



CRYPTO-JAVA : IMPLÉMENTATION DES FONCTIONS CRYPTOGRAPHIQUES EN JAVA

OUMAR DIALLO

**Master 2 Transmission de Données et Sécurité de l'Information
Math-Crypto-Sécurité**

Rapport d'examen de : Crypto-Java.

Sous la direction de : Dr. Demba Sow

Octobre 2023



Résumé du Rapport

Ce rapport présente des captures de quelques fonctionnalités cryptographiques implémentées en Java (CryptoJava) ainsi que des explications de ces captures. L'interface graphique a été réalisée avec Java Swing afin de bien visualiser le résultat de ces fonctions implémentées.

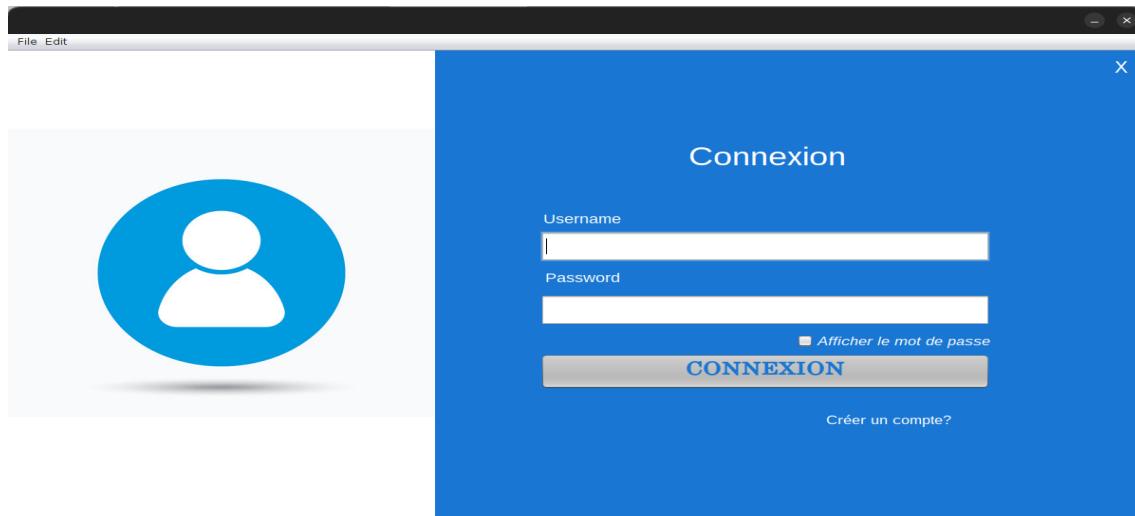
Table des matières

| | |
|---|-----------|
| Introduction | 2 |
| 1 Génération de clés | 4 |
| 1.1 Symétriques | 4 |
| 1.2 Asymétriques | 6 |
| 2 Chiffrement | 8 |
| 2.1 Symétrique | 8 |
| 2.2 Asymétriques | 10 |
| 3 Déchiffrement | 12 |
| 3.1 Symétrique | 12 |
| 3.2 Asymétriques | 14 |
| 4 Hachage | 16 |
| 4.1 Message Digest (MD) | 16 |
| 4.2 Message Authentication Code (MAC) | 18 |
| 4.3 Vérification d'une empreinte | 20 |
| 5 Signature | 24 |
| 5.1 Signature | 24 |
| 5.2 Vérification | 26 |
| 6 Echange de clés Diffie-Hellman | 28 |
| 6.1 Échange de Clés avec 2 parties | 28 |
| 6.2 Échange de Clés avec 3 parties | 31 |
| 7 Mode Live | 33 |
| 7.1 Chiffrement | 33 |
| 7.2 Déchiffrement | 34 |
| 7.3 Hachage | 35 |
| Conclusion | 36 |

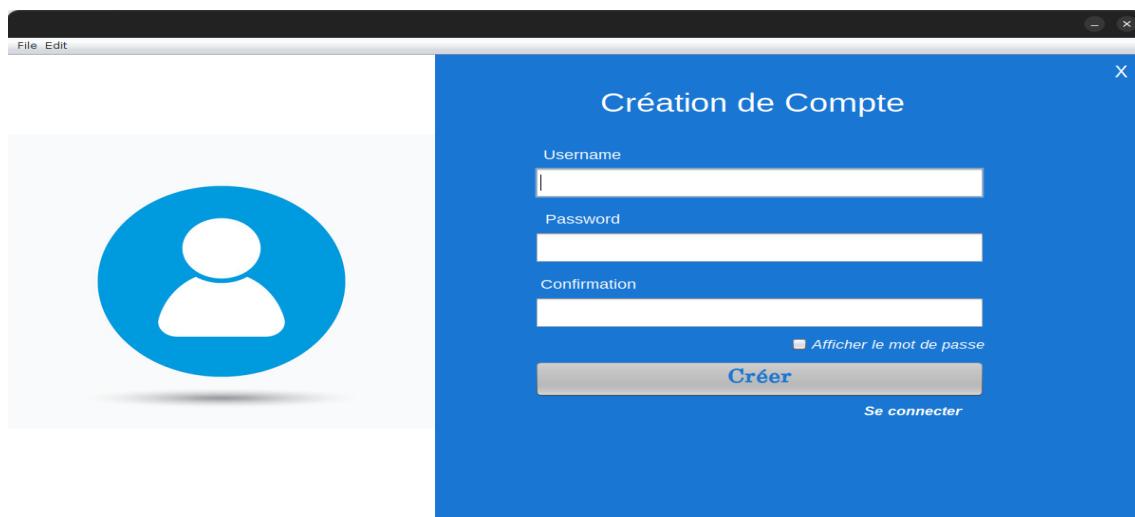
Introduction

L'avènement de la technologie a apporté de nombreux avantages, mais elle a également soulevé des préoccupations majeures en matière de sécurité des données. La confidentialité et l'intégrité des informations sont devenues des aspects critiques dans le monde numérique d'aujourd'hui. C'est dans ce contexte que la cryptographie, la science de la sécurisation des informations, joue un rôle essentiel. De plus, les applications logicielles, en particulier celles impliquant des données sensibles, nécessitent des mécanismes de cryptographie robustes pour protéger les informations contre les menaces. Dans ce rapport, nous explorerons une application conçue pour offrir des fonctionnalités de cryptographie, développée en utilisant Crypto Java, un langage de programmation puissant et polyvalent, ainsi que Java Swing, une bibliothèque graphique permettant de créer des interfaces utilisateur conviviales. Nous examinerons en détail les fonctionnalités de cryptographie de cette application et les avantages qu'elle offre en matière de sécurité des données.

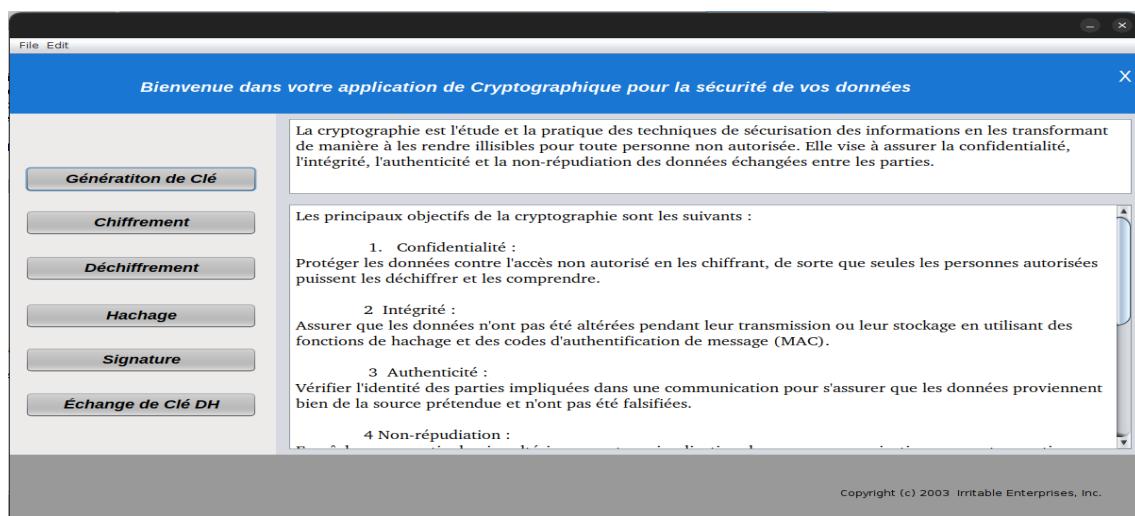
La page de Login



La page de création de compte



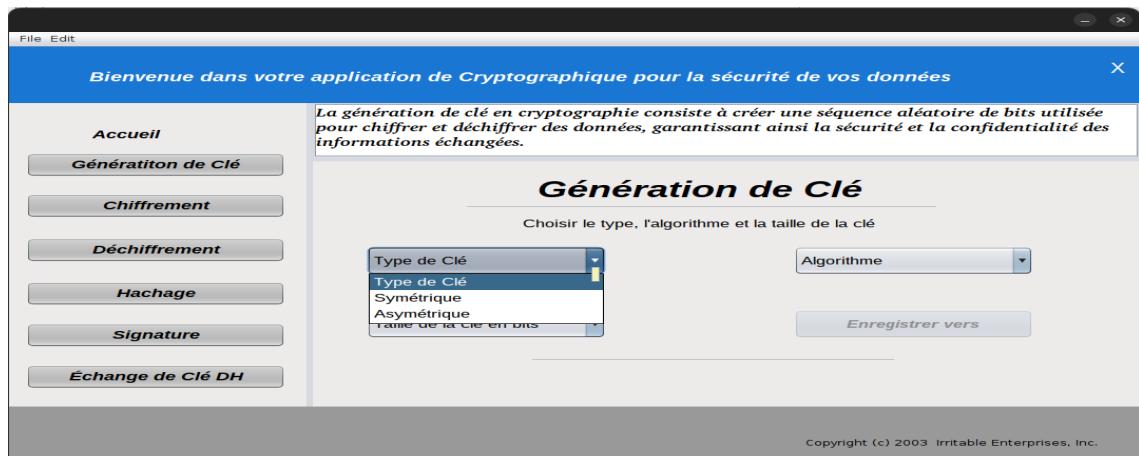
La page d'accueil



Chapitre 1

Génération de clés

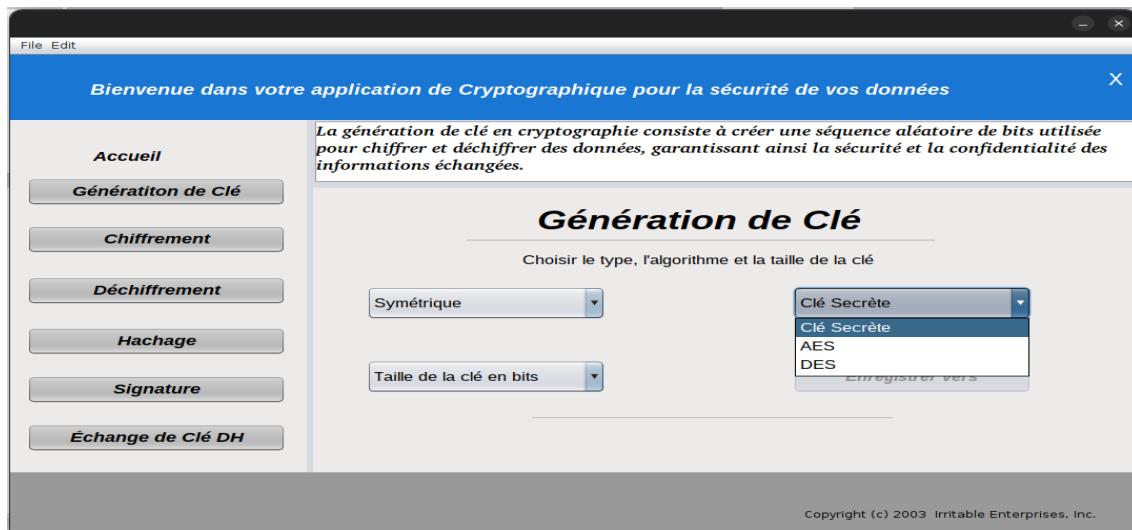
La génération de clés se procède de 4 étapes selon le type d'algorithme choisi (Symétrique ou Asymétrique) :



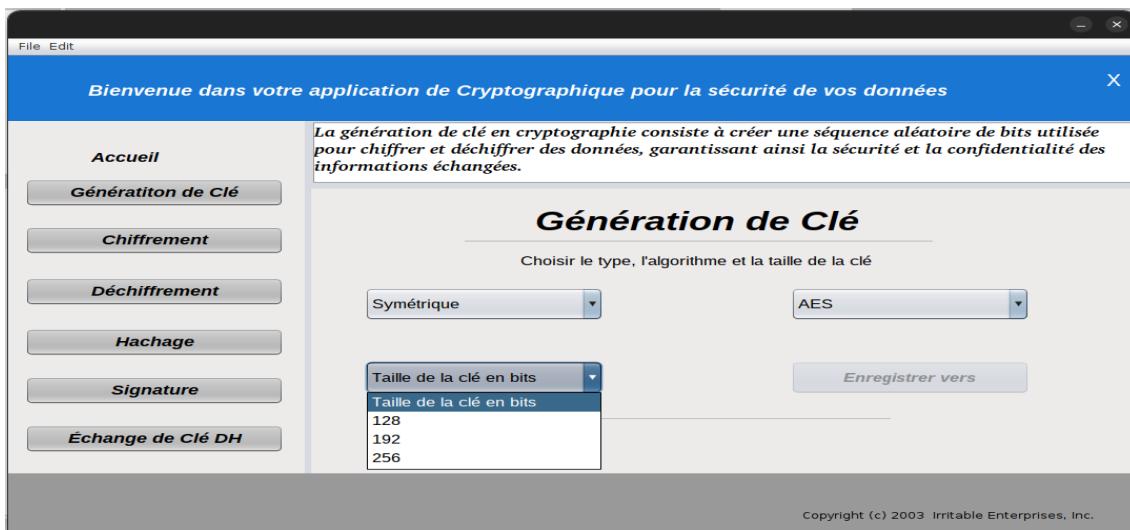
1.1 Symétriques

Pour les algorithmes symétriques :

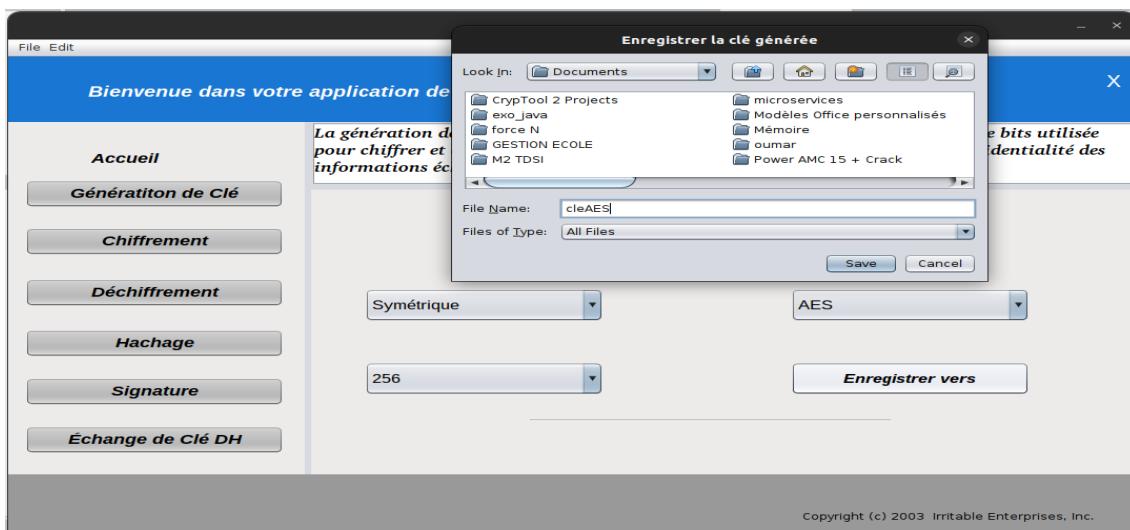
- On choisie d'abord l'algorithme pour une clé secrète



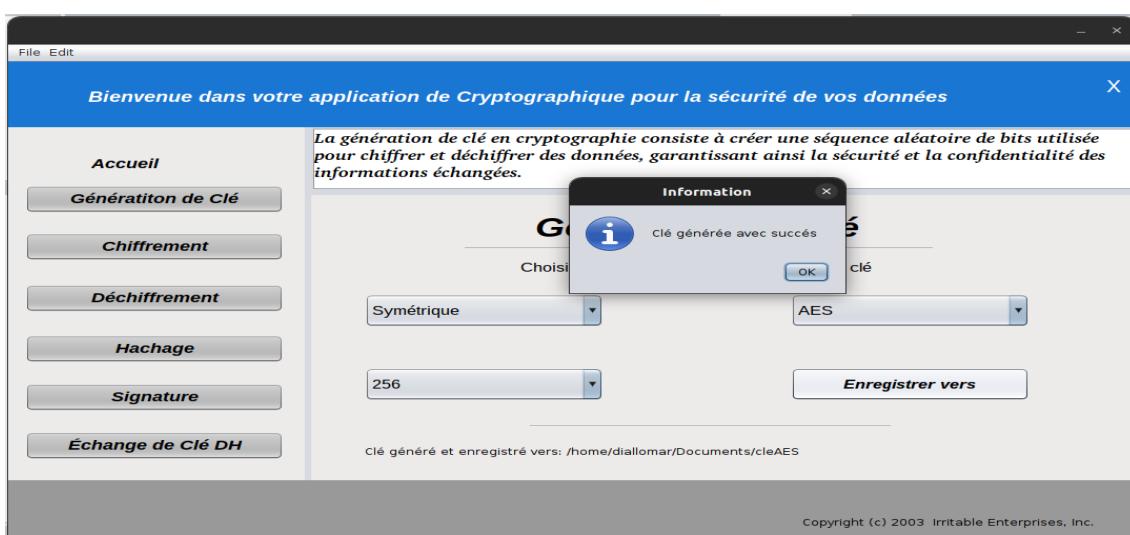
- Ensuite on choisie la taille de la clé



- Puis on indique un chemin d'enregistrement en cliquant sur *enregistrer vers*



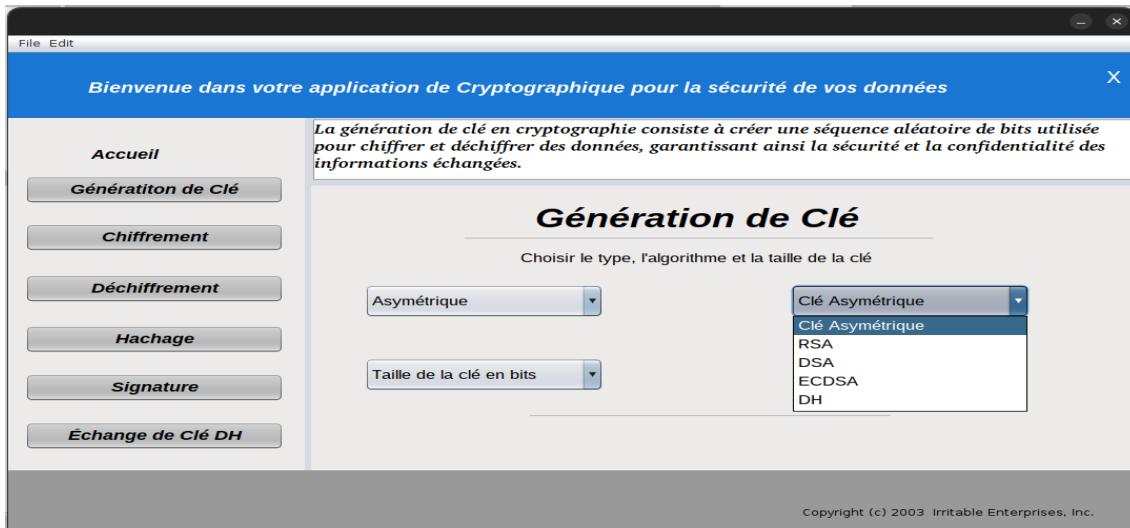
- Et en fin on génère et enregistre la clé vers le chemin indiqué.



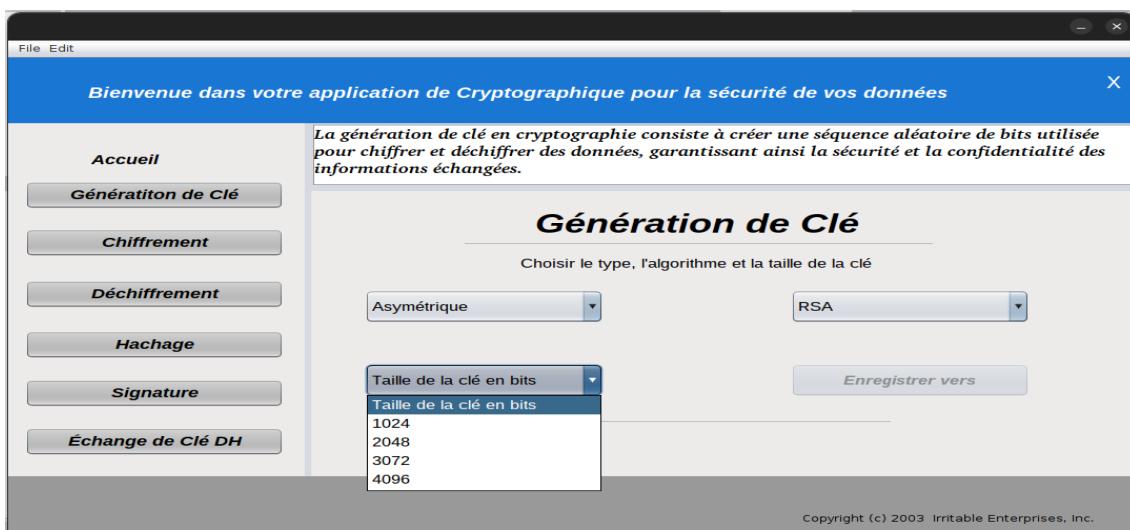
1.2 Asymétriques

De même pour les algorithmes asymétriques :

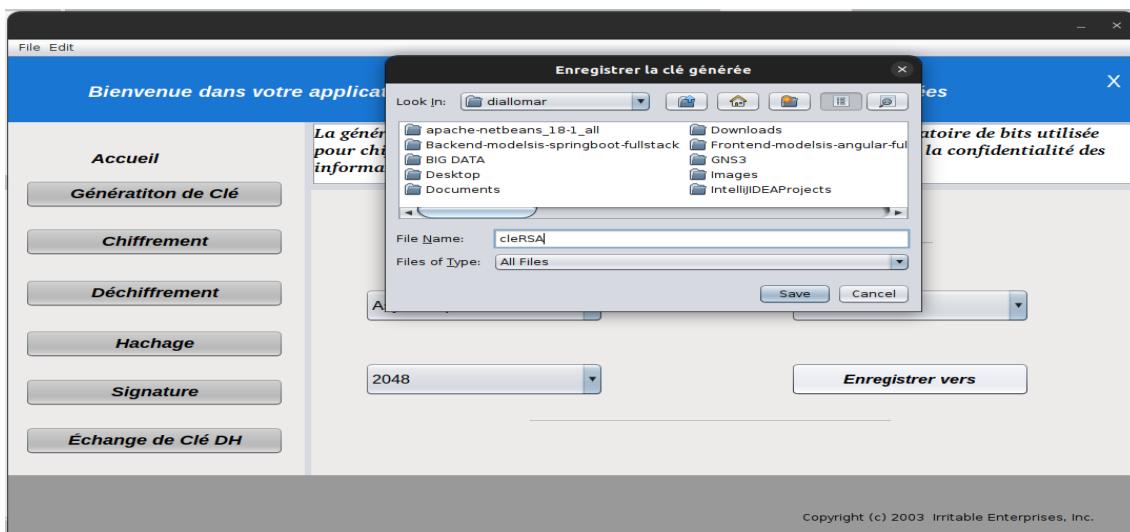
- On choisie d'abord l'algorithme pour une paire de clé asymétrique



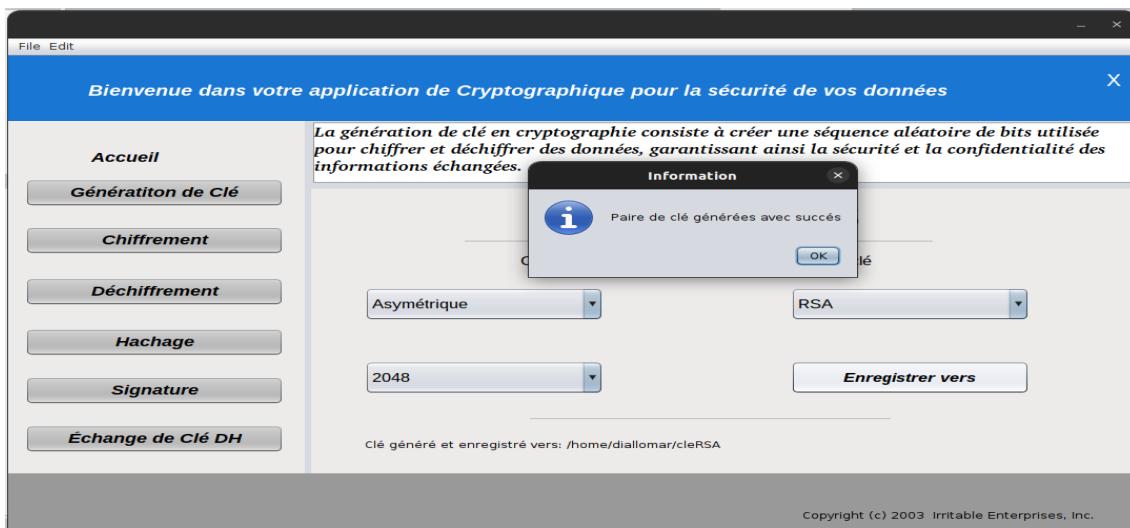
- Ensuite on choisie la taille de la paire de clé



- Puis on indique un chemin d'enregistrement *enregistrer vers*



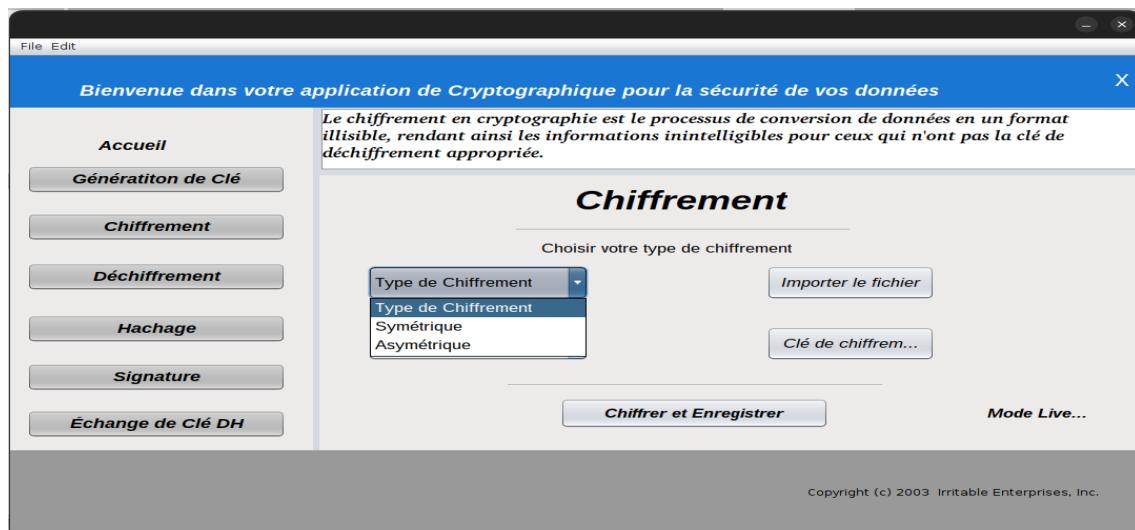
- Et en fin on génère et enregistre la paire de clé le chemin indiqué.



Chapitre 2

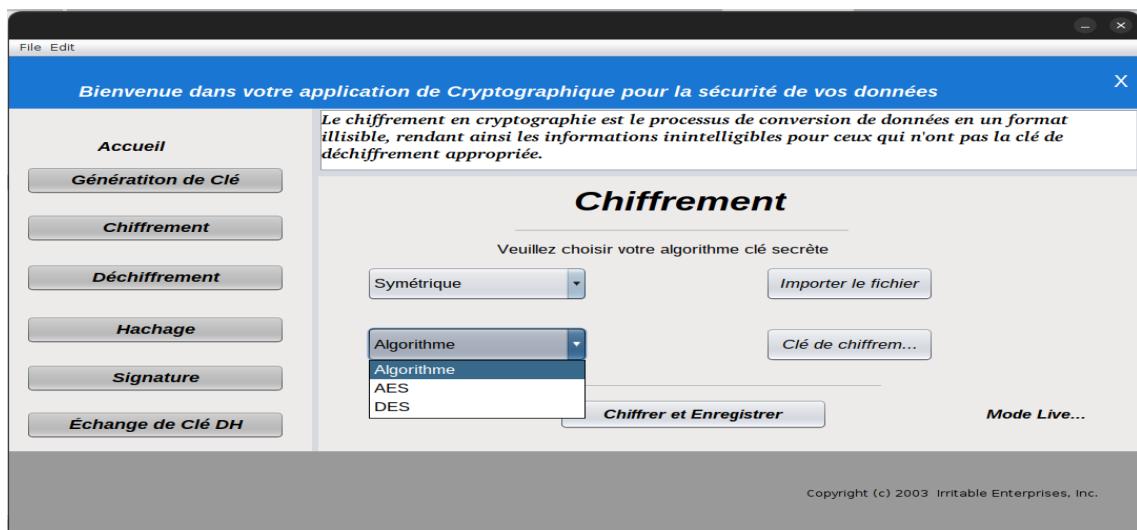
Chiffrement

Le chiffrement se procéde en 4 étapes selon le type d'algorithme choisi (Symétrique ou Asymétrique) :

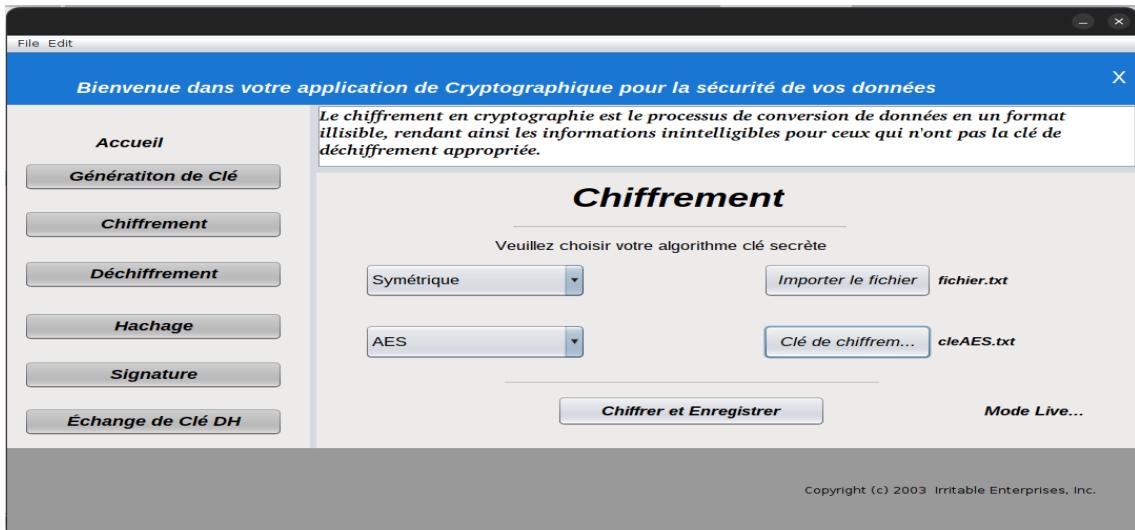


2.1 Symétrique

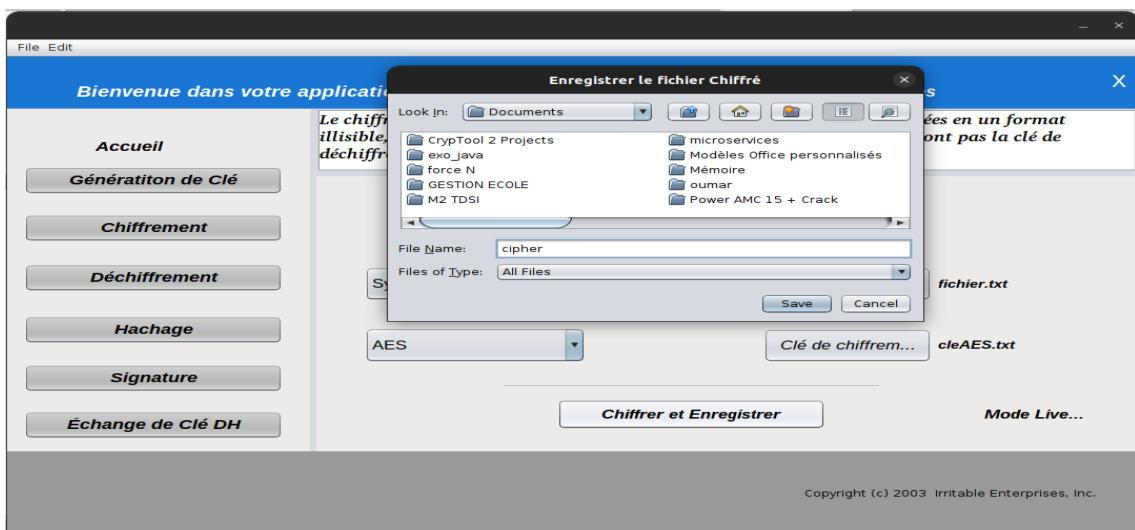
- On choisie d'abord l'algorithme utilisé lors de génération de la clé secrète.



- Ensuite on importe le fichier clair et la clé secrète.



- Puis on indique un chemin d'enregistrement



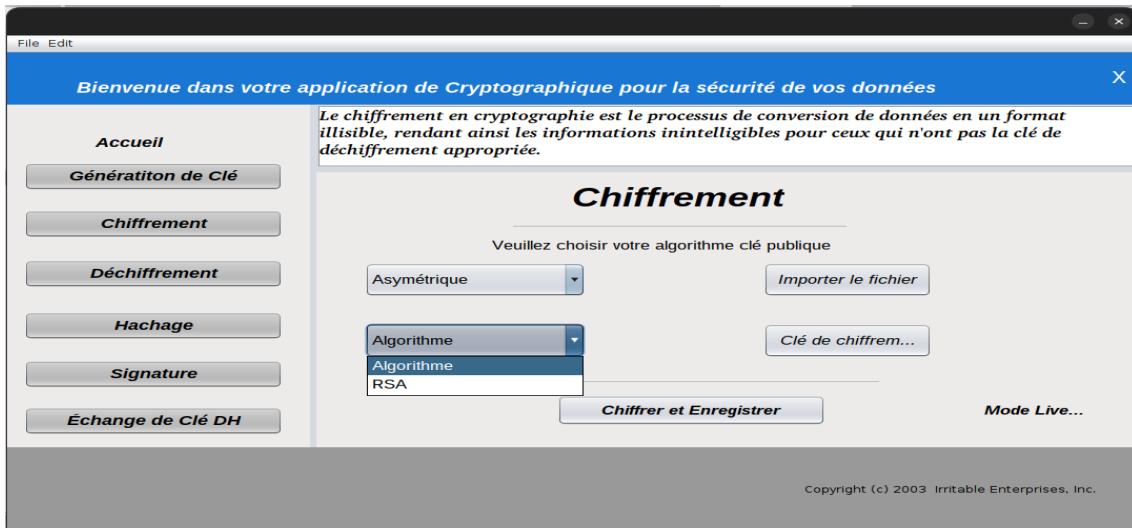
- Et en fin on chiffre et on enregistre vers le chemin indiqué.



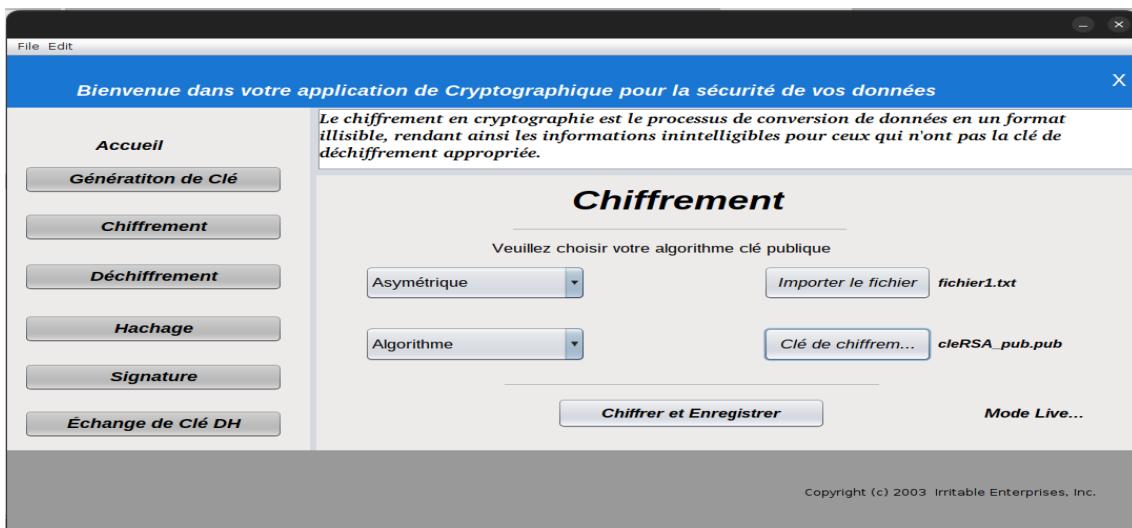
2.2 Asymétriques

De même que le chiffrement symétrique

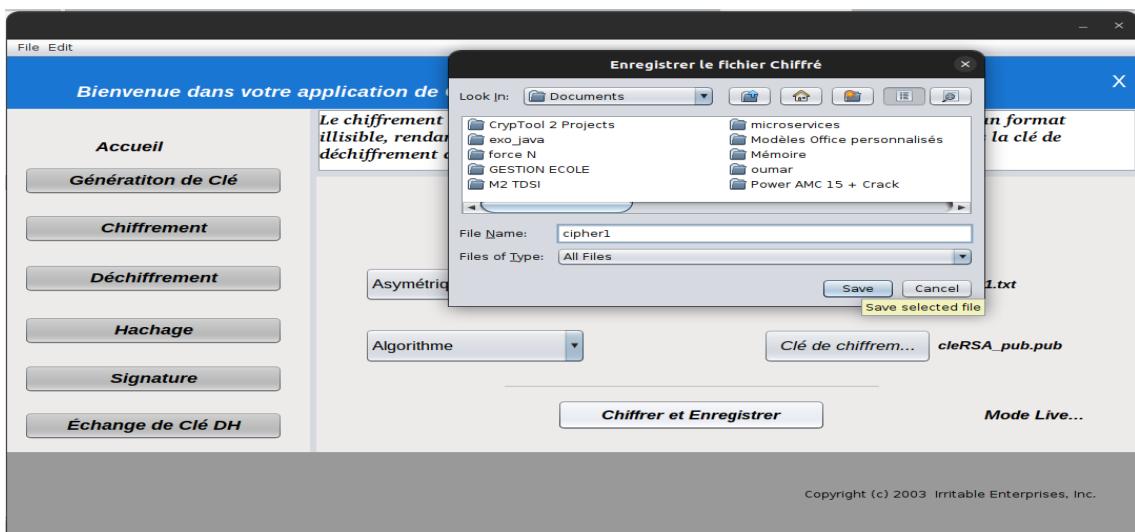
- On choisit d'abord l'algorithme utilisé lors de génération de la paire de clé



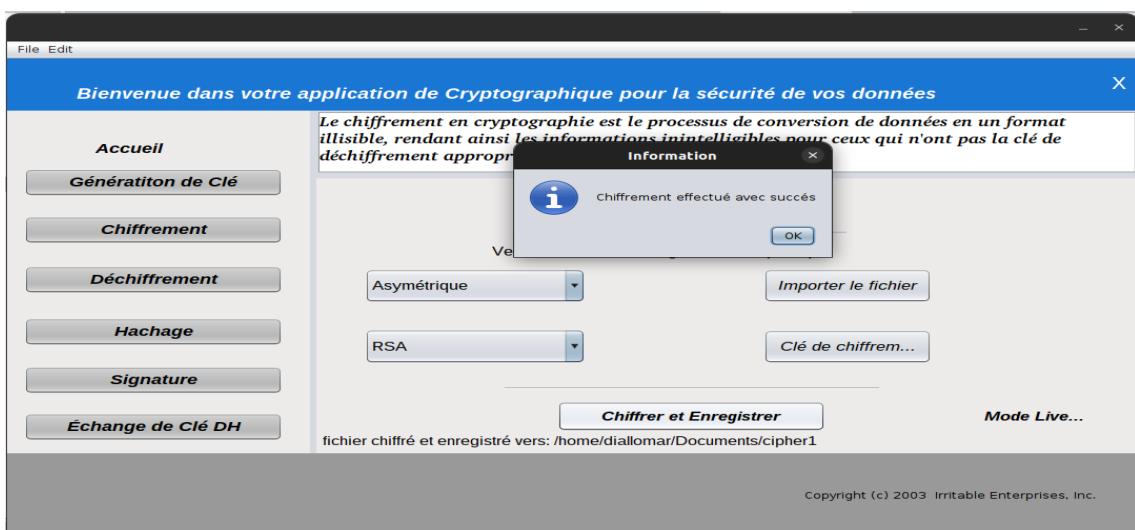
- Ensuite on importe le fichier clair et la clé publique.



- Puis on indique un chemin d'enregistrement



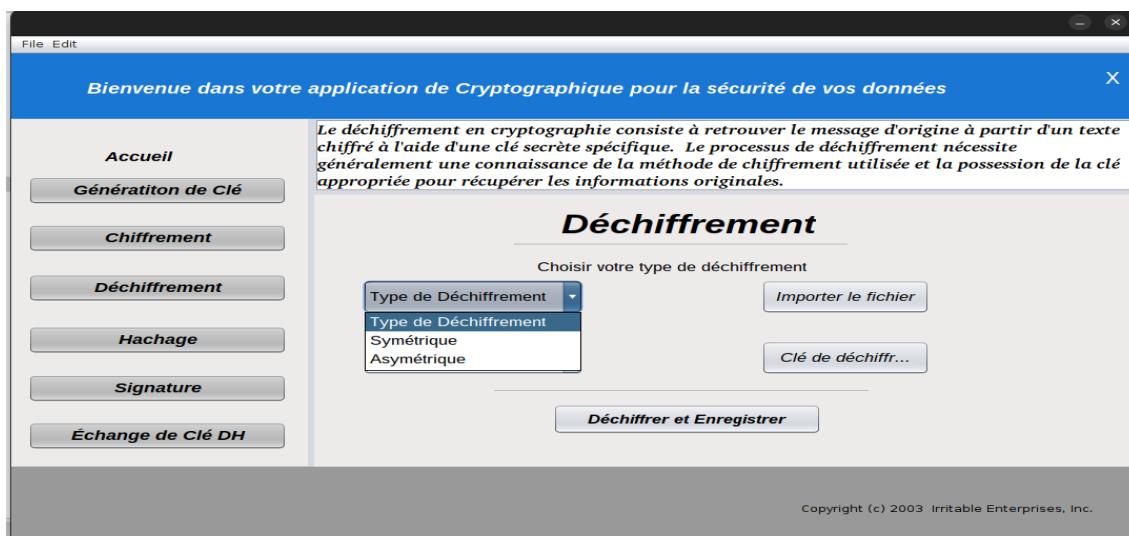
- Et en fin on chiffre et on enregistre vers le chemin indiqué.



Chapitre 3

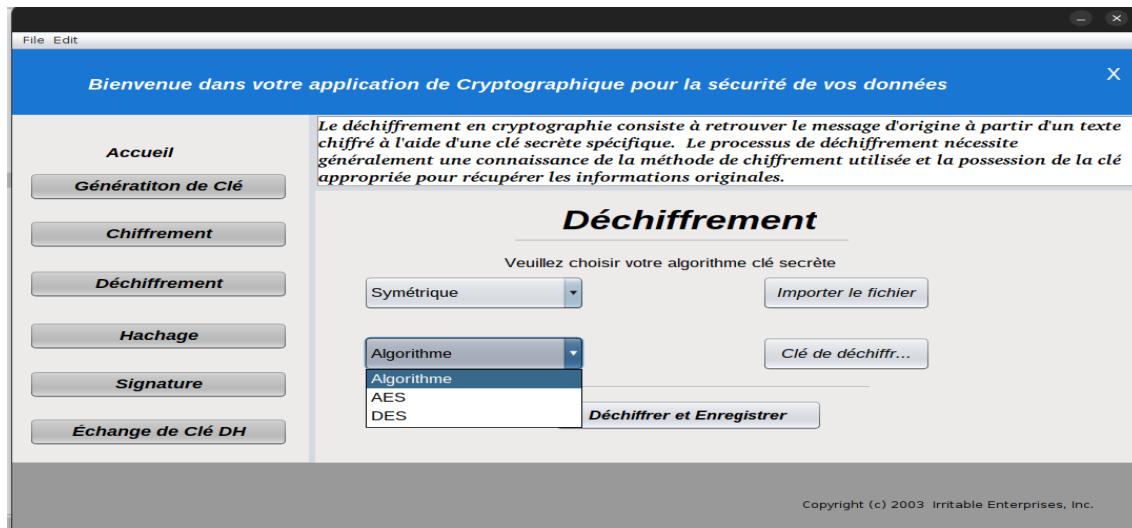
Déchiffrement

De même que le chiffrement, le déchiffrement se procède aussi en 4 étapes selon le type d'algorithme choisi (Symétrique ou Asymétrique) :

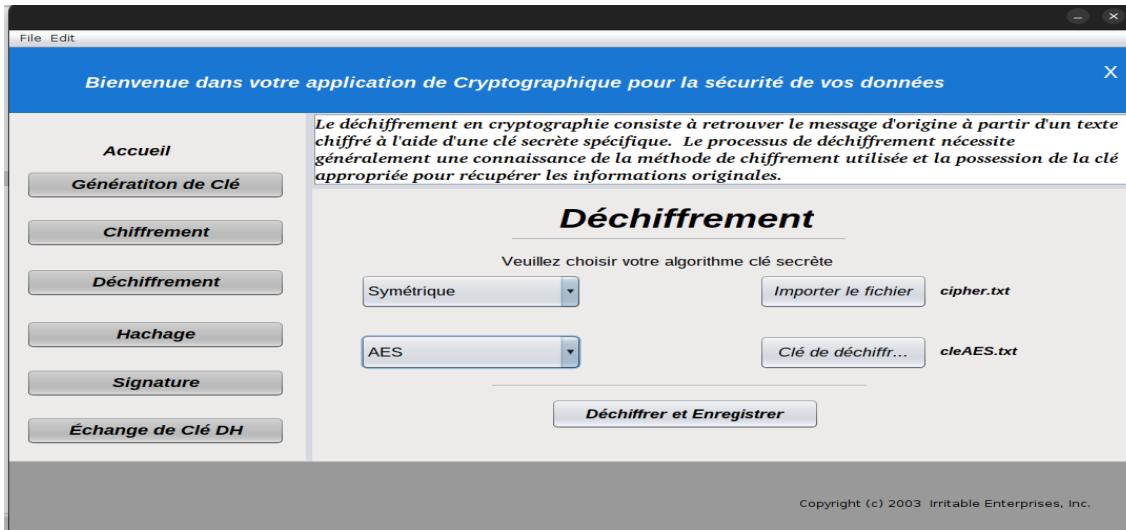


3.1 Symétrique

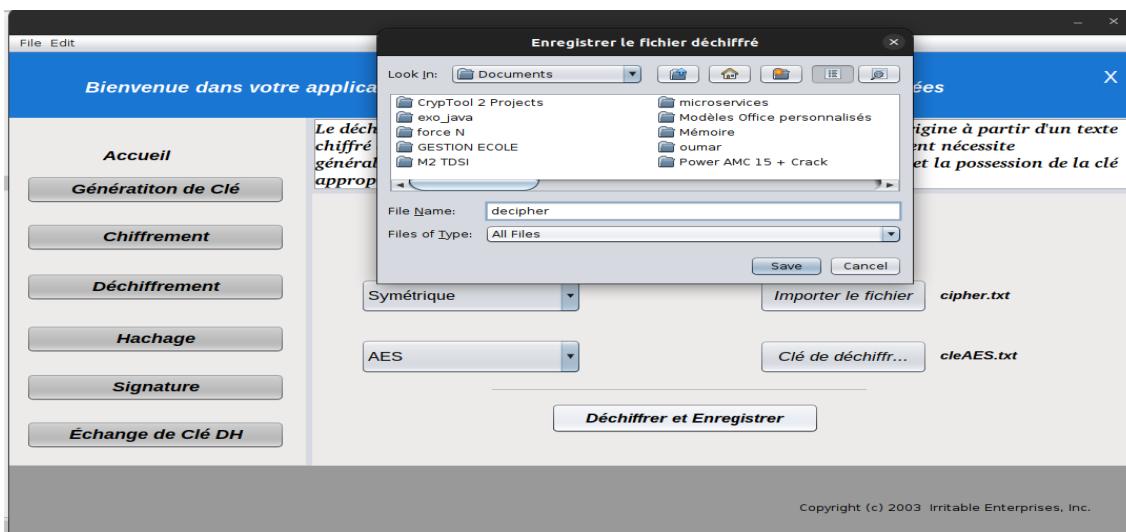
- On choisie d'abord l'algorithme utilisé lors du chiffrement.



— Ensuite on importe le fichier chiffré et la clé secrète.



— Puis on indique un chemin d'enregistrement



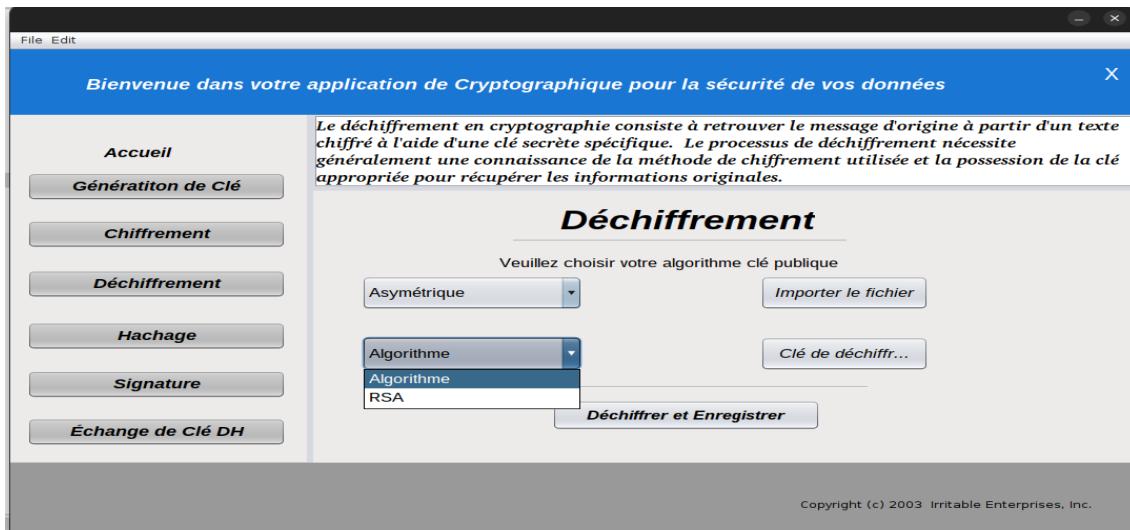
— Et en fin on déchiffre et on enregistre vers le chemin indiqué.



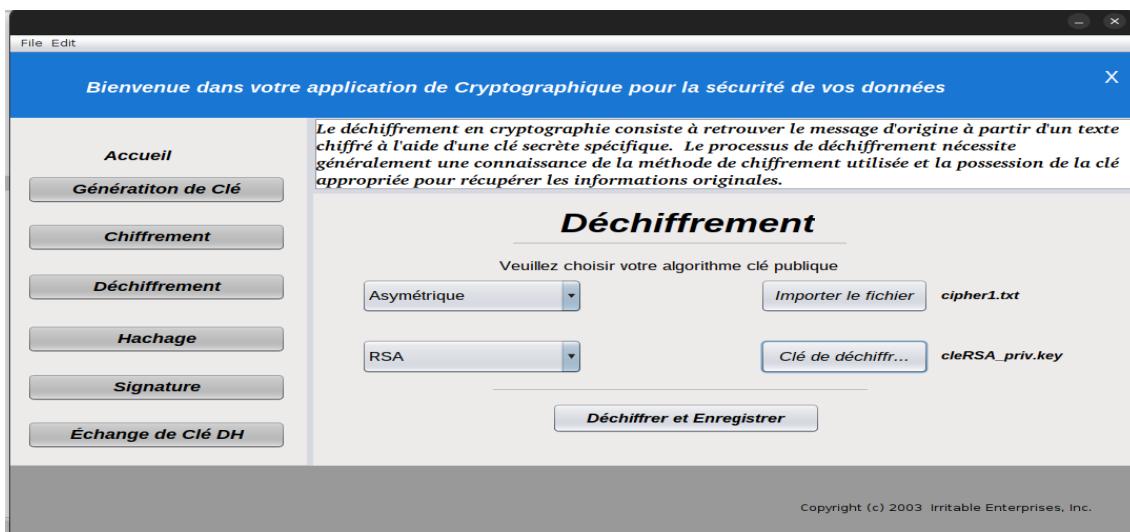
3.2 Asymétriques

De même que le déchiffrement symétrique :

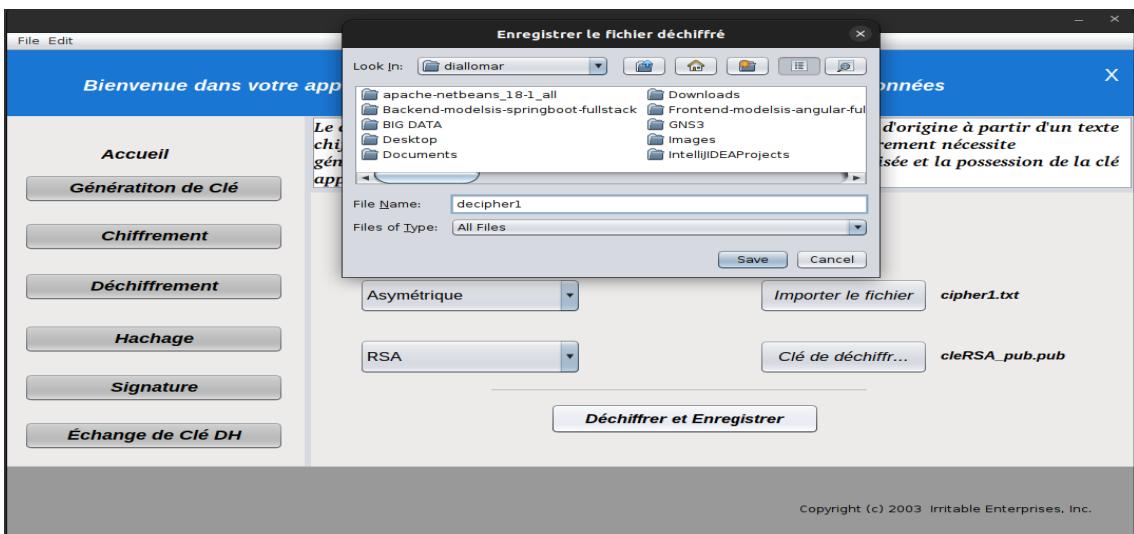
- On choisit d'abord l'algorithme utilisé lors du chiffrement.



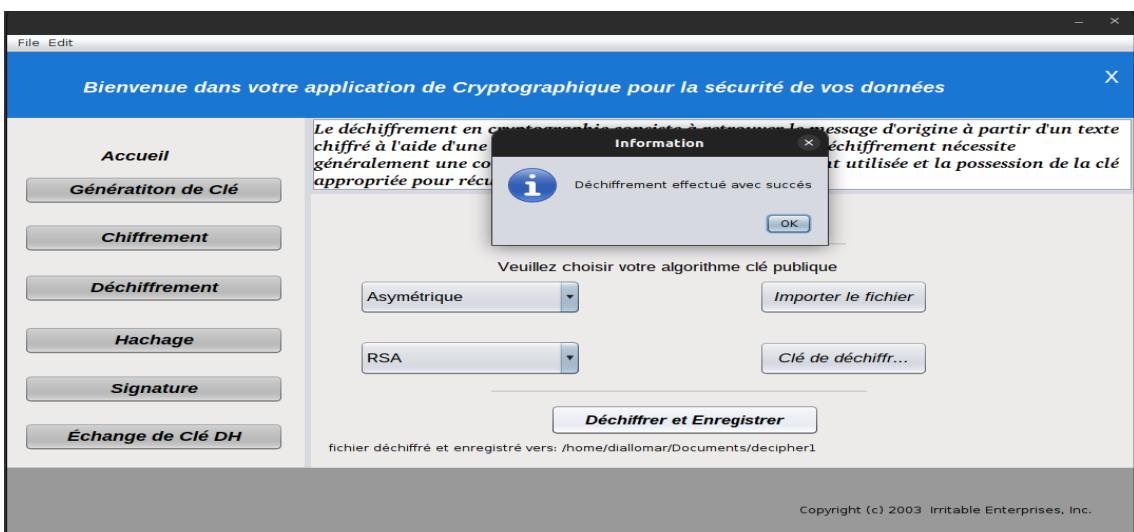
- Ensuite on importe le fichier chiffré et la clé privée.



- On indique le chemin d'enregistrement.



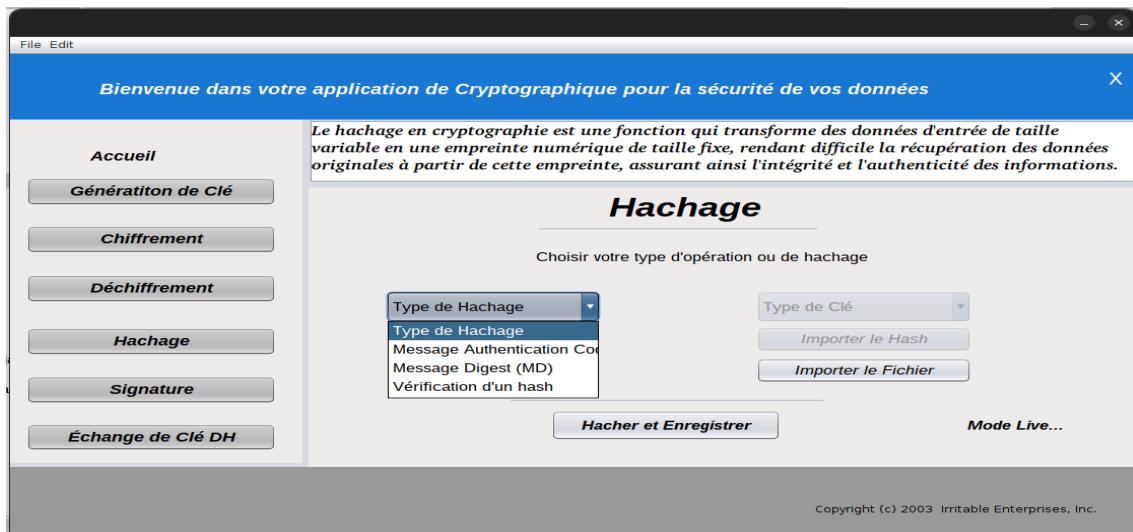
- Et en fin on déchiffre et on enregistre vers le chemin indiqué.



Chapitre 4

Hachage

Dans l'application nous présentons deux façon de hacher un fichier et la vérification de cette empreinte selon le type de hachage choisi (MD ou MAC) :



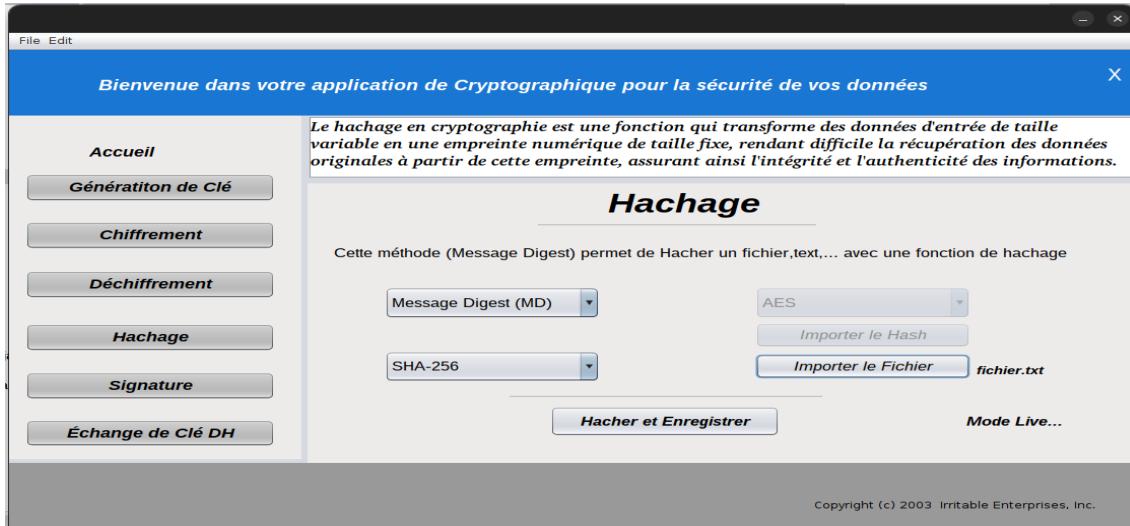
4.1 Message Digest (MD)

Voici comment hacher un fichier avec la méthode MD (Message Digest) :

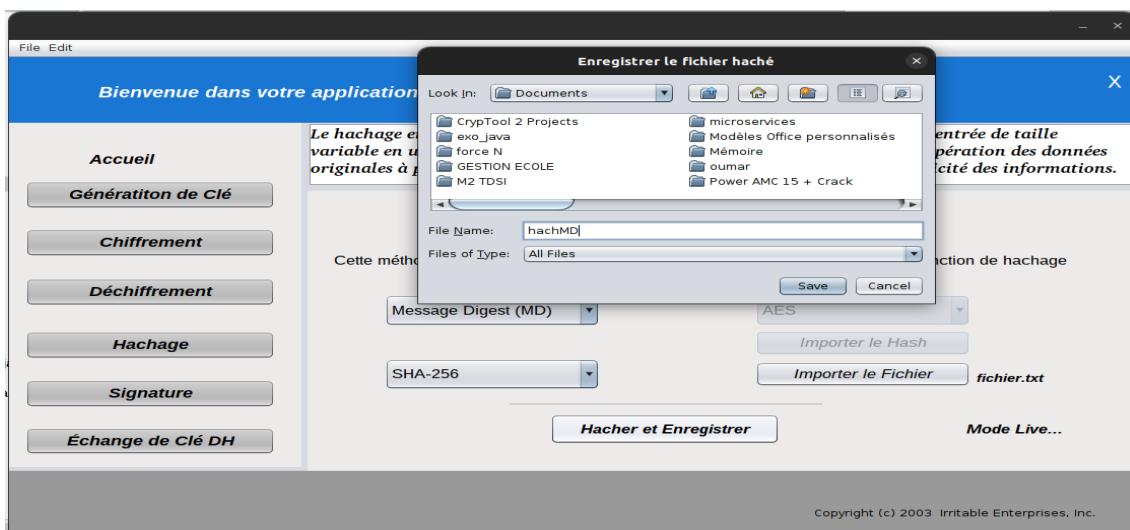
- On Choisie d'abord la fonction de hachage



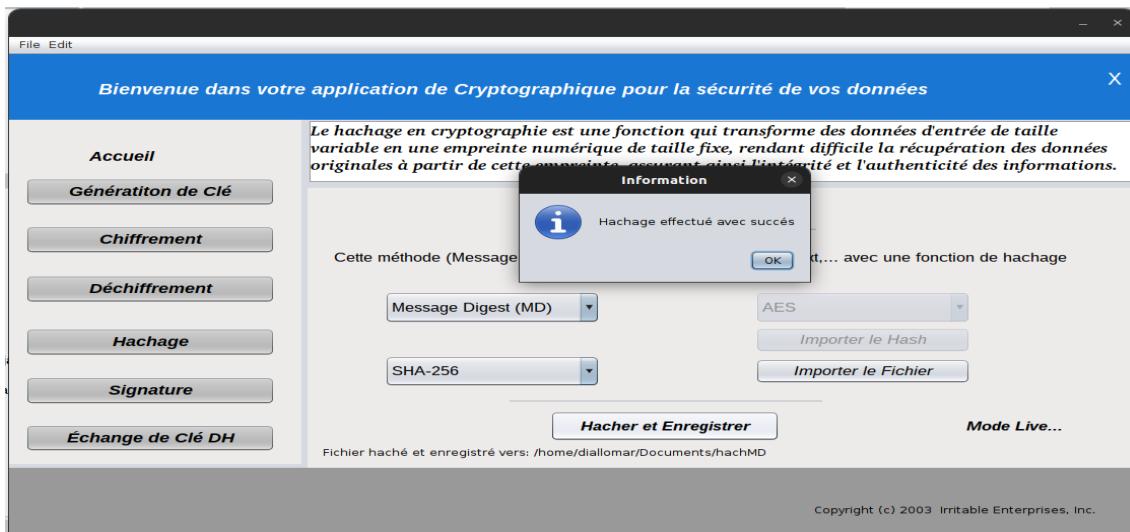
— Ensuite on importe le fichier à hacher.



— On indique un chemin d'enregistrement.



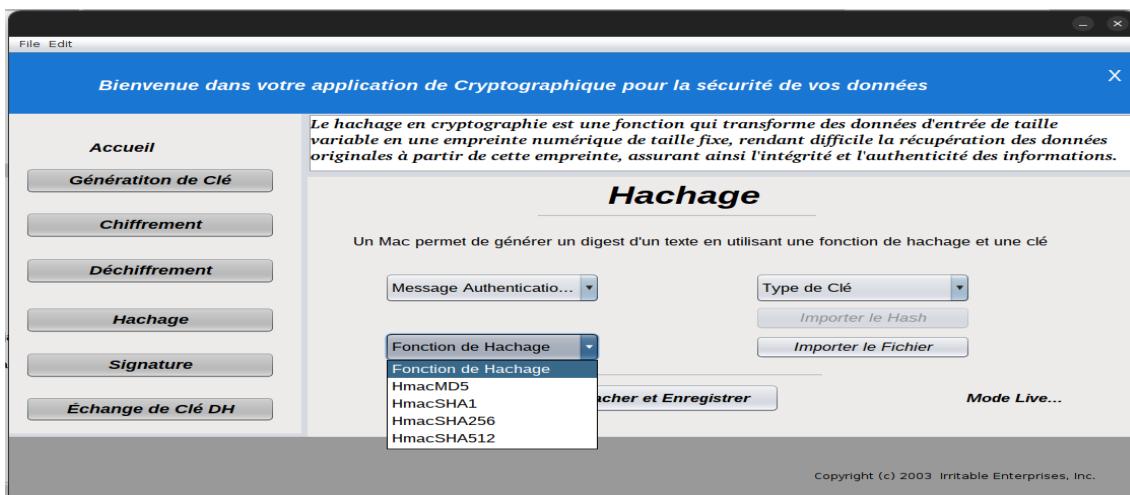
- Et en fin on hache et enregistre l'empreinte vers le chemin indiqué.



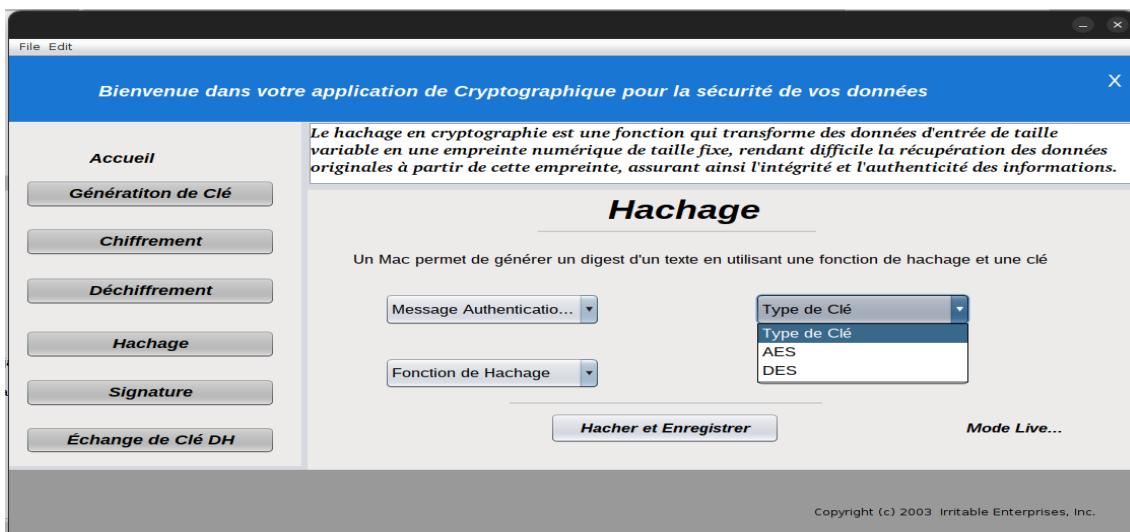
4.2 Message Authentication Code (MAC)

De même voici comment hacher un fichier avec la méthode MAC (Message Authentication Code) :

- On Choisie d'abord la fonction de hachage.



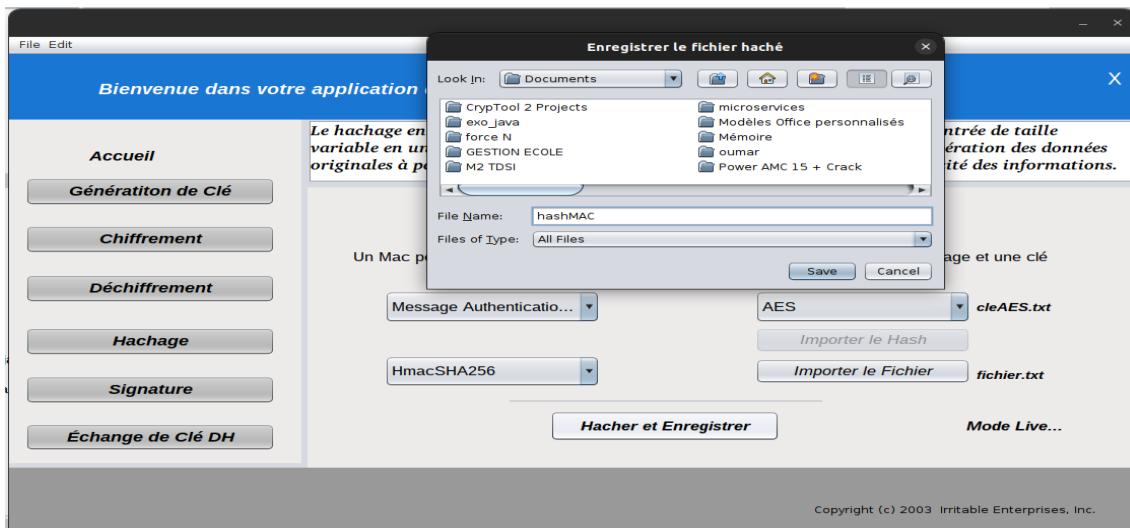
— Ensuite on importe la clé secrète.



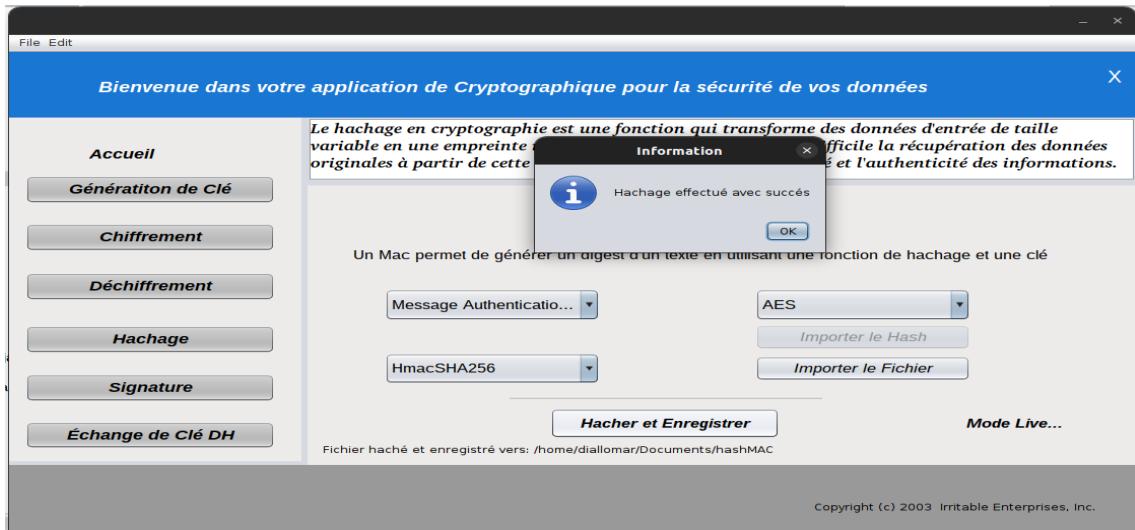
— Puis on importe le fichier à hacher.



— On indique un chemin d'enregistrement.



- Et en fin on hache et enregistre l'empreinte vers un chemin qu'on déterminera.



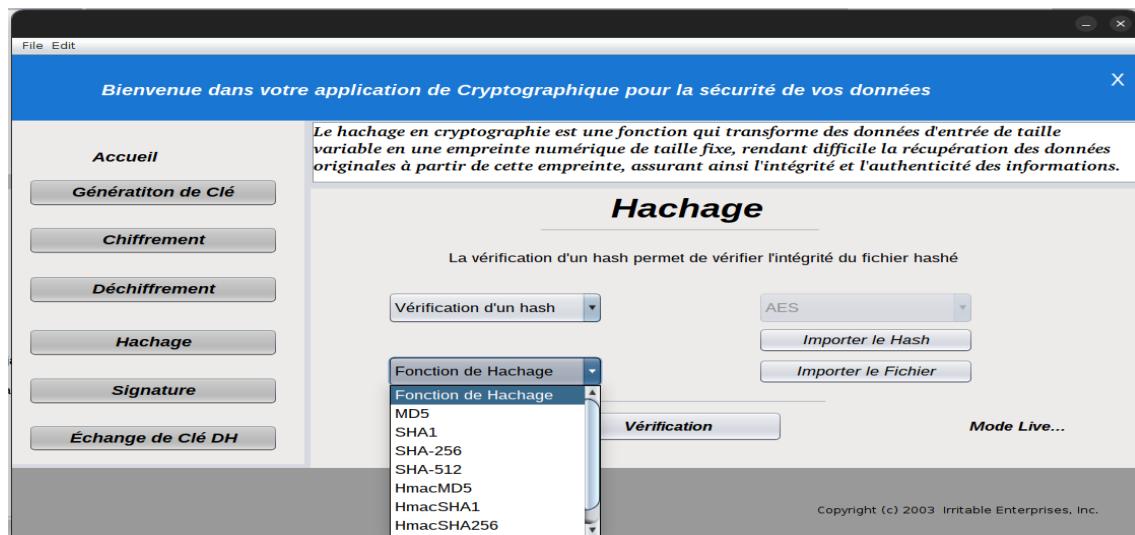
4.3 Vérification d'une empreinte

La vérification se fait de deux façon selon la méthode utilisée lors du hachage, voici ces 2 méthodes :

Message Digest (MD) :

Les fonctions de Hachage utilisées sont : **MD5, SHA1, SHA-256, SHA-512**.

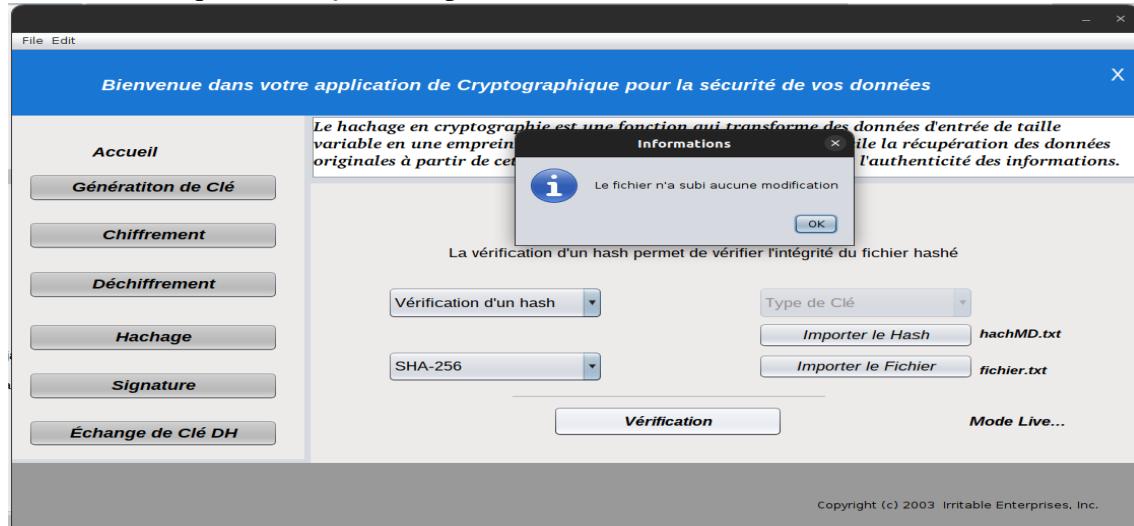
Lorsqu'on clique sur l'une de ces fonctions précédentes l'option *Type de Clé* reste désactivée car ces fonctions sont propre à la méthode Message Digest (qui pour rappel n'utilise pas de clé secrète).



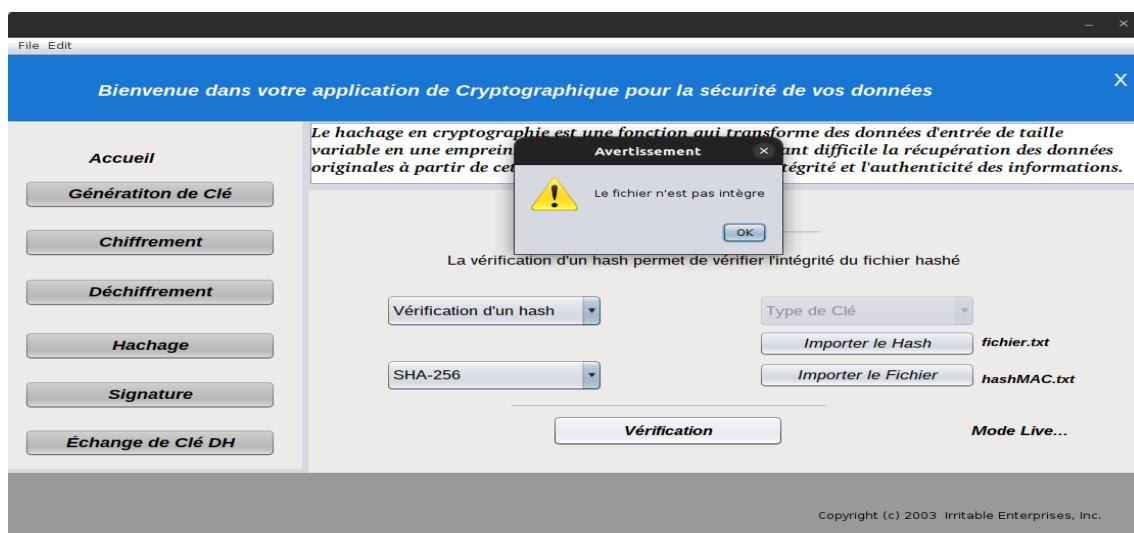
On choisie la fonction de Hachage MD utilisée et on importe le fichier et l'empreinte.



Et enfin on clique sur **vérification** pour avoir le résultat suivant :



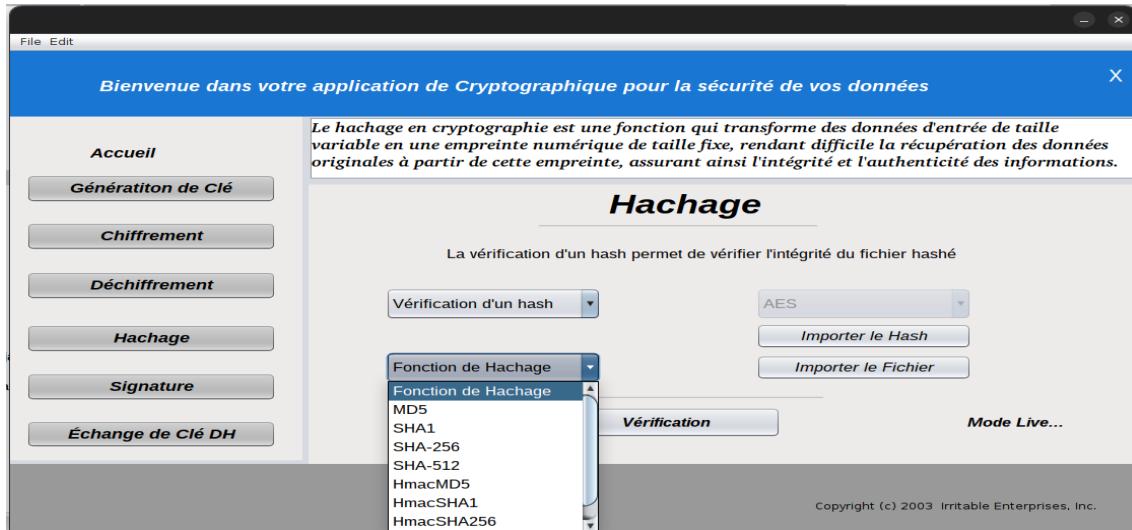
Ou bien ce résultat.



Message Authentication (MAC) :

Pour cette méthode, les fonctions de Hachage utilisées sont : **HmacMD5, HmacSHA1, HmacSHA-256, HmacSHA-512.**

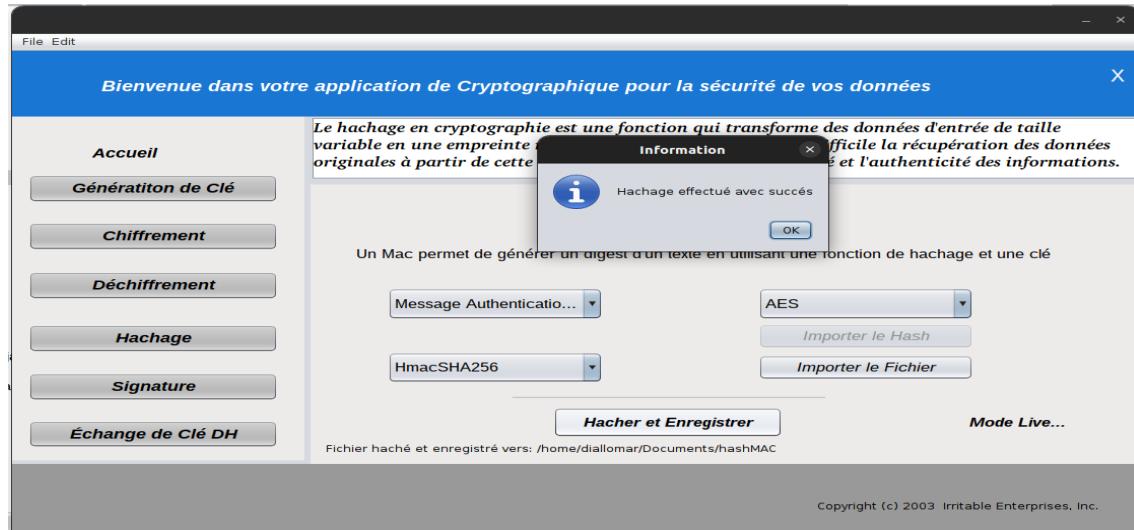
Lorsqu'on clique sur l'une de ces fonctions précédentes l'option **Type de Clé** s'activée car ces fonctions sont propre à la méthode MAC (qui pour rappel utilise une clé secrète pour le hachage).



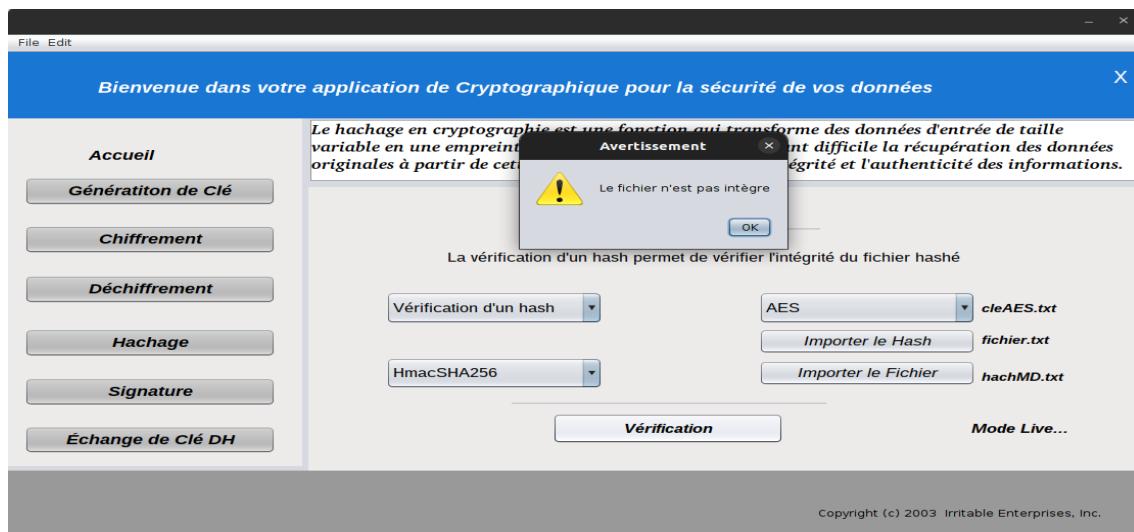
On choisie la fonction de Hachage MAC utilisée et on importe le fichier, l'empreinte ainsi que la clé.



Et enfin on clique sur **vérification** pour avoir le résultat suivant :



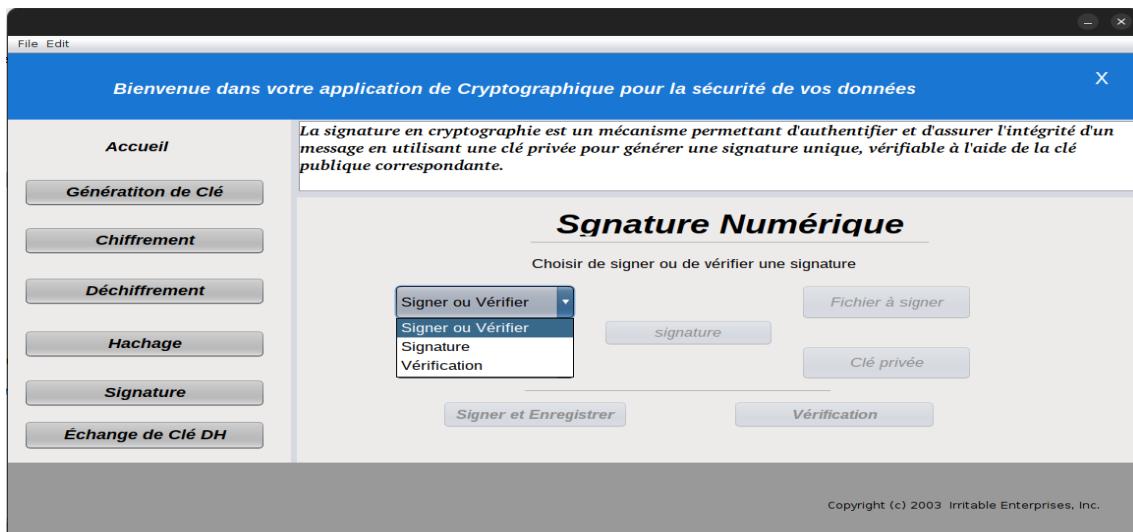
Ou bien ce résultat.



Chapitre 5

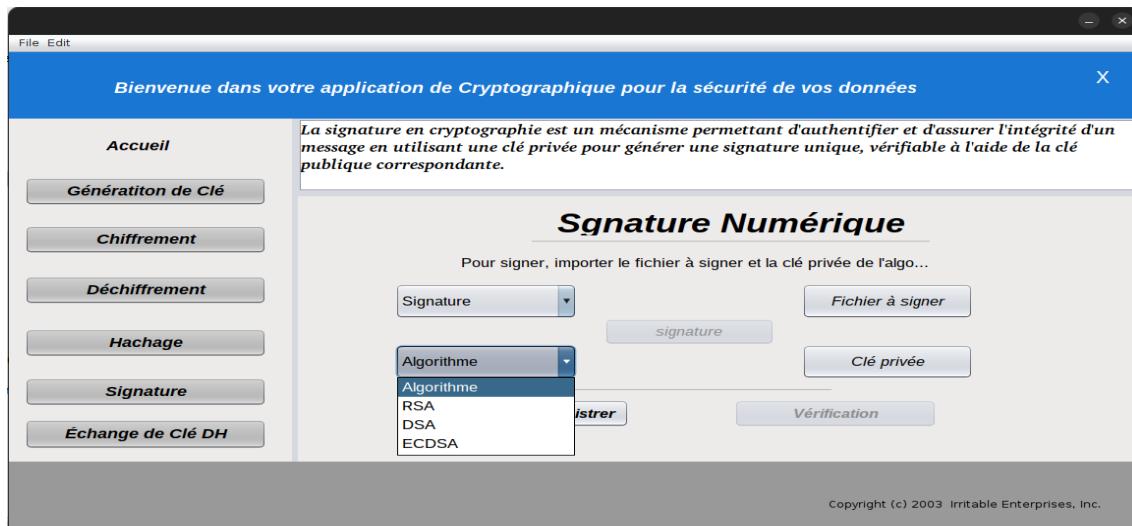
Signature

La signature et sa vérification sont effectuées de la manière suivante selon l'option choisie :

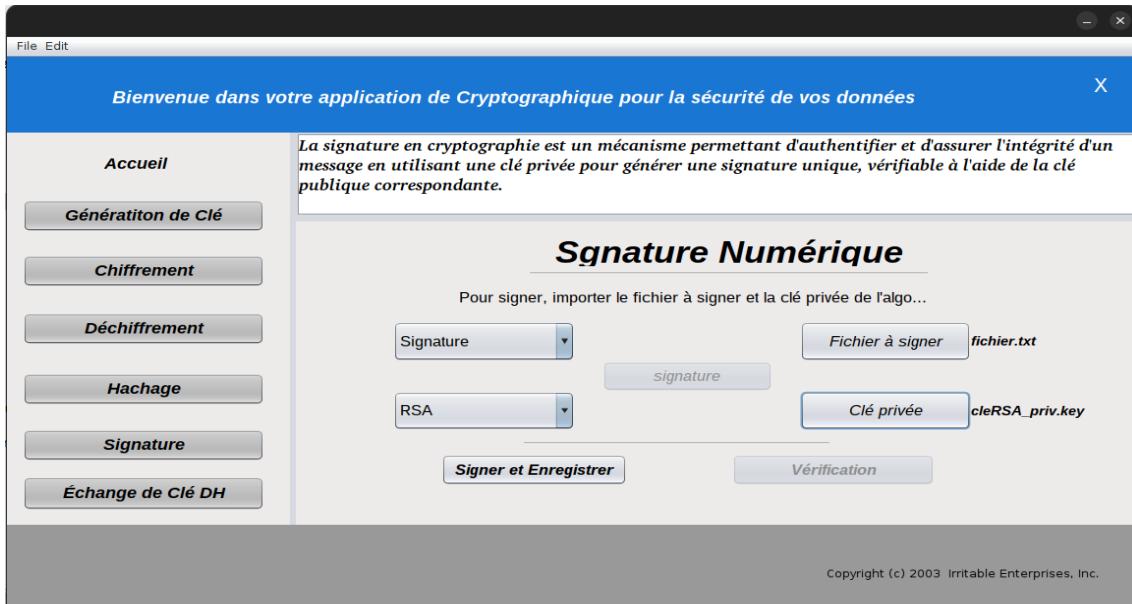


5.1 Signature

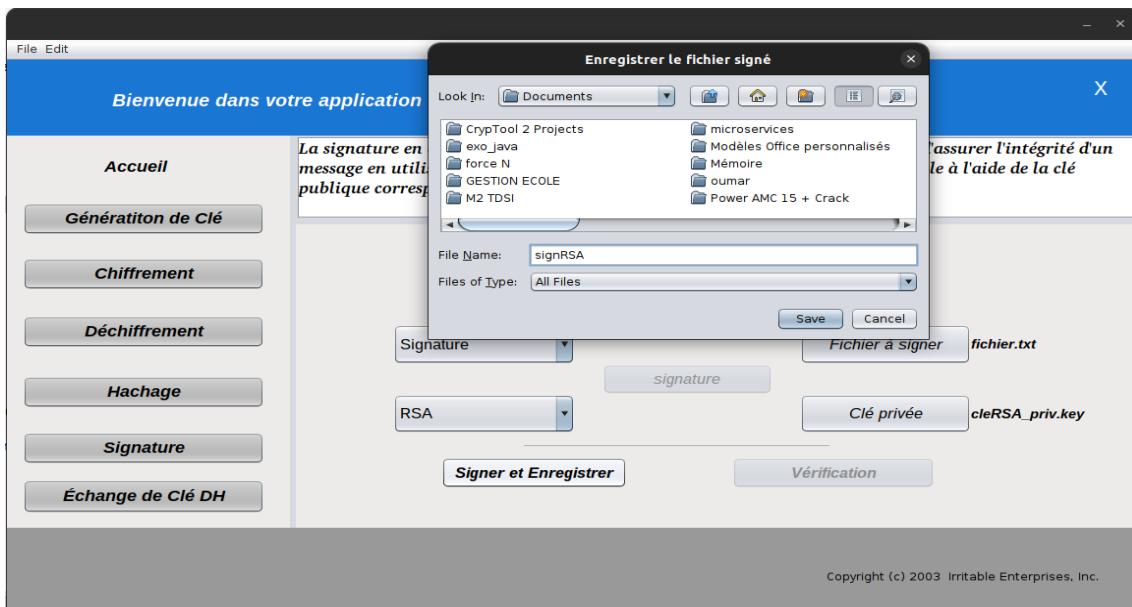
- On Choisie d'abord l'algorithme (Asymétrique).



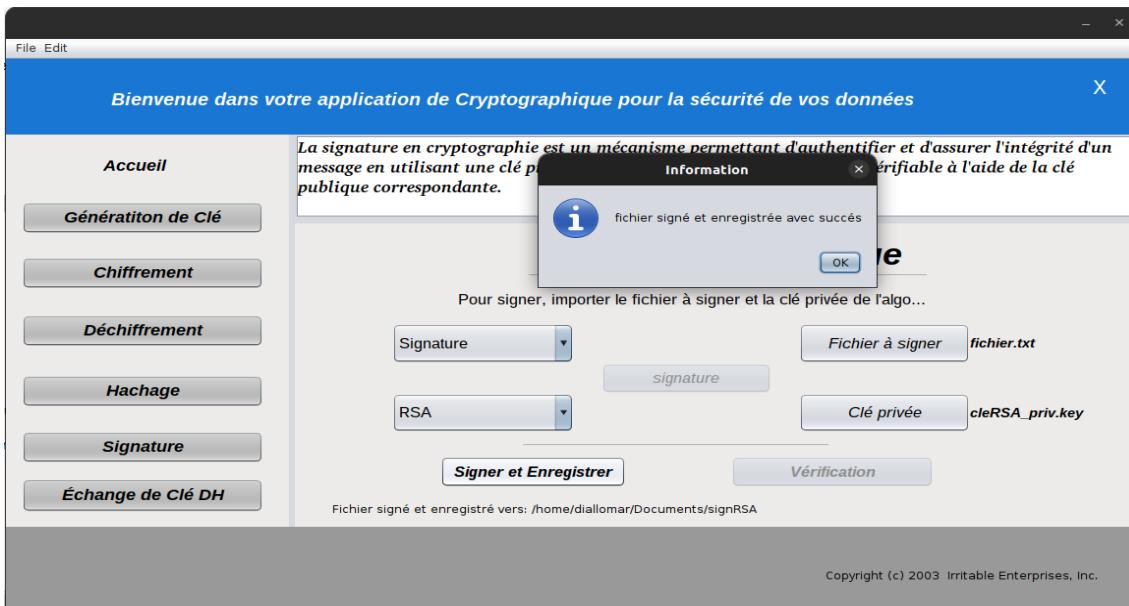
- Ensuite on importe le fichier à signer et la clé privée.



- Puis on choisie le chemin d'enregistrement.

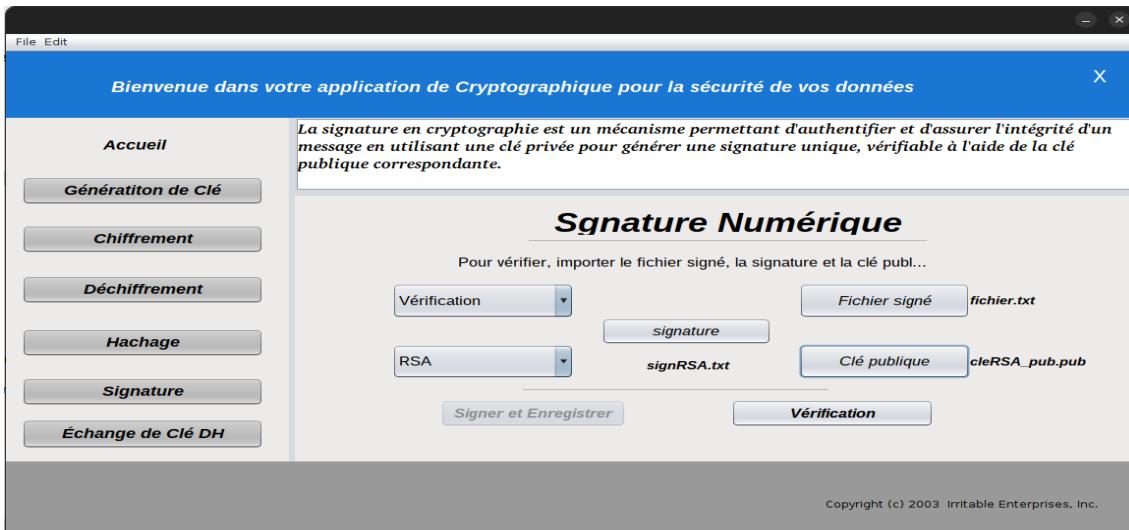


- Et en fin on signe et enregistre la signature vers le chemin indiqué.

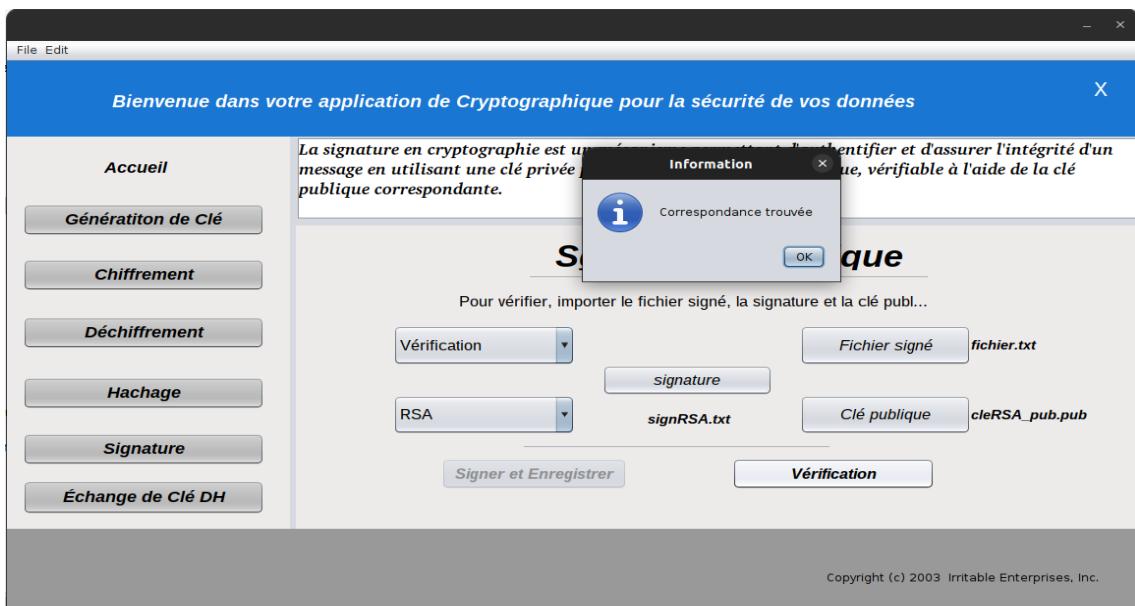


5.2 Vérification

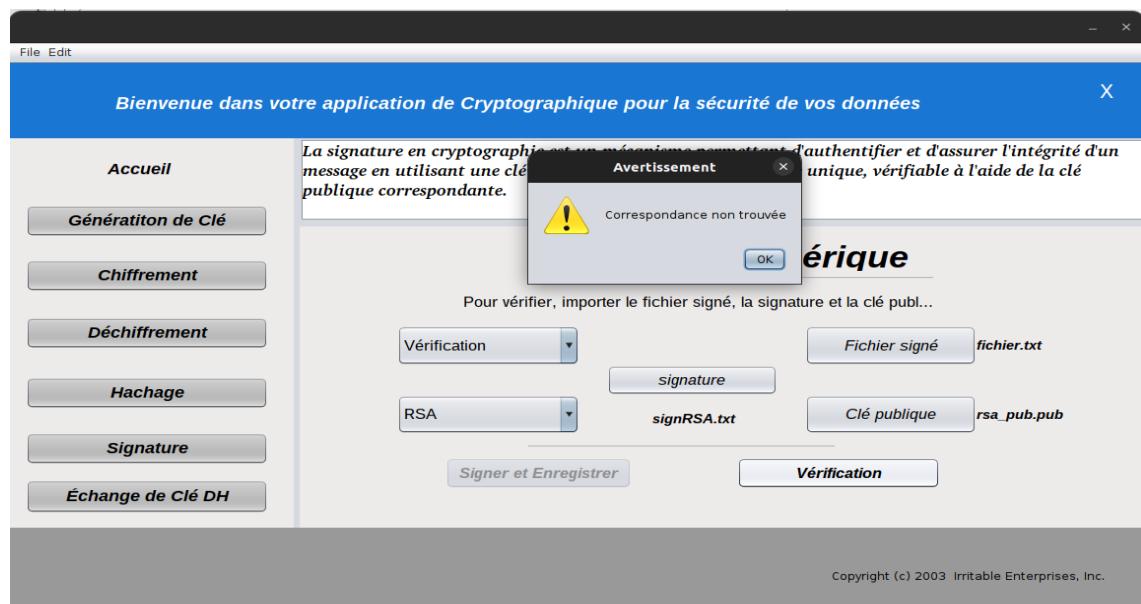
- Pour la vérification, on choisie l'algorithme, la signature, le fichier signé et la clé publique.



— Et en fin on clique sur **vérification** pour obtenir le résultat suivant.



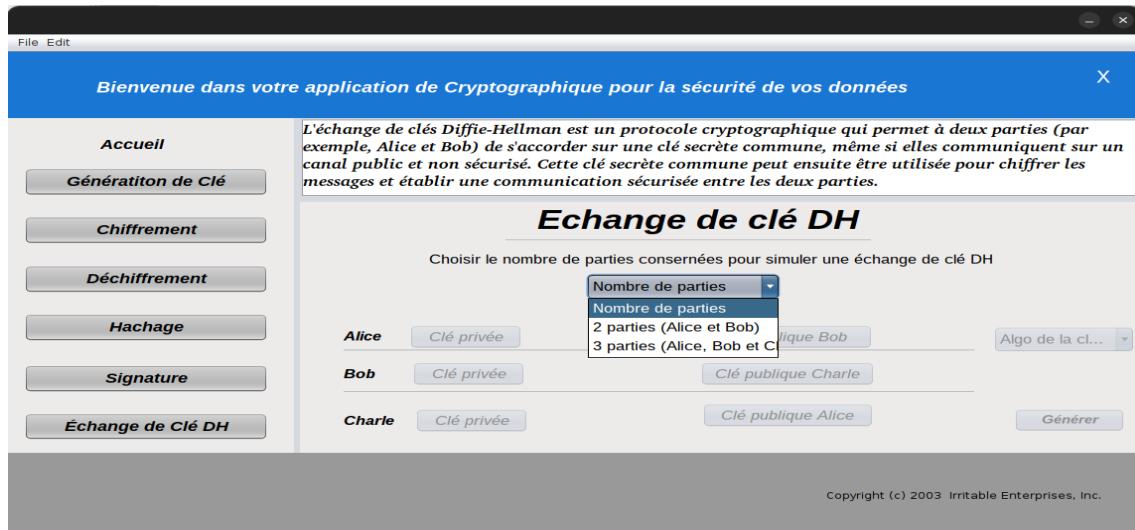
Ou bien le suivant



Chapitre 6

Echange de clés Diffie-Hellman

L'échange de clé avec le protocole Diffie-Hellman à 3 est similaire avec celui à 2 parties. Ces deux échanges se font comme suit selon le nombre de parties concernés :



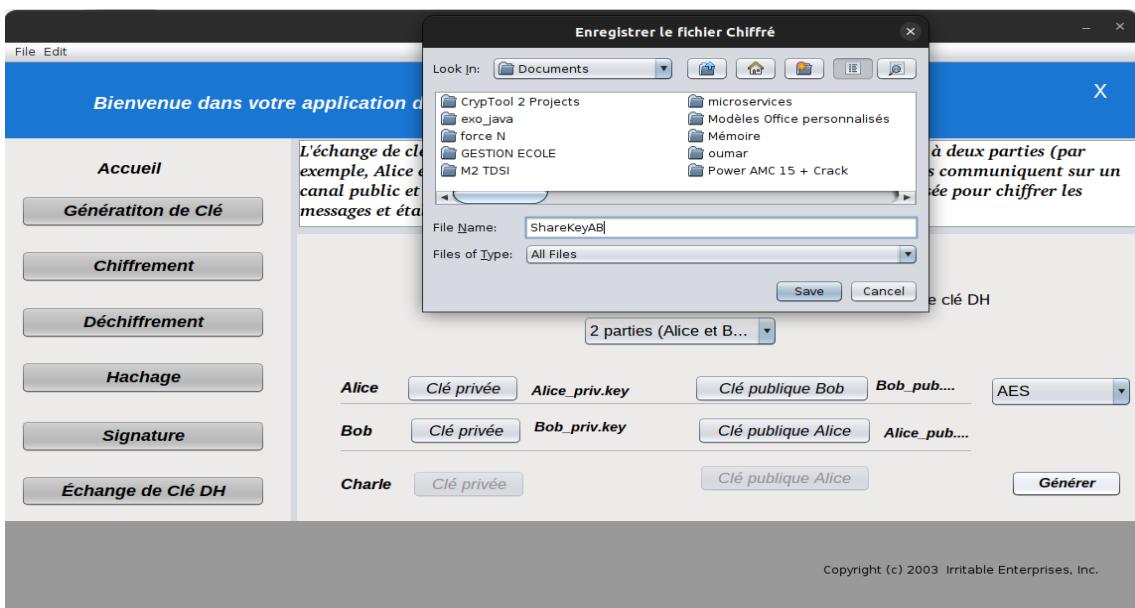
6.1 Échange de Clés avec 2 parties



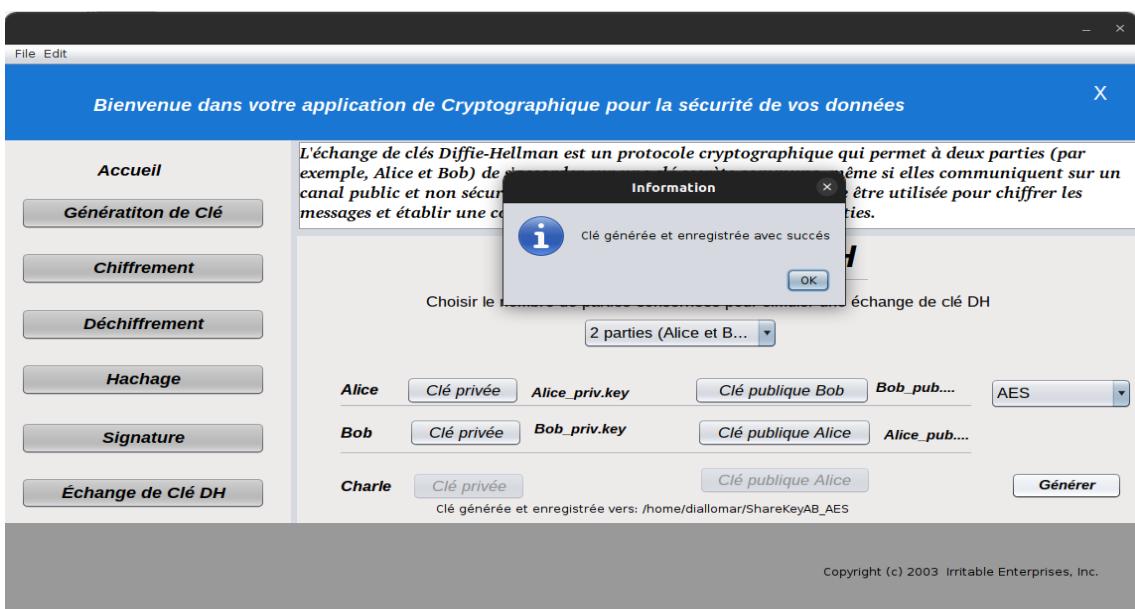
- En suivant les instruction, on importe les clés publiques et privées de Alice et de Bob :
- Ensuite on choisie l'algorithme pour la clé secrète qui sera partagée.



- On indique un chemin d'enregistrement.



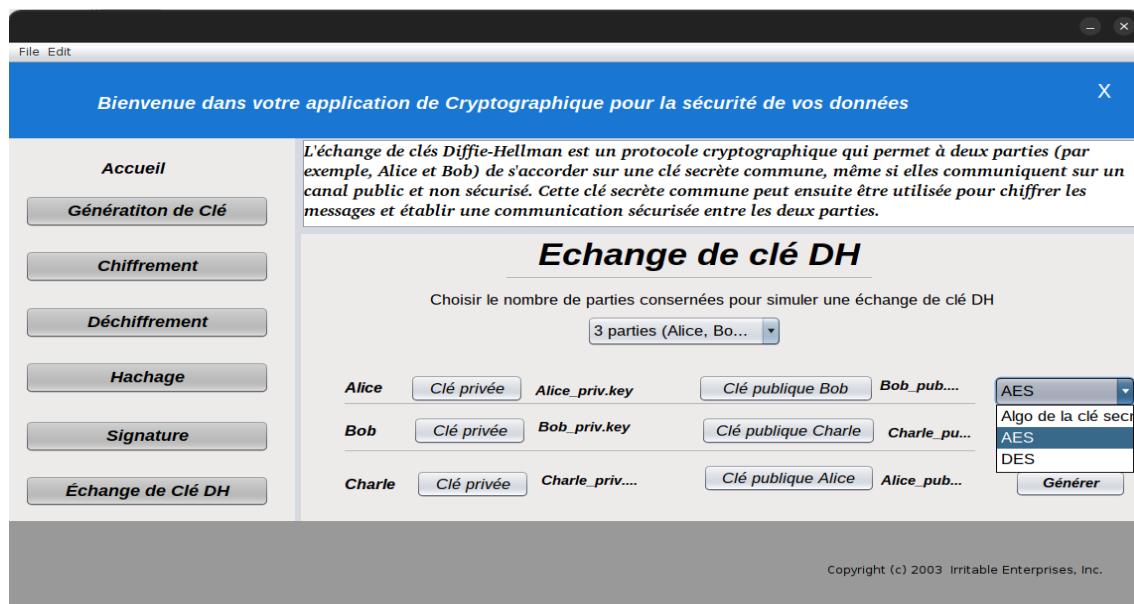
- Et en fin on génère et enregistre la clé secrète (qui est en pratique détenue par Alice et Bob) vers le chemin indiqué.



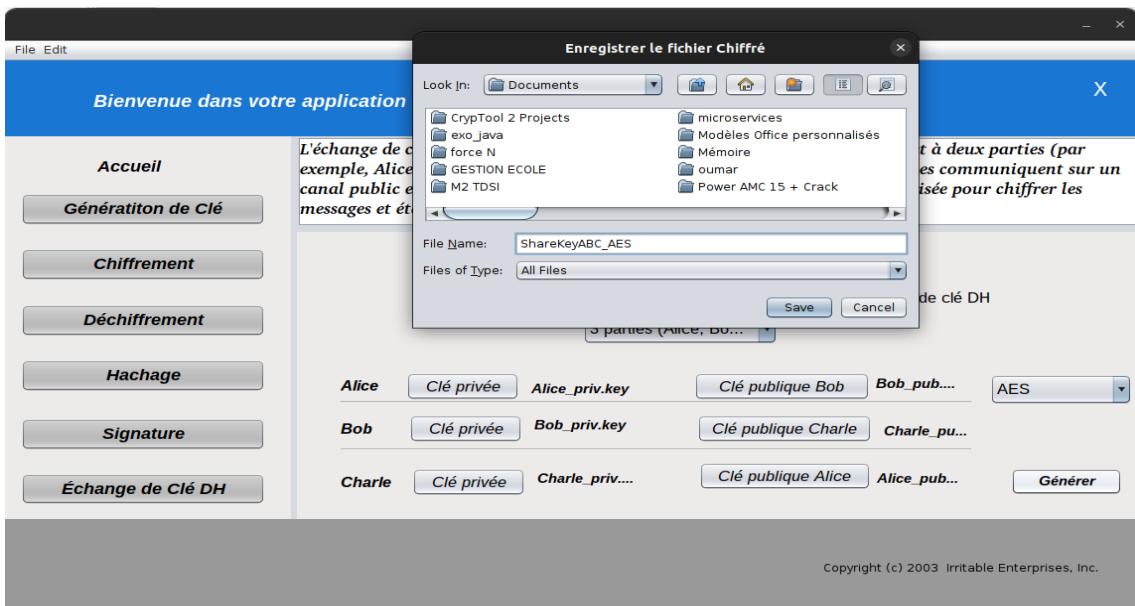
6.2 Échange de Clés avec 3 parties



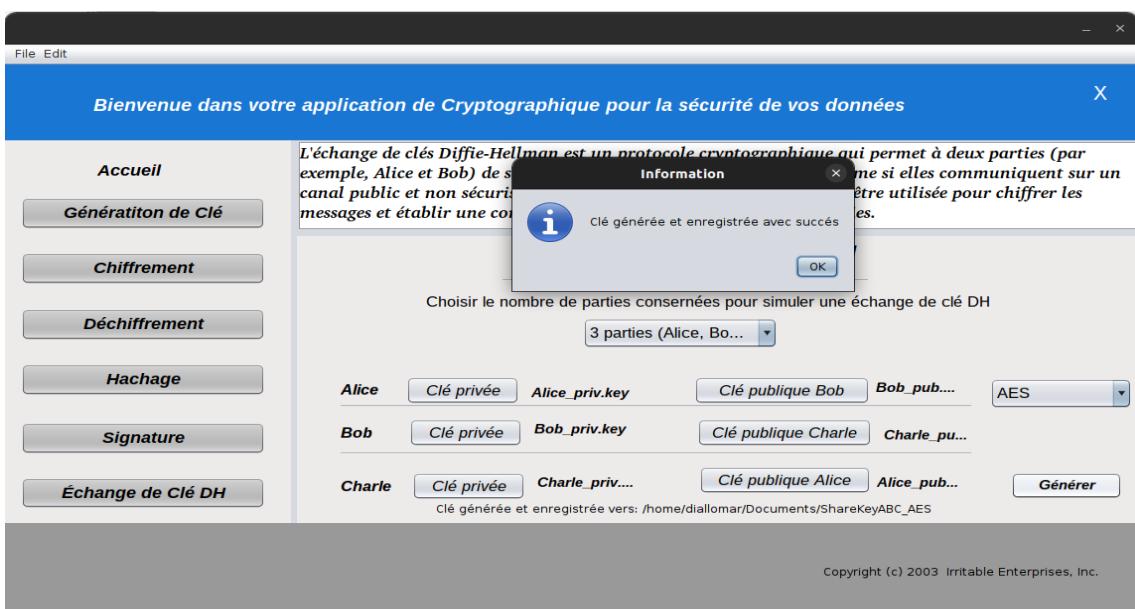
- De la même manière, on importe les clés publiques et privées de Alice, de Bob et de Charle.
- Ensuite on choisie l'algorithme pour la clé secrète qui sera partagée.



- On indique le chemin d'enregistrement



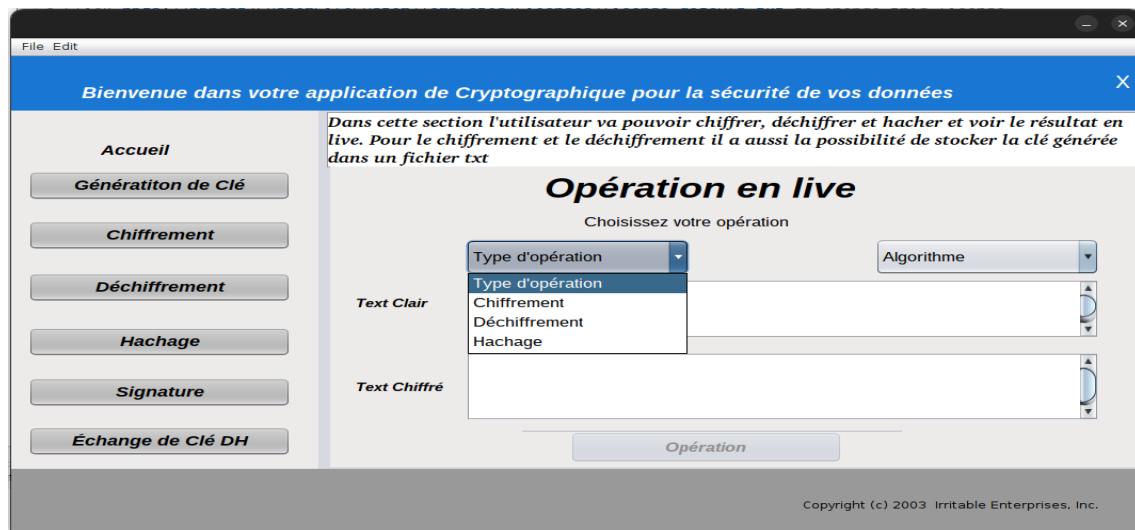
- Et en fin on génère et enregistre la clé secrète (qui est en pratique détenu par Alice, Bob et Charle) vers le chemin indiqué :



Chapitre 7

Mode Live

Dans cette section le mode live permet de chiffrer du texte brut pour avoir le text chiffré à la sortie.

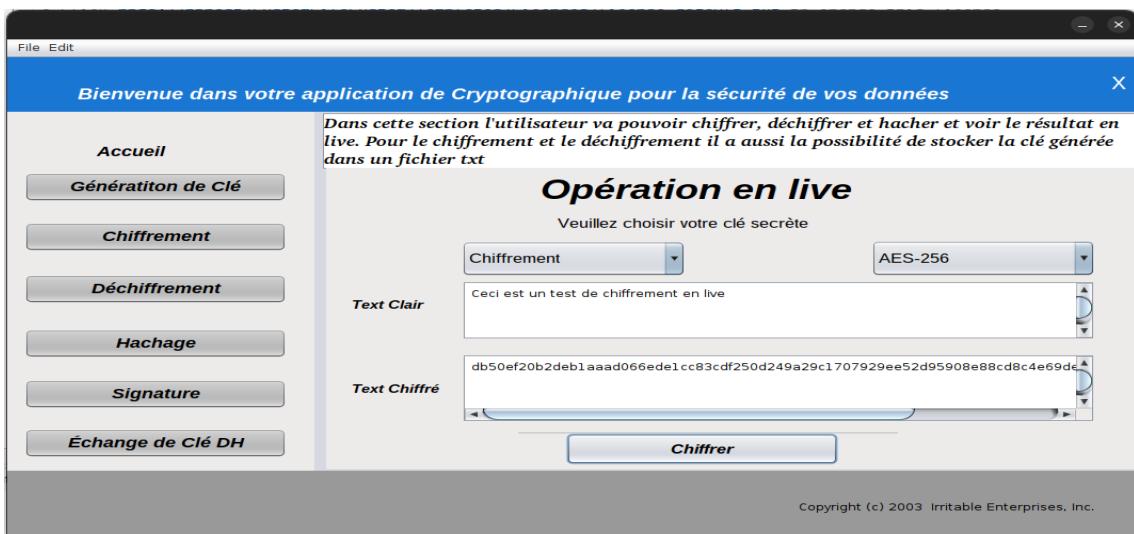


7.1 Chiffrement

- On choisie l'algorithme de chiffrement.

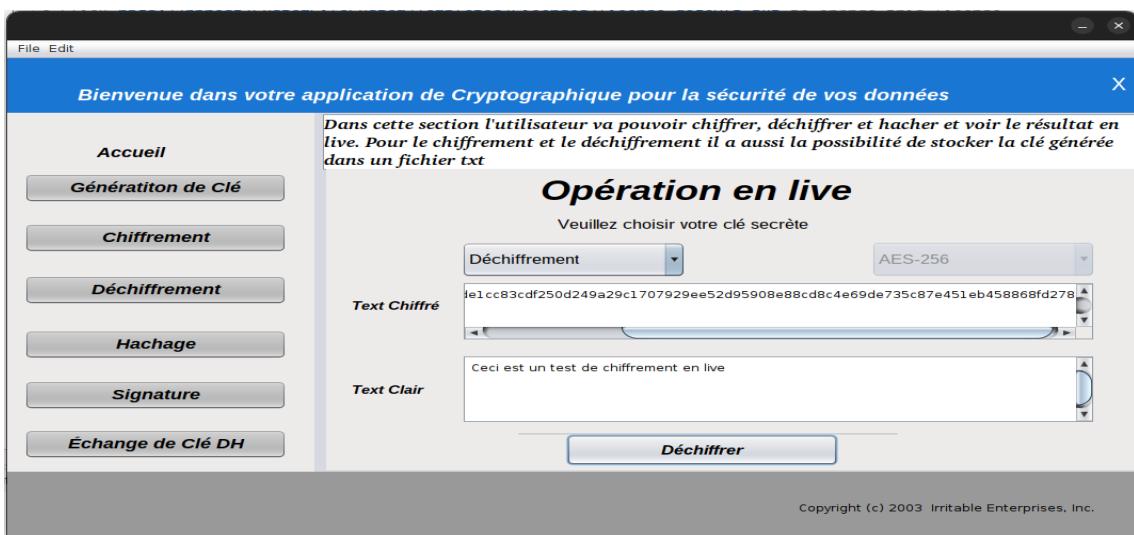


- On saisie le texte et enfin on chiffre pour avoir le résultat suivant :



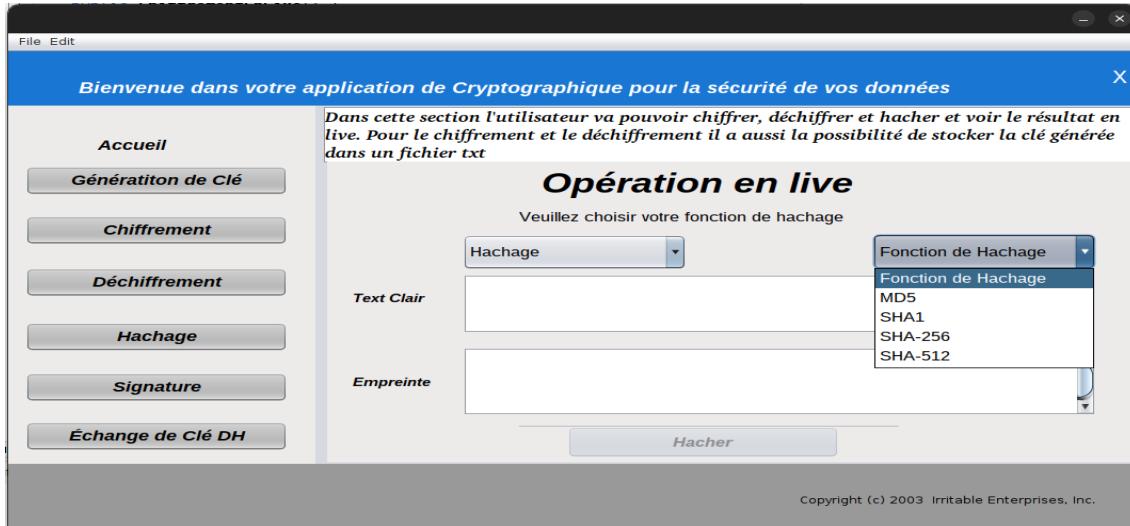
7.2 Déchiffrement

- On choisie l'opération *déchiffrement* puis on clique sur déchiffrer.

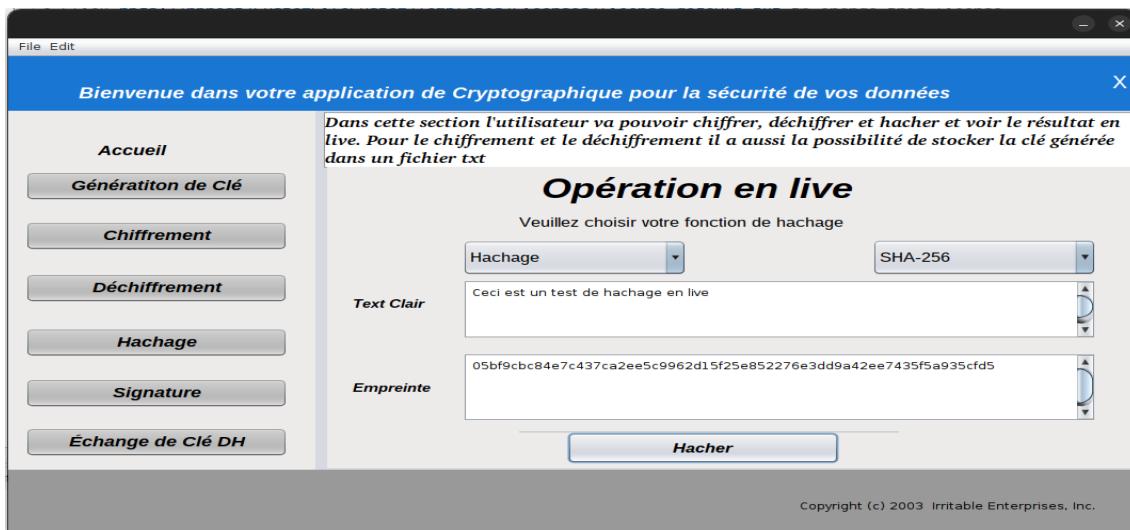


7.3 Hachage

- On choisit la fonction de hachage.



- On saisie le texte et enfin on hache pour avoir le résultat suivant :



Conclusion

Cette application offre un ensemble complet de fonctionnalités de cryptographie, ce qui en fait un outil précieux pour garantir la confidentialité et l'intégrité des informations sensibles. Grâce à son interface utilisateur conviviale développée en utilisant Java Swing, elle est accessible à un large éventail d'utilisateurs, qu'ils soient des professionnels de la sécurité ou des individus soucieux de la protection de leurs données personnelles. En somme, cette application illustre comment la combinaison de la puissance de Crypto Java avec la convivialité de Java Swing peut répondre aux besoins de sécurité croissants de notre ère numérique. Il est indéniable que la cryptographie continue de jouer un rôle crucial dans la protection des données, et cette application en est un exemple concret.