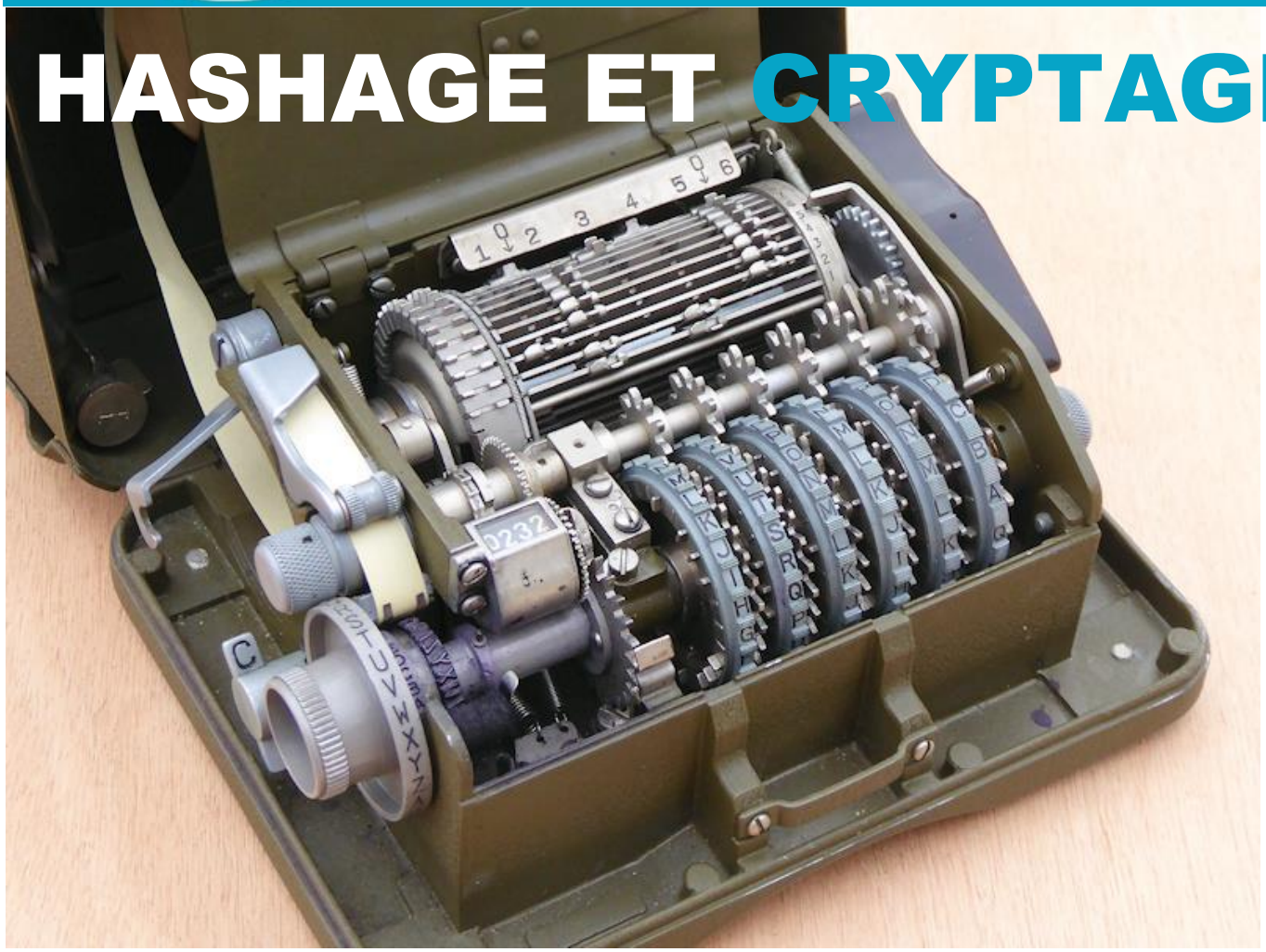
TUTOS-INFO
L'informatique par l'exemple

# HASHAGE ET CRYPTAGE



## SOMMAIRE

## 1 - LE HACHAGE


**Le hachage est un type spécial de calcul unidirectionnel**. Le hachage consiste à prendre une chaîne de données de n'importe quelle taille et la fonction de hachage utilisée donne toujours une sortie d'une longueur déterminée. Cette sortie est appelée **valeur de hachage** ou **résumé du message**.

Peu importe si votre entrée est la totalité du roman *Guerre et Paix* ou simplement deux lettres, **le résultat d'une fonction de hachage aura toujours la même longueur**. Les fonctions de hachage ont plusieurs propriétés différentes qui les rendent utiles :

- **Ce sont des fonctions à sens unique.** Cela signifie qu'il n'existe aucun moyen pratique de déterminer quelle était l'entrée d'origine à partir d'une valeur de hachage donnée.

- **Il est peu probable que deux entrées aient la même valeur de hachage.** Bien qu'il soit possible que deux entrées différentes produisent la même valeur de hachage, les chances que cela se produise sont si faibles que nous ne nous en soucions pas vraiment. À des fins pratiques, les valeurs de hachage peuvent être considérées comme uniques.

- **La même entrée fournit toujours le même résultat.** Chaque fois que vous mettez les mêmes informations dans une fonction de hachage donnée, elle fournira toujours la même sortie.

- **Même le moindre changement donne un résultat complètement différent.** Si même un seul caractère est modifié, la valeur de hachage sera très différente.

**À quoi servent les hachages ?**

Être capable de hacher une sortie unique de taille fixe pour une entrée de n'importe quelle longueur peut sembler rien de plus qu'une astuce obscure, mais les fonctions de hachage ont en fait un certain nombre d'utilisations.

Les fonctions de hachage sont un élément central de vérification de **signatures numériques. Le hachage permet la vérification de l'authenticité et de l'intégrité d'un fichier téléchargé, par exemple, sur Internet notamment**.

**Fonctions de hachage cryptographique communes :**

- **MD5** - Il s'agit d'une fonction de hachage qui a été publiée pour la première fois en 1991 par Ron Rivest. Il est désormais considéré comme non sécurisé et ne doit pas être utilisé à des fins cryptographiques. Malgré cela, il peut toujours être utilisé pour vérifier l'intégrité des données.

- **SHA-1** - L'algorithme de hachage sécurisé 1 est utilisé depuis 1995, mais il n'est plus considéré comme sûr depuis 2005, date à laquelle un certain nombre d'attaques par collision réussies ont eu lieu. Il est maintenant recommandé d'implémenter SHA-2 ou SHA-3 à la place.

- **SHA-2** - Il s'agit d'une famille de fonctions de hachage qui remplacent SHA-1. Ces fonctions contiennent de nombreuses améliorations qui les rendent sécurisées dans une grande variété d'applications. Malgré cela, SHA-256 et SHA-512 sont vulnérables aux attaques par extension de longueur, il existe donc certaines situations où il est préférable d'implémenter SHA-3.

- **SHA-3** - SHA-3 est le plus récent membre de la famille Secure Hash Algorithm, mais il est construit très différemment de ses prédécesseurs. À ce stade, il n'a pas encore remplacé SHA-2, mais offre simplement aux cryptographes une autre option qui peut améliorer la sécurité dans certaines situations.

- **RIPEMD** - RIPEMD est une autre famille de fonctions qui a été développée par la communauté universitaire. Il est basé sur de nombreuses idées de MD4 (le prédécesseur de MD5) et n'est limité par aucun brevet. Le RIPEMD-160 est toujours considéré comme relativement sûr, mais il n'a pas été largement adopté.

- **Tourbillon** - Whirlpool est une fonction de hachage de la famille des chiffrements à blocs carrés. Il est basé sur une modification d'AES et n'est soumis à aucun brevet. Il est considéré comme sûr, mais un peu plus lent que certaines de ses alternatives, ce qui a conduit à une adoption limitée.

## EXEMPLE DE HACHAGE

Si nous tentons de hacher le mot "*Mangeons*" en SHA-256, on obtient, par exemple, le « hash » suivant :
*5c79ab8b36c4c0f8566cee2c8e47135f2536d4f715a22c99fa099a04edbbb6f2*

Si nous changeons un caractère, cela change radicalement le hachage. Une faute de frappe comme "*Mengeons*» donne un résultat complètement différent :
*4be9316a71efc7c152f4856261efb3836d09f611726783bd1fef085bc81b1342*

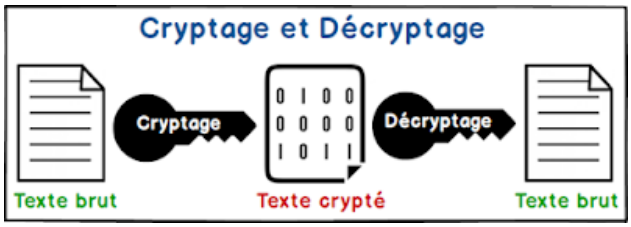
Contrairement au cryptage, nous ne pouvons pas mettre cette valeur de hachage via la fonction à l'envers pour obtenir à nouveau notre entrée. Bien que ces fonctions de hachage ne puissent pas être utilisées de la même manière que le cryptage, leurs propriétés en font un élément précieux des signatures numériques et de nombreuses autres applications.

### Exemple guidé de hachage

- Connectez-vous au site https://www.dcode.fr/fonction-hash
- Saisissez, par exemple, le mot bonjour et lancez le hashage avec l'algorithme « SHA256 »
- Téléchargez le freeware « HASH TAB » : https://www.clubic.com/telecharger-fiche56914-hashtab.html et installez-le sur votre PC
- Créez, sur votre bureau, un petit fichier texte
- Vérifiez l'empreinte de votre fichier avec Hash Tab
- Modifiez ce fichier et vérifiez le hashage SHA256 : il a été modifié par rapport à l'original.

## 2 - LE CRYPTAGE (OU CHIFFREMENT)

Pour le dire simplement, **le cryptage est le processus d'utilisation d'un code pour empêcher les autres parties d'accéder aux informations**. Lorsque les données ont été cryptées, seuls ceux qui ont la clé peuvent y accéder. Tant qu'un système suffisamment compliqué est utilisé, et qu'il est utilisé correctement, les attaquants ne peuvent pas voir les données.




**Les données sont chiffrées avec des algorithmes de chiffrement.**

**IMPORTANT**

*L'une des distinctions les plus importantes entre **le cryptage** (chiffrement) et le hachage est que le cryptage (chiffrement) **est conçu pour aller dans les deux sens**. Cela signifie qu'une fois que quelque chose a été chiffré avec une clé, il peut également être déchiffré (avec une clé).*

Cela rend le cryptage utile dans une série de situations, comme pour stocker ou transférer des informations en toute sécurité : exemple des mails chiffrés sous Thunderbird avec OPEN PGP. **Une fois les données correctement cryptées, elles sont considérées comme sécurisées et ne sont accessibles qu'à ceux qui détiennent la clé** (voir Labo Cyber Open PGP).

Cette fonctionnalité permet aux personnes qui ne se sont jamais rencontrées de communiquer en toute sécurité. Le chiffrement à clé publique est également un élément important des signatures numériques, qui sont utilisées pour valider l'authenticité et l'intégrité des données et des messages.

**Algorithmes de chiffrement courants**

- **Chiffre César -** Il s'agit d'un code simple qui implique que chaque lettre soit décalée d'un nombre fixe de places. Si un chiffre César a un décalage de trois, chaque "a" deviendra un "d", chaque "b" deviendra un "e", chaque "c" deviendra un "f" et ainsi de suite. Il porte le nom de Julius Caesar, qui fut la première personne enregistrée à utiliser le schéma.

- **AES** - La norme de chiffrement avancé est un algorithme complexe à clé symétrique qui sécurise une partie importante de nos communications modernes. Il implique un certain nombre d'étapes sophistiquées et est souvent utilisé pour chiffrer les données dans TLS, les applications de messagerie, au repos et dans de nombreuses autres situations.

- **3DES** - Triple DES est basé sur l'algorithme DES. Lorsque la puissance informatique croissante a rendu le DES insécurisé, 3DES a été développé comme un algorithme renforcé. Dans 3DES, les données sont exécutées via l'algorithme DES trois fois au lieu d'une seule, ce qui rend plus difficile le crack. 3DES peut être utilisé pour plusieurs des mêmes choses que AES, mais seules certaines implémentations sont considérées comme sûres.

- **RSA** - Le chiffrement Rivest-Shamir-Adleman a été la première forme de cryptographie à clé publique largement utilisée. Il permet aux entités de communiquer en toute sécurité même si elles ne se sont pas rencontrées ou n'ont pas eu la possibilité d'échanger des clés. Il peut être utilisé dans un certain nombre de protocoles de sécurité différents, tels que PGP et TLS.

- **ECDSA** - L'algorithme de signature numérique à ***courbe elliptique*** est une variante du DSA qui utilise la cryptographie à courbe elliptique. En tant qu'algorithme à clé publique, il peut être appliqué dans des situations similaires à RSA, bien qu'il soit moins couramment mis en œuvre en raison de certains problèmes de sécurité.

## EXEMPLE DE CRYPTAGE (chiffrement)

Pour vous donner une idée du fonctionnement du chiffrement dans la pratique, nous utiliserons le « chiffre César » comme exemple. Si nous voulions crypter un message de "*Mangeons*", avec un décalage de trois lettres, le "*L*" deviendrait un "O", les "E" deviendraient un "H", etc…

Cela nous donnerait : ***ohwvhdw***

Pour déchiffrer le message, le destinataire doit savoir que l'algorithme de chiffrement implique un décalage de trois lettres, puis recule chaque lettre de trois emplacements. Si nous le voulions, nous pourrions faire varier le code en décalant chaque lettre d'un nombre différent. Nous pourrions même utiliser un algorithme beaucoup plus sophistiqué.

Exemple guidé (Code Cesar) :

- Connectez-vous au site https://www.dcode.fr/chiffre-cesar
- Testez le codage (code Cesar) de votre prénom et de votre nom
- Vérifiez que vous pouvez décoder

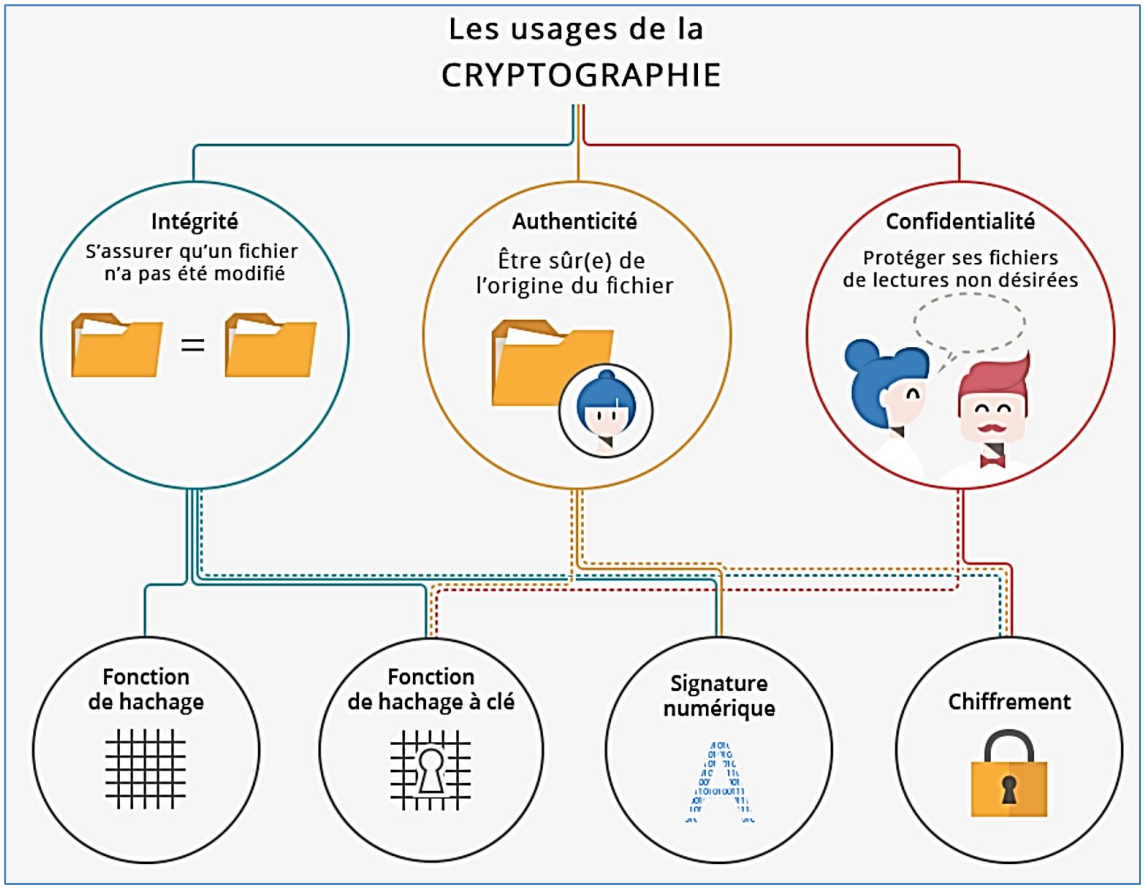
Il existe des outils en ligne pour encoder avec des algorithmes puissants tels que : https://www.devglan.com/online-tools/aes-encryption-decryption

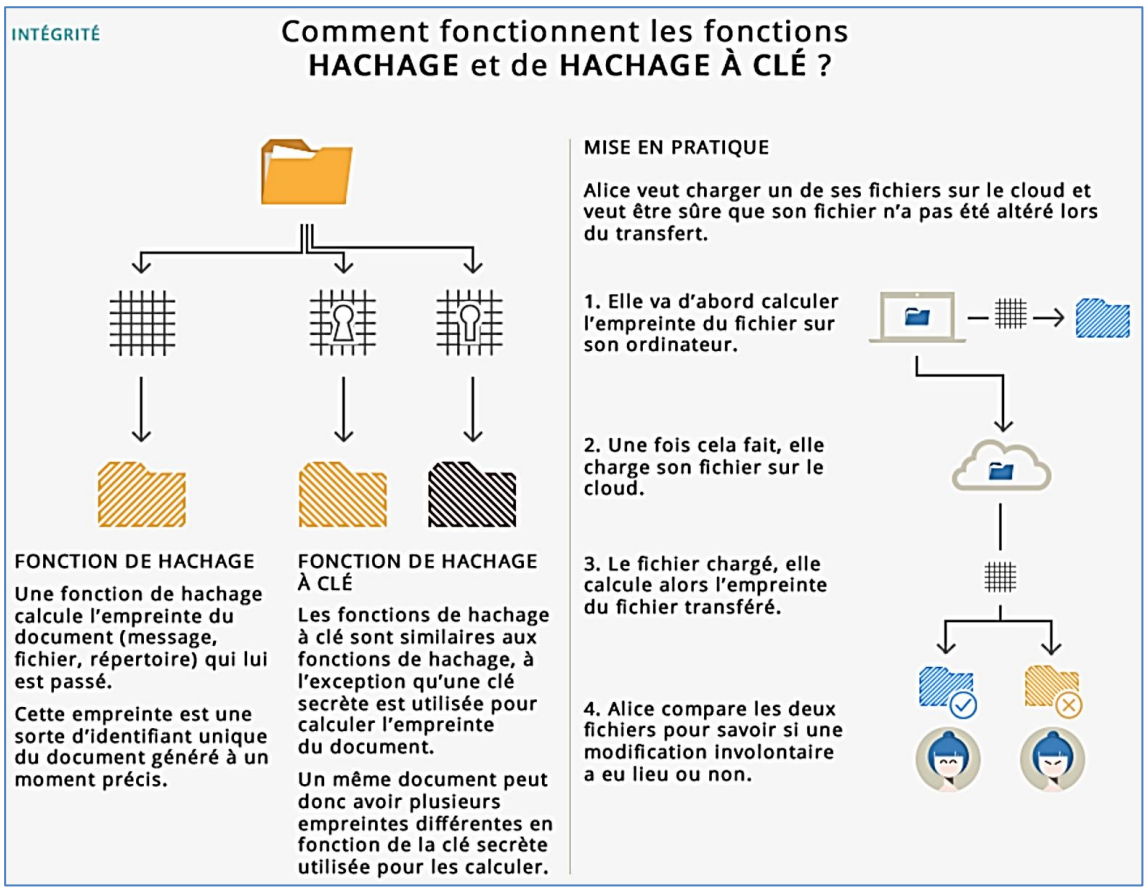
## EN RESUME

**Le cryptage est le processus de codage des informations pour les protéger**. Lorsque les données sont cryptées, elles ne peuvent être décryptées et accessibles que par ceux qui ont la bonne clé. Les algorithmes de chiffrement sont réversibles, ce qui nous permet de garder nos données à l'abri des attaquants, mais de pouvoir y accéder quand nous en avons besoin. Il est largement utilisé pour assurer notre sécurité en ligne, jouant un rôle crucial dans bon nombre de nos protocoles de sécurité qui protègent nos données lorsqu'elles sont stockées et en transit..

En revanche, **le hachage est un processus à sens unique**. Lorsque nous hachons quelque chose, nous ne voulons pas pouvoir le remettre dans sa forme originale. Les fonctions de hachage cryptographique ont un certain nombre de propriétés uniques qui **nous permettre de prouver l'authenticité et l'intégrité des données**, comme par le biais de signatures numériques et de codes d'authentification de message.

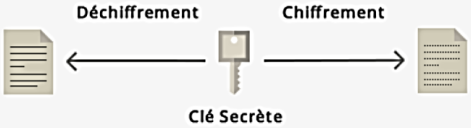
## INFOGRAPHIE (source : CNIL)



CONFIDENTIALITÉ

# Comment fonctionne le CHIFFREMENT ?




## CHIFFREMENT SYMÉTRIQUE

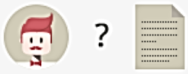
Le chiffrement symétrique permet de chiffrer et déchiffrer un fichier avec la même clé, dite secrète. Pour s'échanger un message il faut donc que les deux parties partagent la même clé.

### MISE EN PRATIQUE

Alice vient d'enregistrer la liste des cadeaux de Noël de sa famille sur l'ordinateur familial. Elle souhaite être la seule à pouvoir y accèder.

1. Pour ce faire, Alice chiffre la liste en utilisant sa clé secrète.

2. Plus tard dans la journée, Bob trouve la liste et cherche à l'ouvrir.



3. Malheureusement pour lui, Bob est incapable de lire la liste car il ne possède pas la clé secrète.

4. La liste est donc bien protégée. Seule Alice peut réussir à la déchiffrer et la lire !


## CHIFFREMENT ASYMÉTRIQUE

Le chiffrement asymétrique repose sur l'utilisation d'une paire de clés : une publique et une privée.

La clé publique, accessible à tous, est utilisée pour chiffrer les fichiers. Seule la clé privée permet de déchiffrer ces fichiers, celle-ci étant connue que d'un seul individu.

### MISE EN PRATIQUE

Alice, hackeuse, vient de découvrir des informations d'intérêt public. Elle veut les transmettre à Bob, journaliste, pour qu'il enquête.

1. Alice vient de récupérer la clé publique de Bob. Elle l'utilise pour chiffrer son document.

2. Elle l'envoie à Bob.

3. Bob reçoit le document et le déchiffre à l'aide de sa clé privée.

4. Une fois le document déchiffré, il rédige un article puis le publie dans son journal.

AUTHENTICITÉ

# Comment fonctionnent les SIGNATURES NUMÉRIQUES ?


## SIGNATURE NUMÉRIQUE

Ce procédé cryptographique permet à toute personne de s'assurer de l'identité de l'auteur d'un document et permet en plus d'assurer que celui-ci n'a pas été modifié.

Le procédé repose sur un couple de clés : l'une est privée et connue uniquement de son détenteur, l'autre est publique et accessible à tous.

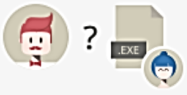
La signature est générée en utilisant la clé privée. La clé publique est utilisée pour vérifier cette signature. Cette vérification peut donc être effectuée par n'importe quelle personne ayant accès à la clé publique.

### MISE EN PRATIQUE

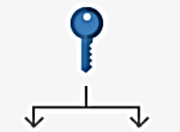
Alice vient de publier un nouveau logiciel et souhaite assurer à ses futurs utilisateurs l'authenticité des copies qu'ils obtiennent.

1. Avant de publier librement son logiciel, Alice prend soin de le signer.

2. Bob vient de télécharger une copie du logiciel mais il veut s'assurer que cette copie provient bien d'Alice.



3. Bob utilise la clé publique d'Alice pour vérifier la signature de la copie.



4. Si la clé reconnait la signature, alors c'est une bonne copie !
Dans le cas contraire, Bob préfère ne pas prendre de risques. Il supprimera la copie.