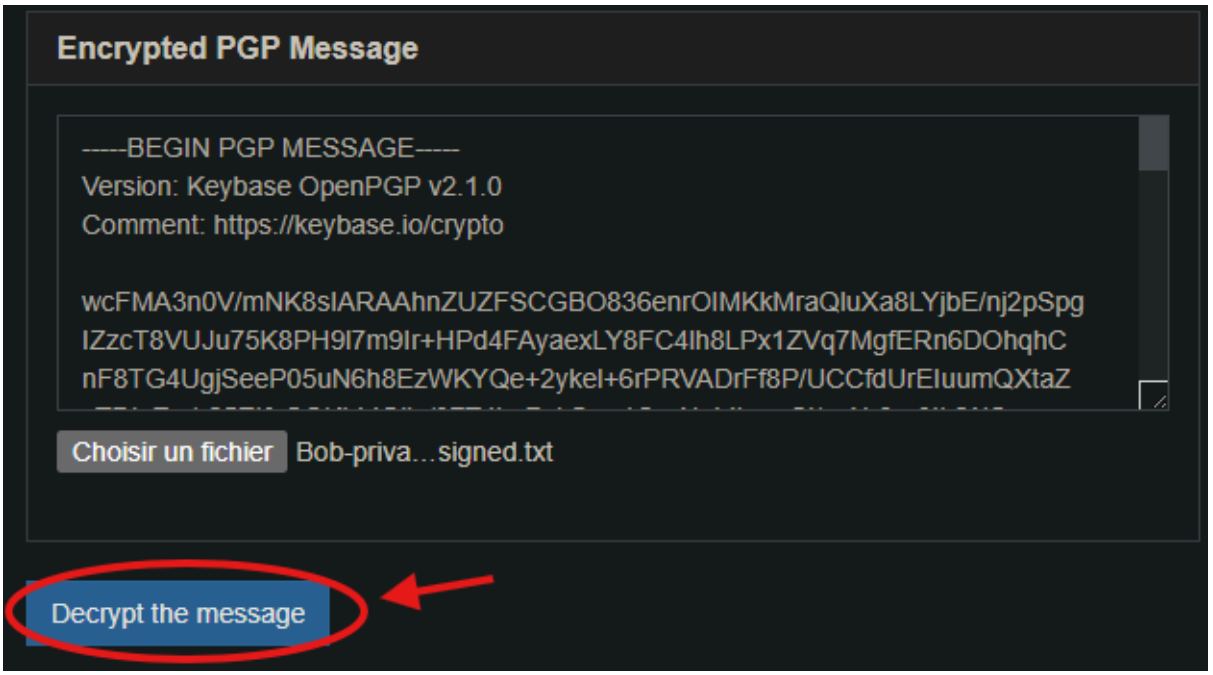
Pour déchiffrer le message envoyé par Bob, Alice doit utiliser sa propre clé privée. Dans la partie gauche de l'écran on charge donc le fichier contenant la clé privée d'Alice et on entre sa passphrase:



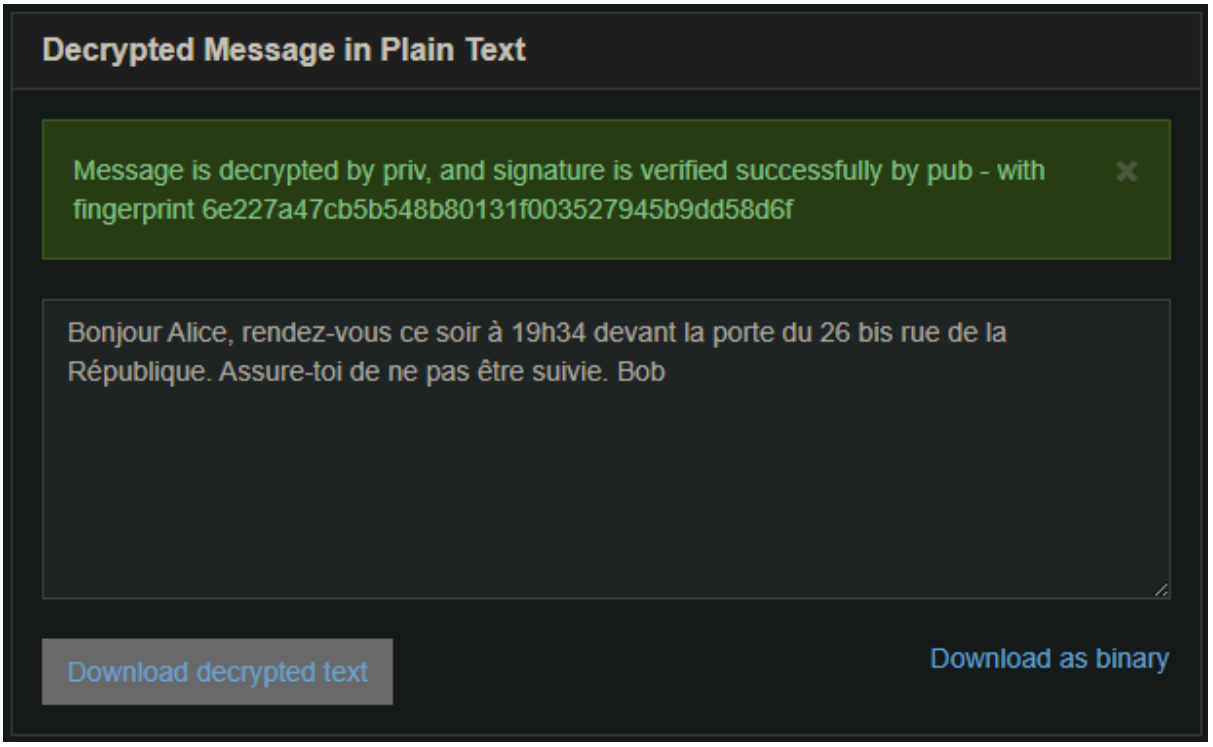
Pour vérifier la signature du message, Alice doit utiliser la clé publique de Bob. Toujours dans la partie gauche de l'écran, on ajoute la clé publique de Bob pour vérifier la signature:

Ensuite, dans la partie droite, on charge le message chiffré et signé par Bob:

Puis, on clique sur le bouton “Decrypt the message”:



On obtient alors le message déchiffré que l'on peut télécharger:



On constate que le message est non seulement déchiffré, mais que la signature est également vérifiée. Cela signifie que la confidentialité et l'authenticité du message sont assurées grâce à l'utilisation du chiffrement et de la signature numérique.

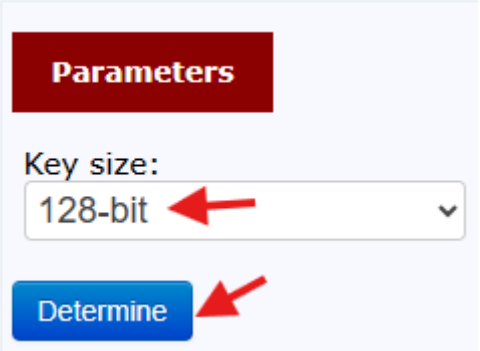
6. Etape 4 : Pour aller plus loin… PGP

Pour s'échanger un très gros volume de données, Bob et Alice vont mettre en œuvre un échange « PGP », qui utilise moins de ressources de leurs ordinateurs que la méthode de chiffrement/déchiffrement asymétrique.

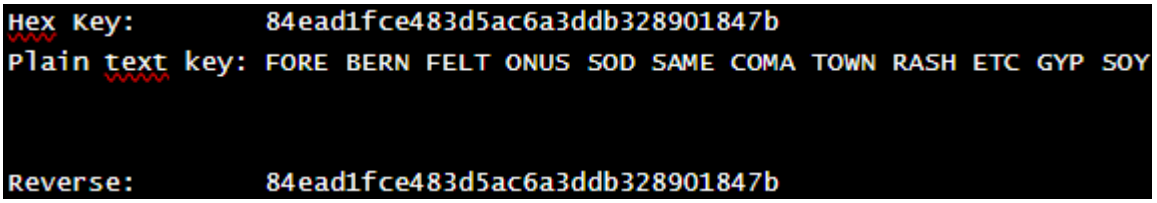
6.1. Génération de la clé de chiffrement AES

Pour cette méthode, il est nécessaire d'obtenir une clé de chiffrement AES, qu'il est possible de créer en se rendant sur le site https://asecuritysite.com/encryption/plain.

Dans les paramètres qui apparaissent à l'écran, nous choisissons une clé 128-bit et cliquons sur "Determine":



On obtient alors une clé de chiffrement AES 128-bit, que l'on copie pour la suite:

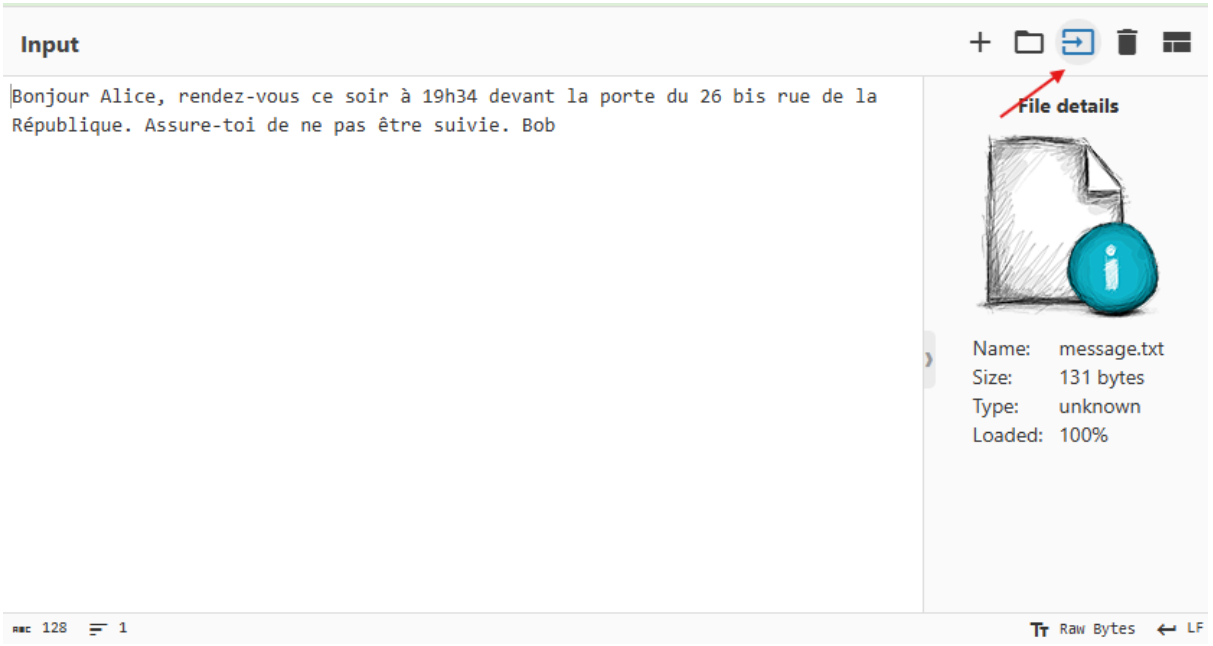


6.2. Chiffrement du fichier avec la clé AES

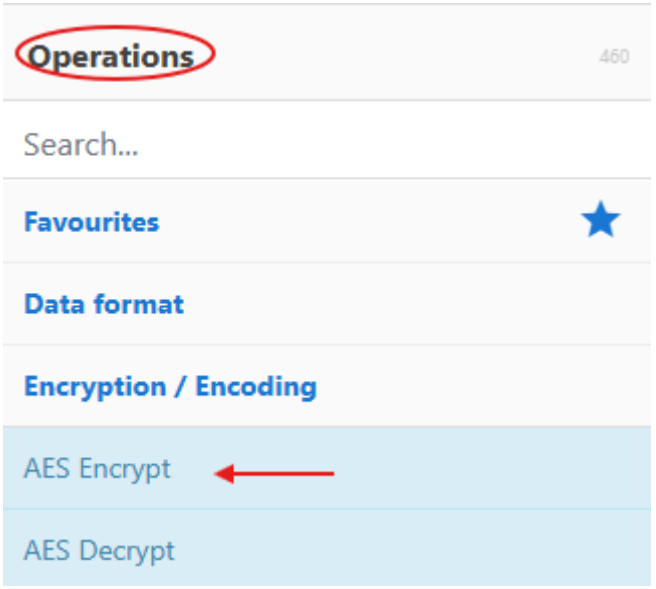
Imaginons que le fichier que Bob veut transférer à Alice est un fichier texte contenant le message secret: "Bonjour Alice, rendez-vous ce soir à 19h34 devant la porte du 26 bis rue de la République. Assure-toi de ne pas être suivie. Bob"

Grâce au site CyberChef, Bob va pouvoir chiffrer ce fichier avec la clé de chiffrement AES obtenue juste au-dessus.

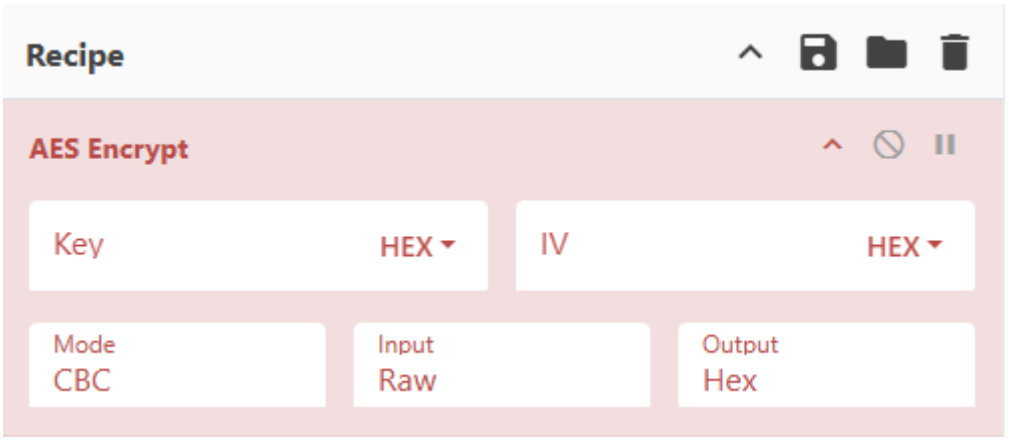
Dans la partie “Input”, Bob peut charger son fichier texte en cliquant sur le signe correspondant, ce qui retranscrit directement le message:



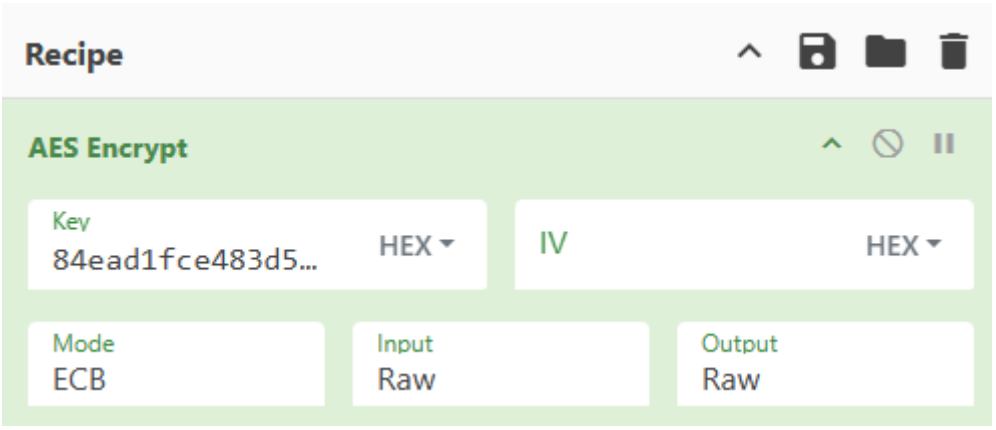
Ensuite, dans “Operations”, il peut choisir “AES encrypt” qui signifie chiffrement avec algorithme AES:



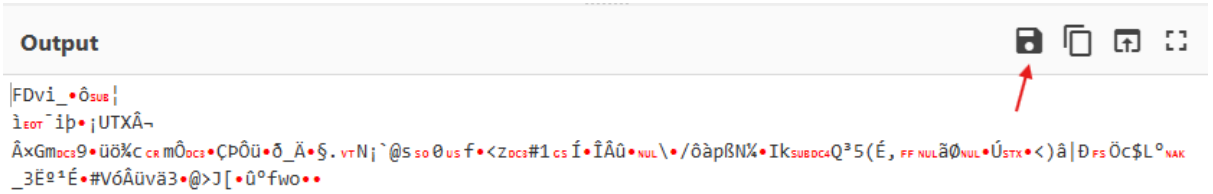
S’affiche alors à l’écran une nouvelle partie dans l’onglet “Recipe”:



Dans cette partie, il faut coller la clé 128-bit créée précédemment, choisir le mode ECB et définir Raw pour Input et Output:

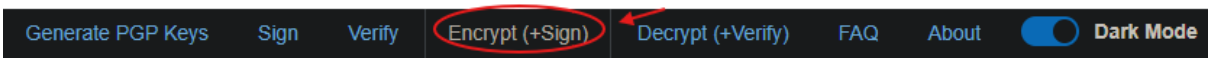


On observe que le message a été chiffré dans “Output”, car le bouton “Auto-Bake” (qui permet de chiffrer automatiquement) est activé par défaut. Il est possible de télécharger ce fichier chiffré:



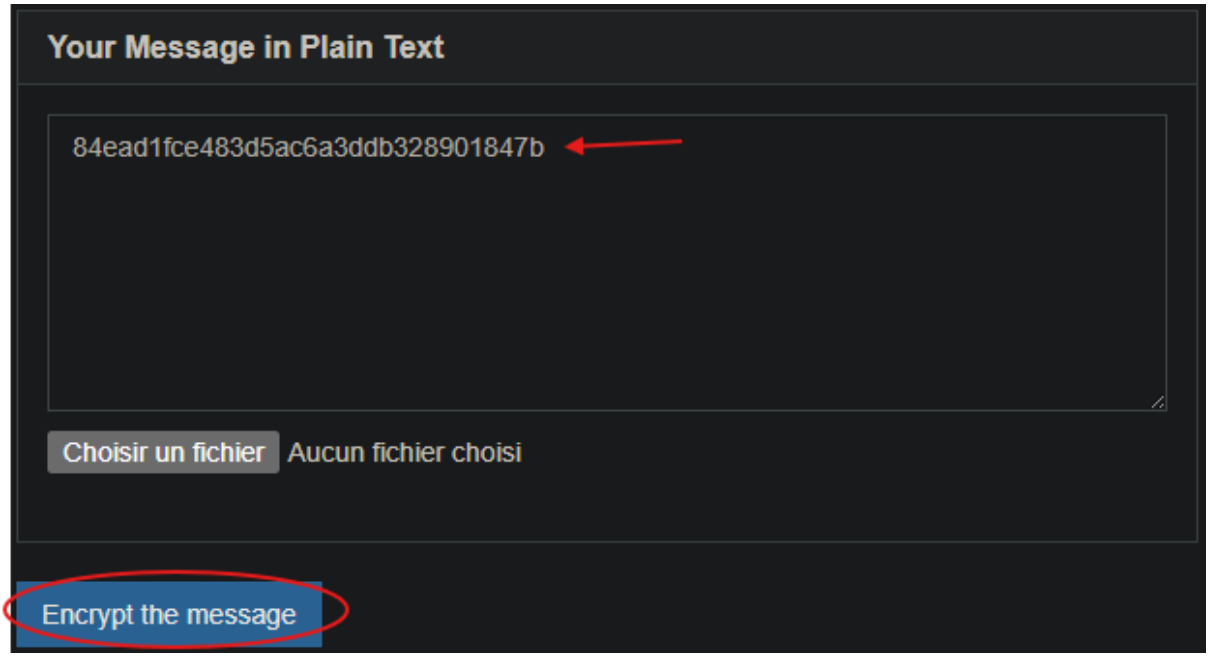
Avec le site https://pgp.craeckor.ch/ , Bob va pouvoir chiffrer la clé AES avec la clé publique d’Alice.

Pour ce faire, on se rend dans l’onglet “Encrypt (+Sign)”:

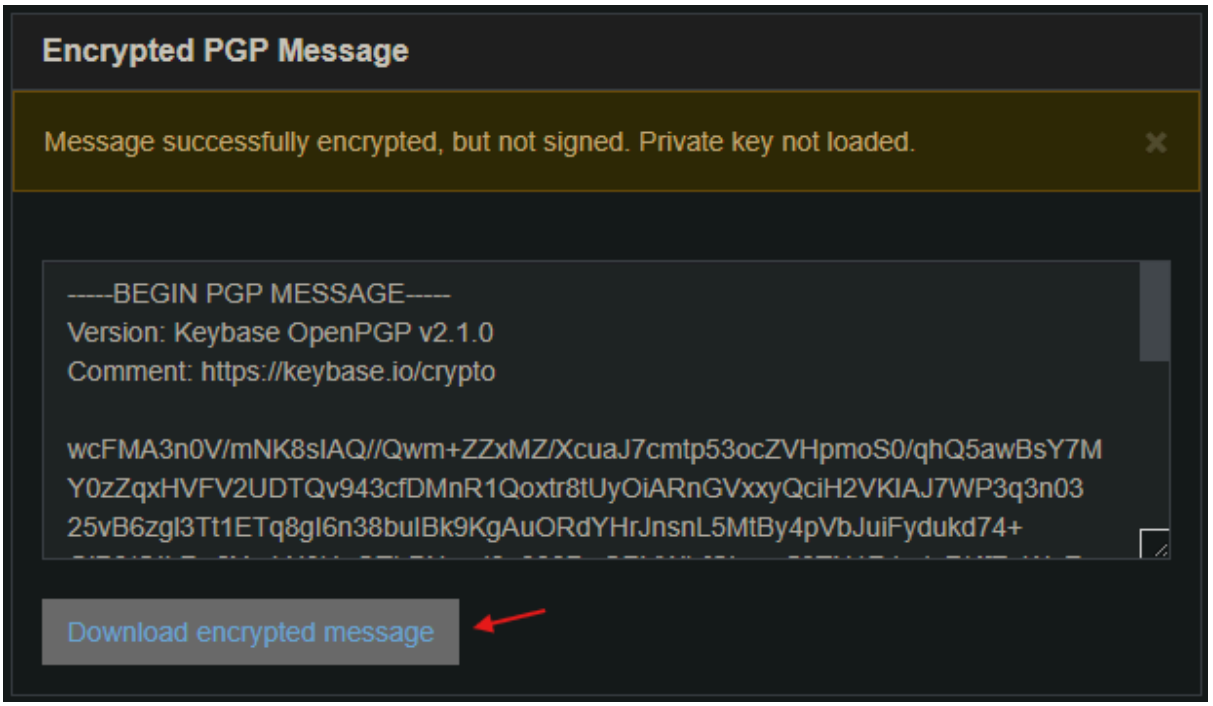


Dans la partie gauche de l'écran, on charge la clé publique d'Alice (puisque c'est elle qui doit pouvoir déchiffrer la clé AES):

Dans la partie droite de l'écran, on entre la clé AES 128-bit créée précédemment, puis on clique sur "Encrypt the message":



On obtient alors la clé AES chiffrée avec la clé publique d'Alice, que l'on peut télécharger:



Alice, étant la seule à posséder la clé privée correspondante, elle seule pourra déchiffrer cette clé AES.On remarque que le message contenant la clé est seulement chiffré car Bob ne l'a pas signé.

6.3. Transmission des fichiers à Alice

Bob envoie ensuite à Alice le fichier texte chiffré avec la clé AES et la clé AES chiffrée avec la clé publique d'Alice.

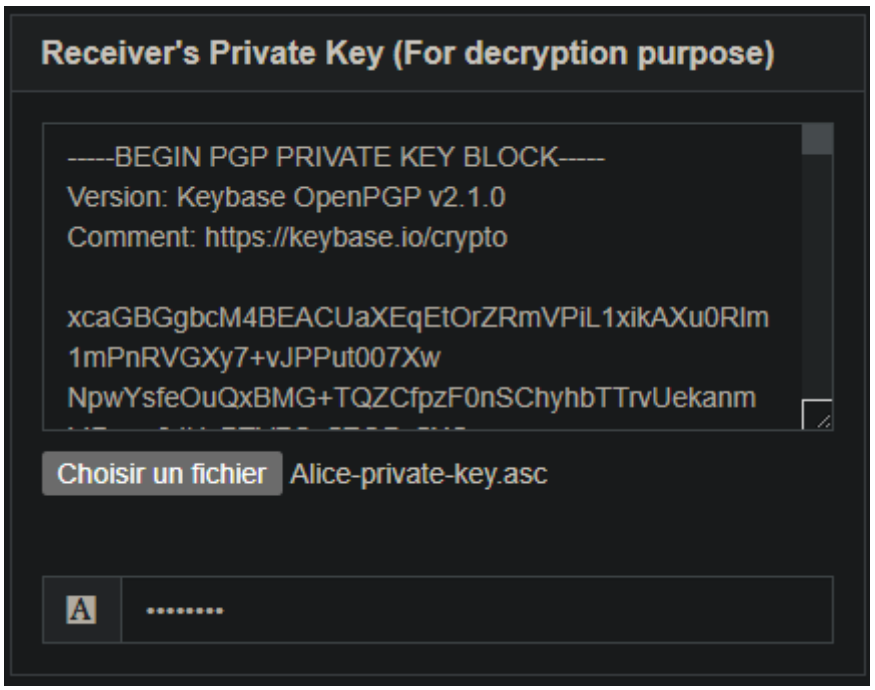
6.4. Déchiffrement du côté d'Alice

Pour récupérer le fichier original, Alice doit suivre deux étapes: déchiffrer la clé AES et déchiffrer le fichier.

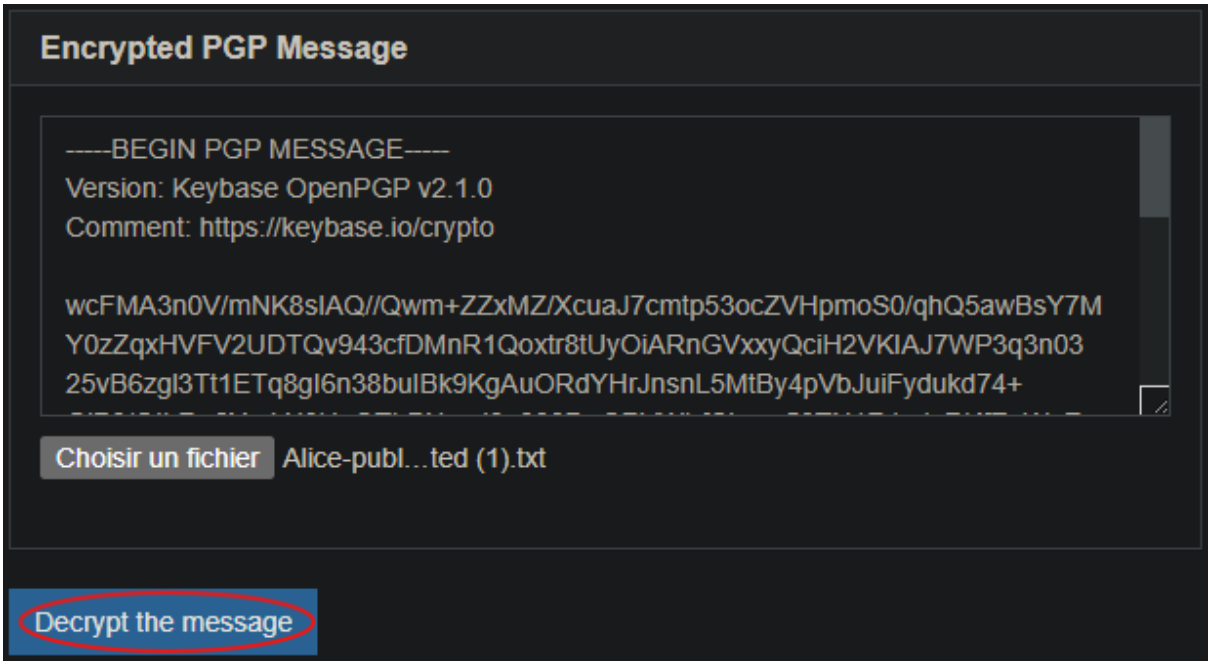
6.4.1. Déchiffrer la clé AES

Alice de son côté va déchiffrer la clé AES. Pour cela, elle va sur le site https://pgp.craeckor.ch/ dans l'onglet "Decrypt(+Verify)":

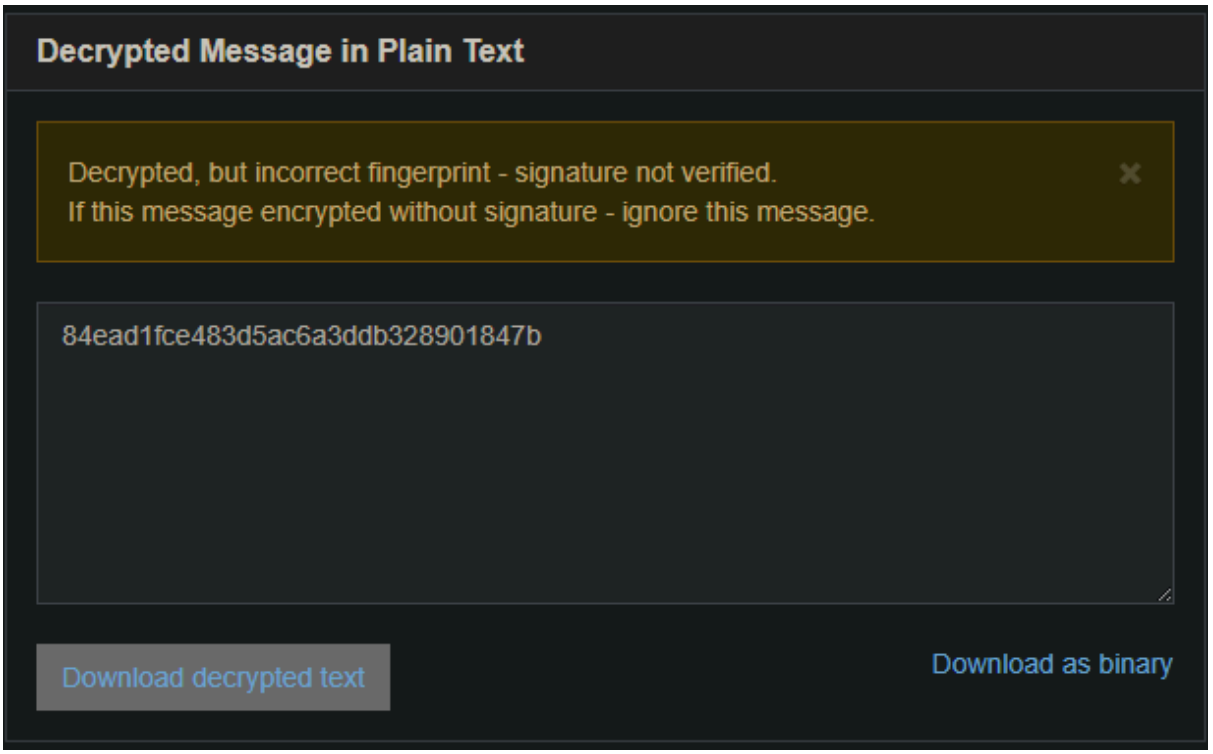
Dans la partie gauche de l'écran, elle charge sa clé privée et entre sa passphrase:



Dans la partie droite de l'écran, elle charge le fichier chiffré avec sa clé publique par Bob et clique sur "Decrypt the message". Il s'agit du fichier contenant la clé AES 128-bit:



La clé apparaît alors en clair:



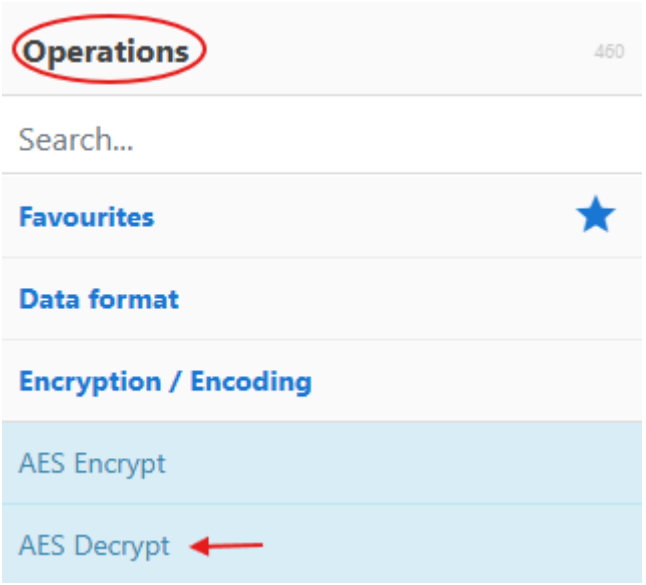
Etant donné que Bob n'a pas signé son message, on nous indique que l'empreinte n'a pas pû être vérifiée.

6.4.2. Déchiffrer le fichier

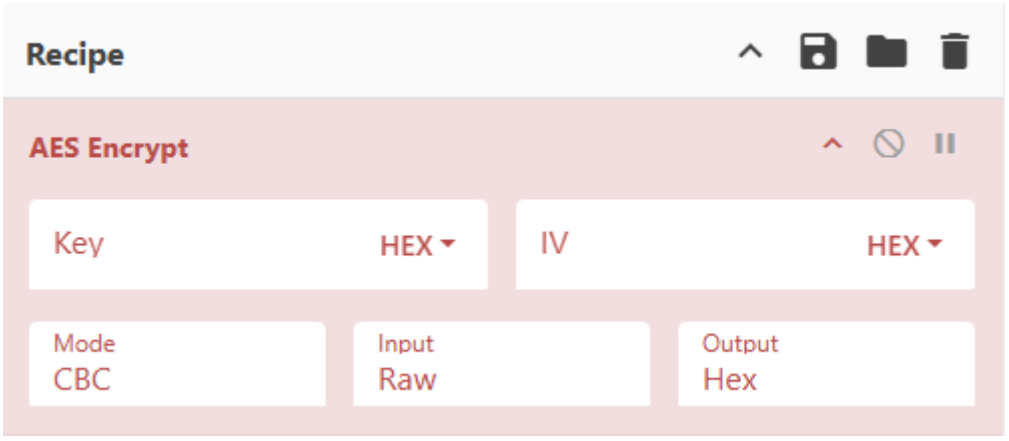
Ensuite, pour déchiffrer le fichier texte, elle va sur le site CyberChef et ouvre le fichier:

On observe que le contenu est bien chiffré.

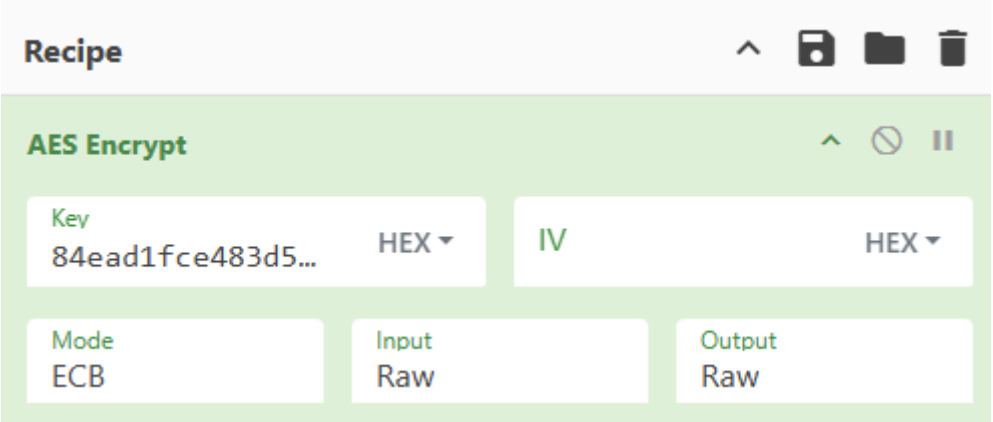
Pour le déchiffrer, Alice va se rendre dans “Operations” et choisir “AES decrypt” qui signifie déchiffrement avec algorithme AES:



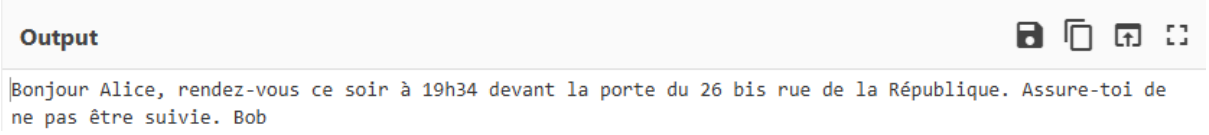
S’affiche alors à l’écran une nouvelle partie dans l’onglet “Recipe”:



Dans cette partie, il faut coller la clé 128-bit qu'Alice a pu déchiffrer juste avant, choisir le mode ECB et définir Raw pour Input et Output:



On observe que le message a été déchiffré dans "Output", car le bouton "Auto-Bake" (qui permet de déchiffrer automatiquement) est activé par défaut:



Après avoir déchiffré ce fichier, Alice peut accéder au contenu et voir le message de Bob concernant un rendez-vous secret.

Après ces manipulations, on peut conclure qu'avec cette méthode hybride PGP la confidentialité du fichier est assurée car :

- Le fichier est chiffré avec une clé AES aléatoire, donc illisible pour toute personne ne possédant pas cette clé.
- La clé AES elle-même est chiffrée avec la clé publique d'Alice : seule Alice, grâce à sa clé privée, peut la déchiffrer et donc accéder au contenu du fichier.

Cependant, l'authenticité n'est pas garantie automatiquement, car Bob n'a pas signé la clé AES ni le fichier.

Cela signifie qu'Alice ne peut pas être certaine que le fichier a bien été envoyé par Bob, car n'importe qui pourrait chiffrer un fichier avec sa clé publique et lui envoyer.

Pour garantir l'authenticité, Bob devrait également signer la clé AES ou le fichier avec sa propre clé privée. De cette manière, Alice pourrait vérifier la signature à l'aide de la clé publique de Bob, et être sûre de l'identité de l'expéditeur.

7. Chiffrement symétrique/asymétrique

Suite aux différentes étapes, il faut bien comprendre que le chiffrement symétrique utilise une seule clé pour chiffrer et déchiffrer les données. Il est très efficace pour chiffrer de grandes quantités de données, mais il demande de partager la clé de manière sécurisée.

Le chiffrement asymétrique, lui, utilise une paire de clés (publique/privée). Il permet de résoudre le problème de distribution de clé, mais il est beaucoup plus lent. C'est pour cela que la méthode hybride PGP est utile: elle permet de combiner les bénéfices des des types de chiffrement.

8. Conclusion

Ce TP a donc permis d'utiliser différentes méthodes pour sécuriser et authentifier les échanges d'information entre Alice et Bob.

Les étapes réalisées ont montré que le chiffrement garantit la confidentialité des messages, tandis que la signature numérique permet de vérifier l'identité de l'expéditeur et d'assurer l'authenticité.

Utiliser ces deux méthodes ensemble permet d'obtenir un échange confidentiel et authentifié.

Enfin, l'utilisation de la méthode hybride PGP avec une clé AES a montré qu'il était possible transférer de gros volumes de données, tout en gardant un haut niveau de sécurité.

Les différentes étapes réalisées prouvent donc l'importance d'utiliser à la fois le chiffrement et la signature pour assurer la protection et la fiabilité des messages échangés.

9. Glossaire

Pour mieux comprendre ce TP, voici un glossaire des mots à connaître concernant la sécurisation et l'authentification des échanges d'information.

9.1. Chiffrer / "Encrypt"

C'est le fait d'encoder de l'information afin qu'elle puisse être transmise ou stockée sans qu'une personne malveillante puisse la consulter et la comprendre. Le chiffrement se fait en appliquant une clé sur la donnée à chiffrer.

9.2. Déchiffrer / "Decrypt"

C'est le fait de décoder l'information chiffrée afin de pouvoir la comprendre.

9.3. Chiffrement/déchiffrement symétrique

C'est le fait de chiffrer une donnée avec une clé et de la déchiffrer avec la même clé.

9.4. Chiffrement/déchiffrement asymétrique

C'est le fait de chiffrer une donnée avec une clé (publique) et de la déchiffrer avec une autre clé (privée).

9.5. Clé publique

C'est la clé mise à disposition par un utilisateur à tout le monde, permettant de déchiffrer une donnée chiffrée par la clé privée correspondante.

9.6. Clé privée

C’est la clé d’un utilisateur qu’il conserve secrètement (ou qu’il partage avec des utilisateurs de confiance) et qui lui permet de chiffrer des données qui seront déchiffrées par d’autres grâce à sa clé publique correspondante dans le cadre d’un chiffrement/déchiffrement asymétrique (ou grâce à cette même clé privée par les utilisateurs de confiance dans le cadre d’un chiffrement/déchiffrement symétrique).

9.7. Signer

C’est l’action d’un utilisateur d’appliquer sa clé privée sur une donnée pour prouver qu’il est l’auteur (ou à l’origine) de cette donnée.

9.8. Vérifier (la signature)

C’est l’action d’utiliser la clé publique d’un utilisateur pour valider qu’il est bien à l’origine de la donnée (le signataire).