UNIVERSITÉ
CAEN
NORMANDIE

Université de Caen Normandie
UFR des Sciences

Master 1 : Informatique
Année 2023/2023

Compte Rendu Projet

Crypted or Uncrypted

Fouché Stanislas (22007315)
Priou Antoine (22008452)

Table des matières

1 Introduction

Ce projet a été réalisé dans le cadre d'une éventuelle enquête criminelle, où l'on récupère un ordinateur ou un disque. Nous devons déterminer si l'appareil contient des fichiers chiffrés/partition chiffrés ou non. Après avoir effectué une analyse préalable des logiciels et algorithmes de recherche de fichiers, de dossiers ou de partitions chiffrés, nous sommes sur le point de déployer un protocole visant à identifier les solutions les plus efficaces dans ce domaine. Cette démarche fait suite à notre précédent rapport détaillant le fonctionnement de ces outils. L'objectif de ce compte-rendu est de démontrer l'efficacité des différentes méthodes en termes de temps et de performances. Cette expérience est importante dans une enquête criminelle afin que les recherches de chiffrement soient les plus efficaces et rapides possibles. En évaluant rigoureusement ces solutions, nous visons à fournir des recommandations précises quant au choix des logiciels et des approches les mieux adaptés.

2 Protocole de réalisation du Dataset

Dans le cadre de recherches et d'expérimentations sur les algorithmes et logiciels de recherche de fichiers chiffrés, nous proposons de tester la robustesse et l'efficacité des outils mentionnés dans notre rapport précédent.

Pour mener à bien ces tests, nous avons besoin d'un ensemble de données comportant une centaine de fichiers que nous chiffrerons à l'aide de plusieurs algorithmes de chiffrement connus.

2.1 Sélection des outils

Nous allons donc citer les outils et logiciels utilisés pour les différentes étapes de l'expérience.

Outils/Méthodes de détection

- cipher.exe (Windows) ;
- TcHunt ;
- Elcomsoft Encrypted Disk Hunter
- MAGNET Encrypted Disk Detector
- algorithme de recherche par extension de fichiers
- SVM basé sur l'entropie

À noter que tous ces outils sont gratuits d'utilisation.

2.2 Dataset

Nous avons choisi de créer nous-même notre propre dataset afin de pouvoir analyser différents types de fichiers les plus courants (pdf,csv,png,jpg et txt) et aussi de différentes tailles.

Les fichiers du dataset proviennent de plusieurs dataset en ligne accès sur la plateforme Kaggle. Toutes les informations et sources sont placées dans le fichier data.txt dans le dossier dataset du GitHub.

Le dataset comporte actuellement 195 fichiers, les différentes

- 27 csv
- 40 jpg
- 39 pdf
- 40 png
- 49 txt

Pour obtenir le dataset chiffré, tout d'abord (en ayant récupéré le projet sur le GitLab Unicaen) il faut se placer dans le dossier **/dataset** et lancer le script **init_chiffrement.py** sous linux (ou utiliser un wsl comme nous le faisons) servant à créer automatiquement la parti chiffré du dataset avec openssl et, avec différents algorithmes. Voici la listes des algorithmes. Attention, l'ensemble des datasets représente plus de 28Go, veillez à réserver l'espace requis.

Algorithme de chiffrement

- aes-256-cbc
- aes-128-cbc
- aes-256-ecb
- des-ede3-cbc
- des-ede3-ecb
- bf-cbc
- camellia-256-cbc
- cast5-cbc
- rc4
- gpg

Ainsi que trois autres datasets :

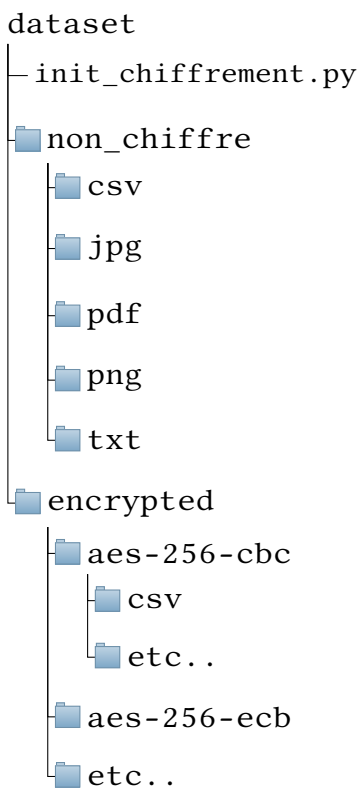
- randomized
- headless
- halfheaded

randomized : fichiers générés aléatoirement, ayant une entropie bien supérieure. Dans le but de tromper les méthodes basées principalement sur l'entropie.

halfheaded : datasets ou chaque fichier est chiffré par aes-256-cbc sans le .enc de chaque fichiers. Dans le but de retirer un éventuel biais de reconnaître par l'extension, ou bien si l'utilisateur de l'ordinateur tente de cacher ses fichiers chiffrés.

headless : similaire à halfheaded, mais aucune extension de fichiers,
example "png/0" et non png/0.png.enc"

L'arborescence du dataset est la suivante :



Cette arborescence permet une étude plus approximative des différents types de fichiers une fois chiffrés. Elle va aussi nous permettre l'apprentissage de notre SVM.

2.3 Choix OS

L'utilisation des applications/méthode se font principalement sur Linux(Ubuntu) qui est l'OS le plus compatible et simple d'utilisation avec nos outils.

Cependant, certains logiciels seront seulement utilisables sur Window.

Dans nos tests personnels, nous avons utilisé un WSL.

2.4 Analyse

Dans un premier temps, nous allons utiliser les logiciels permettent d'identifier si un disque est chiffré ou si un système comprend des partitions/volumes chiffré.

Pour cela, nous allons tester avec un Volume monté Veracrypt avec une partition chiffré sur un disque virtuel et un volume Veracrypt.

Dans un deuxième temps, nous voulons comparer les logiciels et les méthodes permettent d'identifier des fichiers chiffrés, en utilisant le dataset créer auparavant.

Nous serons ainsi en mesure de discerner les subtilités et les performances de chaque approche, mettant en avant leur précision et leur efficacité respectives.

3 Expérimentation

3.1 Détection de Partition/Volume

Les logiciels pouvant détecter des partitions/volumes sur un disque sont :

- Elcomsoft Encrypted Disk Hunter [4]
- MAGNET Encrypted Disk Detector [10]

On rappelle qu'on utilise ici Veracrypt avec une partition chiffré sur un disque virtuel et un volume Veracrypt.

3.1.1 Elcomsoft Encrypted Disk Hunter

Ce logiciel d'Elcomsoft est gratuit, en libre accès et uniquement utilisable sur Window.

Il parcourt tout le système à la recherche de trace de chiffrement.

Pour l'utiliser il suffit de lancer le fichier **eddh.exe** (fichier présent dans le GitHub).

Après expérience, on remarque que l'application détecte bien la présence de la partition chiffrés (spécifie le type de chiffrement). Elle a aussi détecté des signes de présence de volumes chiffrés sur le disque.

Remarque : l'application créée une copie des résultats dans un fichier txt.

3.1.2 MAGNET Encrypted Disk Detector

L'application de MAGNET est très similaire au logiciel d'Elcomsoft, utilisable uniquement sur Window, gratuit et en libre service.

Il parcourt aussi le système à la recherche de traces de chiffrement.

Pour l'utiliser, il faut lancer le fichier **EDDv310.exe** (aussi présent dans le gitHub)

À nouveau, on a bien réussi à détecter la partition chiffrée et les volumes chiffrés.
L'application fait une recherche approfondie de trace Bitlocker.

On a cependant ici les détails sur chaque lecteur. Contrairement au logiciel d'Elcomsoft on nous indique ici les lecteurs comportant des signes de chiffrement VeraCrypt. Ce qui le rend un peu plus pertinent.

3.2 Détection de fichiers chiffrés

3.2.1 Cipher.exe

Cipher est un EFS (Encrypting File System) intégré à Window (version Pro uniquement). Dans le cas d'une enquête, certaines personnes pourraient chiffrer leur fichier directement avec l'outil de chiffrement de base de Window.

Dans le cadre de détection de fichiers chiffrés, on peut utiliser la commande :

cipher /u /n

Cependant il ne permet que de détecter les fichiers chiffrées par Cipher lui-même et donc n'est pas très efficace.

3.2.2 TCHunt

TCHunt est une application qui détecte les fichiers chiffrés, en se basant essentiellement sur la taille, l'entropie et la signature des fichiers (analyse en-tête).

Pour l'utiliser, on se place à la racine du répertoire et on effectue la commande :

find ./ -type f | tchuntng - -s

Voici un extrait de résultat :

- ./TCHunt-ng/test/samples/karabina encfs/sk1pJ [data]
- ./TCHunt-ng/test/samples/message.txt.asc [data, pgp/gpg]
- ./TCHunt-ng/test/samples/gpg pubkey.asc [keys, pgp/gpg]
- ./dataset/encrypted/randomized/0.txt [data]

Lorsque TCHunt détecte un fichier chiffré, il nous présente la nature du fichier (exemple : keys,data) ou encore certains type de chiffrement.

On remarque que TCHunt ne permet pas l'identification de tout les types de chiffrement, par exemple il ne détecte pas les fichiers de type openssl en .enc.

Le fichier 0.txt qui a été détecter est un fichier non chiffré, mais ayant été généré aléatoirement, il a donc été considérer comme chiffré à cause de sa grande entropie alors que le fichier n'est pas chiffré, TCHunt peut donc admettre des faux positifs, pouvant être problématique, surtout dans le cadre d'une enquête.

3.2.3 Recherche par extension

Nous avons créer un fichier détectant les fichiers chiffrés par leur extensions.

Il faut lancer le fichier **recherche_extension.py** et spécifier le dossier de recherche voulu.

Cependant, cette méthode ne permet pas de détecter tout les types de chiffrement, les extensions non connues ne seront pas détectées et un fichier chiffré caché (par un conteneur) non plus.
Si l'on change l'extension d'un fichier chiffré, on ne pourra pas le détecter.

3.2.4 Support Vector Machine

Pour mettre en pratique les domaines d'apprentissage majeurs du second semestre il a été décidé de réaliser nous même une méthode de détection par apprentissage, en particulier d'un Support Vector Machine (SVM), ce dernier se basant sur l'entropie d'un fichier pour déterminer s'il est chiffré, pour cela la méthode est de récupérer l'ensemble des fichiers du dataset non chiffré pour le fusionner avec un dataset chiffré passé en argument pour obtenir un dataset de 195 fichiers dont 70% du dataset chiffré et 30%, l'explication étant que pour détecter si un fichier est chiffré ou non il faut aussi que l'apprentissage se fasse sur les deux types de fichiers, le but étant la détection des chiffrés, il représente donc une bonne majorité du set.

On utilise le module svm de sklearn avec train_test_split pour split le dataset fusionné, 70% sert à entraîner le modèle et 30% servent à établir la précision en comparant avec les valeurs prédites : 0 si le svm estime que le fichier n'est pas chiffré, 1 si il estime qu'il l'est.

En expérimentant avec les différents chiffrements on a obtenu des chiffres de précision à hauteur d'environ +80% d'accuracy, mais divers test ont révélés que le SVM n'obtenait pas une précision uniforme pour tout les types de fichiers, les résultats sont donc exprimés, pour chaque type de fichiers présents sur chaque dataset chiffré

4 Résultats Expérimentaux

On peut comparer les temps d'exécution des différents algorithmes :

- Elcomsoft Encrypted Disk Hunter : 30s
- MAGNET Encrypted Disk Detector : 30s
- cipher : instantané
- TCHunt : -1mn
- recherche par extension : -1mn
- SVM : environ 5mn

Il n'y a pas d'énorme différence d'exécution entre les algorithmes excepté le SVM qui lui est plus précis.

Détection de disque/partition : MAGNET Encrypted Disk Detector / Elcomsoft Encrypted Disk Hunter

- même temps d'exécution
- gratuits et libre d'accès
- utilisation simple (.exe)
- Les deux applications sont assez efficaces et pertinentes
- l'application MAGNET Encrypted Disk Detector précise lui les lecteurs attachés aux conteneurs/partitions chiffré.

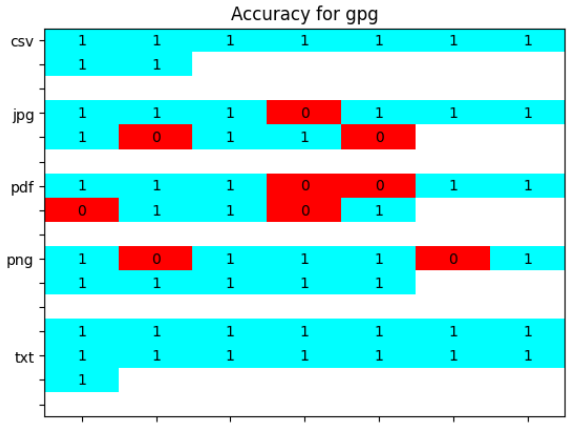
L'Utilisation de MAGNET Encrypted Disk Detector serait donc un peu plus pertinent que Elcomsoft Encrypted Disk Hunter. Cependant, les deux applications sont tous les deux très pertinentes.

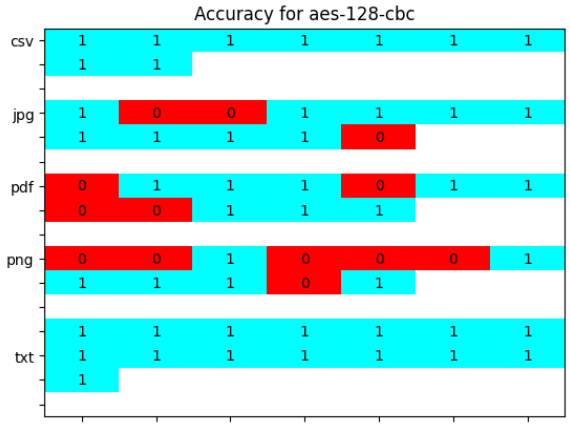
Analyse de fichiers chiffrés : Cipher / analyse par extension / TCHunt

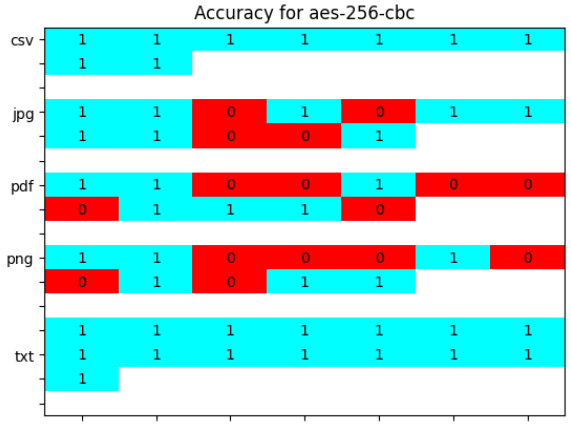
- temps d'exécution : presque similaire
- cipher pas très efficace, pertinent seulement dans le cas particulier où l'utilisateur a utilisé cipher pour chiffrer ses fichiers
- TCHunt se base sur plus de facteurs que juste une simple recherche par extension de fichiers donc + pertinent

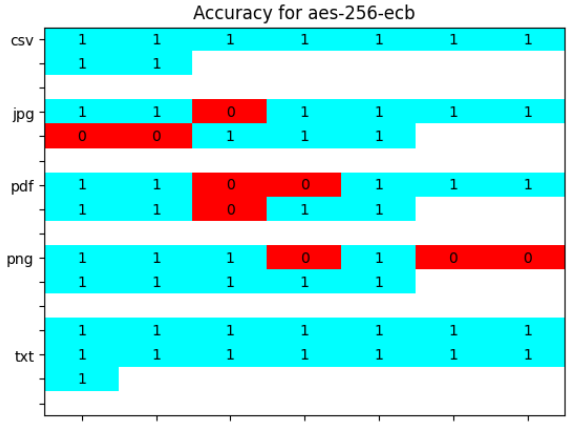
TCHunt est plus pertinent que les autres méthodes.

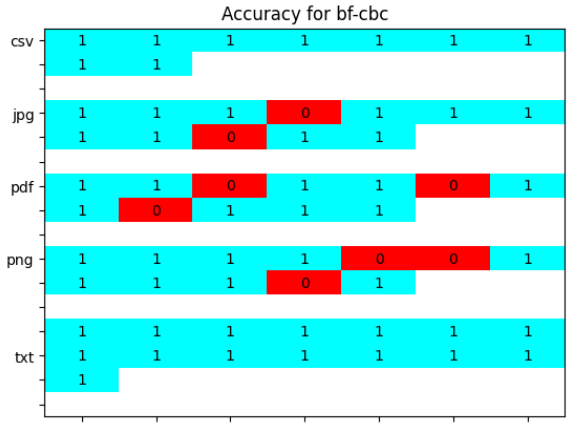
On obtient donc les résultats des 13 datasets avec pour chaque : Pour chaque Dataset son type de chiffrement/caractéristique en légende Pour chaque ligne le type de fichiers Le résultat des prédictions du SVM : 1 pour une bonne prédiction en bleu sinon 0 en rouge

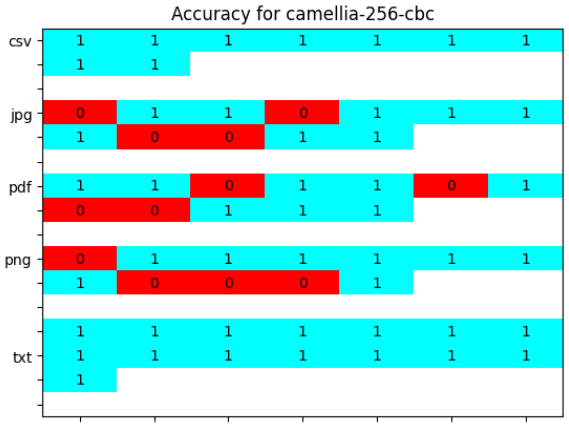


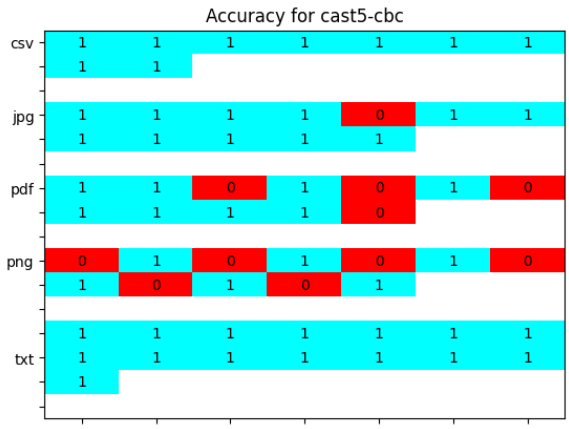


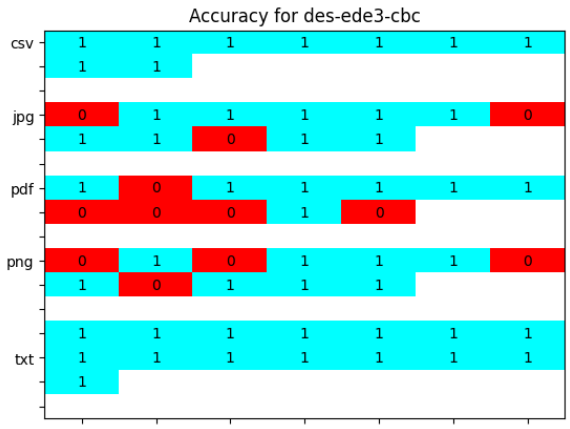


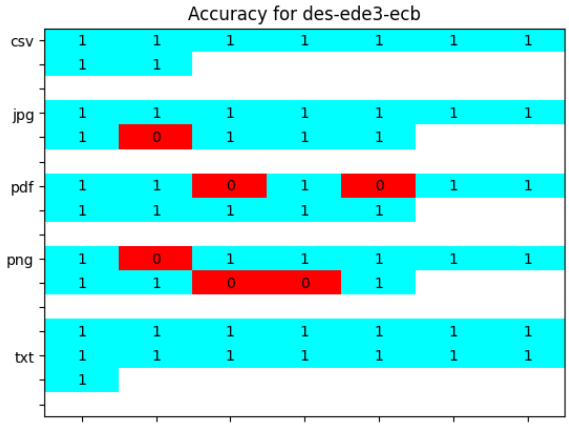


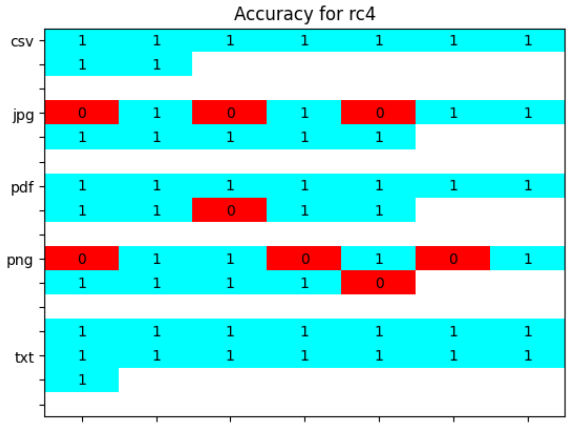


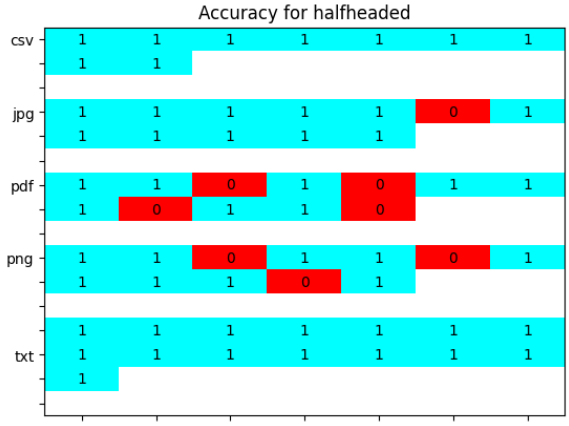


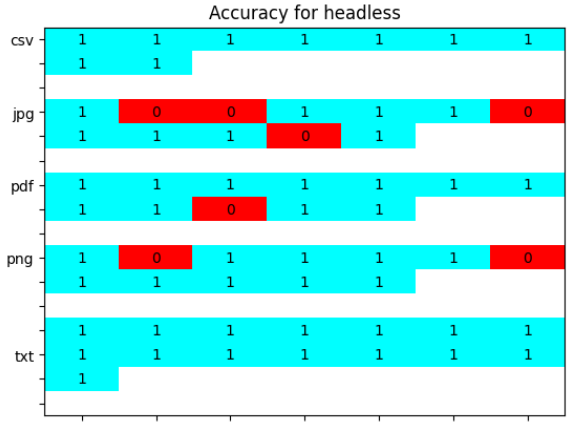

Accuracy for cast5-cbc
csv
jpg
pdf
png
txt

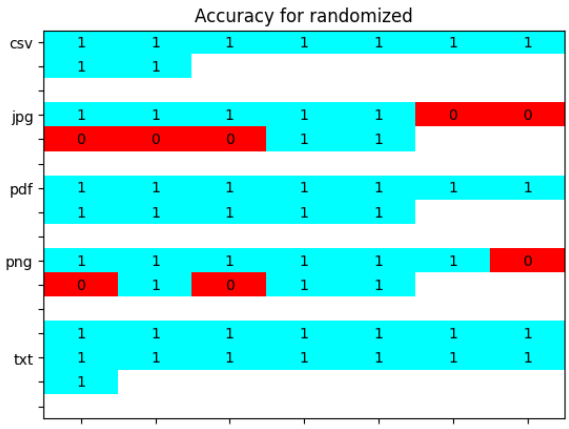

Accuracy for des-ede3-cbc
csv
jpg
pdf
png
txt


Accuracy for des-ede3-ecb
csv
jpg
pdf
png
txt


Accuracy for rc4
csv
jpg
pdf
png
txt


Accuracy for halfheaded
csv
jpg
pdf
png
txt


Accuracy for headless
csv
jpg
pdf
png
txt



De par ces résultat on observe tout d'abord, qu'effectivement le traitement n'est pas le même selon le type de fichiers, les fichiers csv et les fichiers textes présentent 100% de précisions sur les prédictions du SVM, et ceux pour tout les datasets chiffrés, ainsi que ceux sans extensions, extensions de chiffrement supprimées et ceux générés aléatoirement.

On note aussi que le SVM fait aussi des erreurs sur randomized en dépit de son statut de non chiffré, même si il obtient des meilleures performances, une haute entropie peut parfois tromper le modèle. :

Les résultats sont tout de même satisfaisant, cependant une piste d'amélioration pourrait être un apprentissage sur encore plus de données et la prise en compte de facteurs supplémentaires à l'entropie.

5 Répartion du projet

Stanislas

- Constitution du dataset chiffré
- Script de chiffrement pour obtenir l'ensemble des datasets chiffrés
- Rédaction du rapport
- Expérimentation des applications

Antoine :

- Script pour ordonner le dataset non chiffré
- Réalisation d'un SVM basé sur l'entropie
- Expérimentation et affichage des résultats obtenus
- Rédaction du rapport

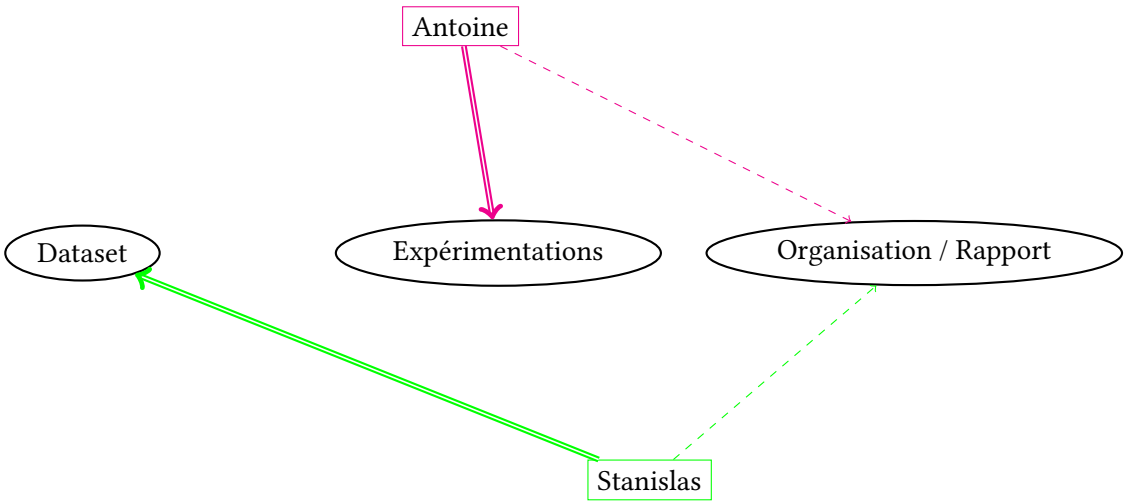


FIGURE 4 – Répartition du travail

Bien que ne représentant seulement la moitié de l'effectif initial nous estimons avoir tout de même donné des éléments de réponse à la question "Crypted Or Uncrypted" ou "comment détecter la présence de fichiers chiffrés"

6 Conclusion

Ce projet nous a permis de bien comprendre les méthodes et outils existants pour détecter des fichiers ou des partitions chiffrés. À travers nos expérimentations, nous avons pu évaluer l'efficacité de divers logiciels et algorithmes, et déterminer lesquels sont les plus adaptés à une utilisation dans le cadre d'une enquête criminelle.

En effet, dans une enquête criminelle, la rapidité et la précision sont des éléments cruciaux. La capacité à détecter des fichiers chiffrés peut fournir des indices essentiels et potentiellement révéler des informations dissimulées par des individus malintentionnés. Par exemple, des criminels pourraient utiliser le chiffrement pour cacher des preuves incriminantes telles que des plans, des communications ou des transactions illégales. Notre étude des différents outils et méthodes a mis en lumière des solutions efficaces comme MAGNET Encrypted Disk Detector et Elcomsoft Encrypted Disk Hunter pour la détection de partitions chiffrées, et TCHunt et notre modèle SVM pour la détection de fichiers chiffrés avec en plus cipher si l'utilisateur s'est servi des services de base de windows pour chiffrer.

Nous avons également noté que chaque méthode présente ses propres avantages et inconvénients. Par exemple, les outils basés sur la recherche d'extensions peuvent être contournés en modifiant les extensions de fichiers, tandis que les méthodes basées sur l'entropie, bien que plus robustes, peuvent être trompées par des fichiers non chiffrés mais ayant une haute entropie.

L'implémentation de notre propre modèle SVM a montré des résultats prometteurs avec une précision élevée pour certains types de fichiers. Cela ouvre la voie à des approches personnalisées et évolutives basées sur l'apprentissage automatique, qui pourraient s'adapter aux nouvelles techniques de chiffrement et aux évolutions futures dans le domaine.

Sachant que le monde de la sécurité et du chiffrement évolue constamment, des nouvelles méthodes de chiffrements plus complexes vont apparaître mais aussi des nouveaux outils et méthodes vont aussi voir le jour. Notamment grâce à l'implémention/évolution des IA et de l'apparition des machines quantiques.

Références

[1] Cipher.exe documentation cipher.exe Documentation

[2] TcHunt (2017) - https://github.com/antagon/TCHunt-ng

[3] MAGNET Encrypted Disk Detector (2022) - https://www.magnetforensics.com/resources/encrypted-disk-detector

[4] Elcomsoft Encrypted Disk Hunter (2020) - www.elcomsoft.com

[5] AES Encryption AES

[6] Chiffrement par bloc cbc

[7] Data Encryption Standard ecb

[8] Encrypted Files Extentions - popular encrypted files extentions - Lists of encrypted files extentions

[9] Support Vector Machines svm

[10] camellia chiffrement camellia

[11] BlowFish chiffrement bf

[12] cast5 chiffrement cast5

[13] Rc4 chiffrement rc4

[14] GPG gpg

Pour une meilleure compréhension du projet, vous pouvez également lire l'état de l'art sur la détection de fichiers chiffrés à partir de la page 11, effectué en première partie de projet du Master 1 par demande de nos tuteurs de projet.

7 Annexe

Crypted Or Uncrypted - Part 1

FOUCHÉ Stanislas, PRIOU Antoine

7.1 Introduction

The security of data has become a major concern in today's world. Information equates to power and plays a crucial role. At the core of this concern lies the fundamental element of data security : encryption. Encryption is a process that transforms data into an unreadable format, known as encrypted. This technique aims to protect data from the risk of being compromised or stolen by the intrusion of unwanted individuals.

The main objective of this state of the art aims to explore in depth the different existing identification methods and possible developments in the future.

We can then wonder about this reflection : How can we determine whether a computer contains encrypted files or not ?

Firstly, we will explore how files are encrypted, followed by an examination of how encryption software operates. Then, we will delve into the various methods and tools used to identify encrypted files. Finally, we will focus on the future evolution of methods for identifying encrypted files

7.2 Encryption methods

Encryption algorithms, as their name suggests, are the algorithms used to encrypt and decrypt data sent and received. The principle behind an encryption algorithm is that it cannot be deciphered by anyone who does not possess the encryption key. These algorithms can be used for a number of purposes. The most common use are for encrypting bank data, end-to-end encrypted messaging, and HTTPS protocol.

7.2.1 Symmetrical encryption

A symmetrical encryption algorithm is an encryption algorithm where the key used to encrypt the data is the same as that used to decrypt it. In other words, with a message "A" to be encrypted, the Sender "S" must encrypt the message "A" with the key "K", and the Recipient "R" must then decrypt it with the key "K".

This encryption method poses a number of major problems. As the key is shared between the sending and receiving peers, the chances of the key being leaked are high. Each key must be shared between all the participants in the "conversation", so each person has the same key. However, with each additional person participating in the conversation, the risk of key leakage is exponential. This poses a problem because when the key is compromised, all the data encrypted with that key is compromised, both the data to be transmitted and the data previously transmitted. What's more, key transmission is an even more difficult problem to manage than key management, because all it takes is for one person to pretend to be someone else for security to be compromised.

7.2.2 Asymmetrical encryption

An asymmetric encryption algorithm is an encryption algorithm that is supposed to solve the major problems caused by symmetric encryption. This algorithm is used in the following way : Two keys are generated, a public key and a private key. The public key is used to encrypt transmitted data, and the private key is used exclusively for decryption. A public key is linked to one and only one private key. This solution therefore solves the problem of key management in symmetric encryption.

Although this technique solves the problems of symmetric encryption, it still poses some major problems. The main problem is performance. Managing keys in this way is very costly and complex for both the recipient and the sender, so transactions can take longer to process. The keys also have to be longer to ensure an adequate level of security, which also leads to performance problems.

These two types of encryption are nevertheless susceptible to common problems. The first is the fact that if a user loses their key in any way, they are no longer able to decipher the messages they receive. The other problem that arises as technologies evolve ever more rapidly is vulnerability to quantum attacks, a threat that is becoming increasingly preoccupying as time goes by. [13]

7.2.3 Encryption software

Encryption has become an essential component of IT security, ensuring the confidentiality and integrity of data. Encryption software plays a key role in this protection, providing solutions to secure files, disks and digital communications. The applications of encryption software are diverse, ranging from protecting personal information to securing business communications. Here are some of the most common uses :

- **Protection of Sensitive Files :** Encrypt individual files to prevent unauthorized access.
- **Disk Encryption :** Secure your entire hard drives to ensure the confidentiality of all stored data.
- **Securing Communications :** Encrypt emails and messages to ensure confidentiality of exchanges.

There are several encryption methods, including symmetric and asymmetric encryption. The former uses the same key for encryption and decryption, while the latter uses a separate key pair.

description of common software With the description that we are going to give, we will highlight the encryption methods and algorithms used by current encryption software. What is important in this point, is that if we know how the common software encryption method works. So we know how to detect them because the algorithms used take a certain form in the machine's storage. Also, knowing the type of support that the software supports for encryption allows you to rule out possibilities of encryption methods or software.
For example, if software only supports removable formats (USB type), I can exclude software that does not support this medium to check whether the device is encrypted or not.

- **BitLocker :** [1] BitLocker is an encryption feature built into Windows operating systems, developed by Microsoft. It encrypts the entire volume where the operating system is installed, as well as other data volumes you choose to protect. It generally uses the AES [15] encryption algorithm with a key of 128, 256 It uses symmetric encryption to protect data, and access to encrypted information requires authentication, usually in the form of a password, USB drive, or a combination. It can leverage the TPM [16] to store the volume encryption key. This adds a layer of security by preventing the system from booting from another disk.
- **TrueCrypt :** [4] TrueCrypt developers have stopped development of the software, suggesting users migrate to other encryption solutions. Available on all Windows, Linuw and macOS platforms and open source. It has the ability to encrypt entire disks, partitions or even removable storage devices. TrueCrypt supports several encryption algorithms, such as AES[15], Serpent[22], and Twofish[21]. TrueCrypt used different modes of operation, including XTS[20] (XEX-based Tweaked CodeBook mode with CipherText Stealing), CBC[19] (Cipher Block Chaining) It generated encryption keys from the password provided by the user. The software also applied iterations and salting [17] to strengthen the security of the key derivation. Such as PBKDF2 [18] (Password-Based Key Derivation Function 2) TrueCrypt stored configuration information and encryption keys in a volume-specific header.
- **VeraCrypt :** [3] VeraCrypt is free, open source software that provides disk, file, and partition encryption. It is an evolution of its predecessor, TrueCrypt, and is designed to provide enhanced security and improved functionality and remains usable on all platforms. It offers several encryption algorithms, such as AES[15], Serpent[22], and Twofish[21]. The user can choose a specific algorithm or opt for a combination (cascade) of these algorithms to enhance security. The software generally uses the XTS[20] mode of operation, which is suitable for disk encryption operations VeraCrypt generates encryption keys from the password provided by the user. Key derivation is performed using a hash function, and iterations are applied to strengthen security. It uses salting[17] and iteration techniques to make brute force attacks more difficult. VeraCrypt offers a hidden volume feature, which allows the user to create an encrypted volume inside an encrypted volume. Finally, it integrates mechanisms to resist attacks aimed at recovering encryption keys by analyzing the RAM of a system that has been turned off (Cold Boot Attacks)[23].

— **GPG(GnuPG) :** [2] GnuPG is a free implementation of the OpenPGP [24] standard, which is an encryption and digital signature protocol. GnuPG is widely used for file encryption, protecting the privacy of email communications, and verifying file authenticity.

GnuPG uses asymmetric cryptography, which is based on a pair of keys : the public key and the private key. The public key is shared and can be used by anyone to encrypt messages or verify signatures, while the private key is kept secret and is used to decrypt messages or sign documents. GnuPG uses a trust model called the "Web of Trust"[25]. Users can sign others' public keys to indicate that they trust the association between the key and the user's identity. Public keys can be published on key servers, allowing users to retrieve public keys centrally.

7.3 Encrypted file identification

Encrypted file identification methods uses mostly various mathematical techniques, such as computing Shannon entropy, chi-square, mean, Monte Carlo simulations, and Serial Correlation Coefficient . These methods are employed to discern patterns, characteristics, and anomalies associated with encrypted data, aiding in the identification and analysis of concealed information. Another method would be to simply look in the disk if there is traces or complete programs to encrypt files installed.

File signatures or headers can help too as they are unique sequences of bytes that identify the file type. Some encrypted containers, like those created BitLocker have an identifiable FVE-FS signature at the beginning of the partition.

Examining file metadata, such as creation and modification dates, can also provide insights into potential encryption.

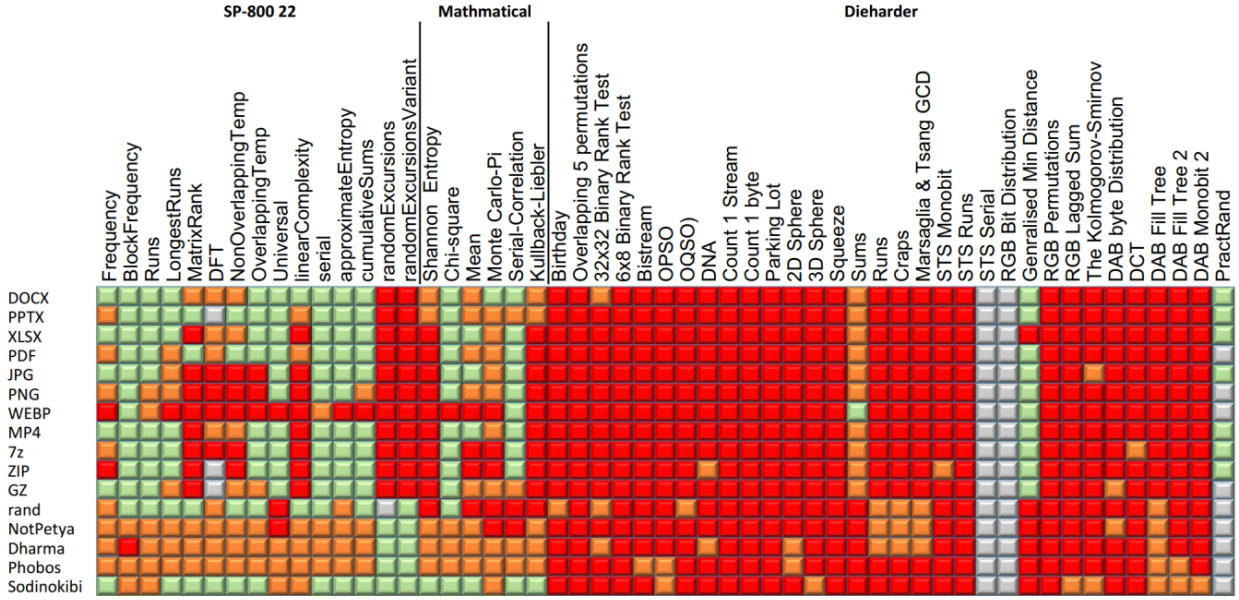


FIGURE 5 – Methods accuracy

Color coding : | 100 % Success | 80 % Success | Bellow 80 % Success | Failure |

However we observe various accuracy results depending on the method used as shown in Figure 1 diagram made by the Edinburgh Napier University [1] by evaluating randomness to differentiate crypto-ransomware encrypted files and common non-encrypted files using 53 distinct tests against a dataset of 320 files, providing nearly 17,000 results.

7.3.1 Methods/Tools

find file extensions

The simplest way to determine if a file is encrypted is to look at its extension.

Some encryption software does not directly conceal encrypted files and saves them in a format specific to the encryption software, making them visible under a particular extension.

The first thing to do is to check for the presence of encryption software. If there is any, you can verify the presence of corresponding encrypted files through a file extension search, using either the command prompt (cmd) or file manager.

For example, if the software AxCrypt is installed, one can check for the presence of .axx files using the cmd.

Under Windows, one can use the command : `dir /s RootFolder/*.axx` (/s performs a recursive search in all subdirectories of the specified folder)

If no encryption software is installed, one can still check if a particular file extension seems suspicious. To verify if it might be an encrypted file, try to open it or check if the extension is associated with encryption software. Referring to a list of known file extensions can also be helpful. [7]

cipher.exe (Window)

Cipher.exe is a built-in tool in Windows that allows encryption, decryption, and identification of encrypted files. From the local disk, one can search for the presence of encrypted files using this tool.

Cipher.exe is easy to use, but it primarily focuses on detecting encryption at the file level rather than on a broader scale, such as at the partition or entire disk level.

Launch a cmd and use the command below :

```
cipher /u /n
```

/u [/n] Finds all encrypted files on the local drive(s). If used with the /n parameter, no updates are made.

If used without /n, /u compares the user's file encryption key or the recovery agent's key to the current ones, and updates them if they have changed. This parameter works only with /n. [8]

TcHunt

TCHunt[9] is a command line tool that can be used to find encrypted True Crypt volumes on the system. It has been specifically designed to demonstrate the possibility of finding True Crypt volumes even if they are not mounted or hidden by the user.

The program aims to demonstrate that it can discern TrueCrypt containers, even when they are relatively small and deliberately disguised by the user. Verifying the presence of a TrueCrypt container without technical assistance is challenging, especially if the container is modest in size or strategically placed for inconspicuity because the sample is too small to be analyzed properly with mathematical methods for example. While it is conceivable to scrutinize each potential container file on a system, this process could be time-consuming.

TCHunt systematically scans a designated folder or partition on the computer, examining four specific attributes inherent to TrueCrypt volumes :

- The file size modulo 512 must be zero.
- The file size should be at least of 19 KB.
- The file contents must pass a chi-square distribution test.
- The file must not contain a common file header.

TCHunt then returns : likely to be encrypted, file not encrypted or error/interruption.

TCHunt underscores the feasibility of detecting TrueCrypt volumes, even when they are not actively mounted. However, it is important to note that the program does not possess the capability to brute force or circumvent the encryption itself. TCHunt-ng is an open-source software licensed under GPLv3, last updated in 2017, with its code available on Github. It is compatible with both Linux and Windows operating systems.

Limitations : TCHunt-ng has no way to tell apart a genuinely encrypted file and a file made up of random data. Files smaller than 32 bytes, unless recognized by its type, are ignored. (source https ://github.com/antagon/TCHunt-ng)

MAGNET Encrypted Disk Detector MAGNET Encrypted Disk Detector [10] is a command line tool that can be used to find encrypted volumes on a computer. Encrypted Disk Detector checks the local physical drives on a system for TrueCrypt, PGP, VeraCrypt, SafeBoot, or Bitlocker encrypted volumes. If no disk encryption signatures are found in the MBR, the program also displays the OEM ID and, where applicable, the Volume Label for partitions on that drive, checking for Bitlocker volumes. (source : https ://www.magnetforensics.com/resources/encrypte disk-detector)

Encrypted Disk Detector scans only for encrypted partitions, unlike TCHunt, which can scan for volumes and containers.

Encrypted Disk Detector (last updated in 2022) is a free and operating on Windows but cannot be downloaded directly and asks to fill a form on the website to receive the installer.

Machine learning There are several applications for using machine learning to detect encrypted files. The most convincing example is that used to detect crypto-ransomware. Using a dataset of thwarted previous ransomware attacks, the machine learning algorithm will determine similarities between the files scanned and those in its dataset, while trying to consider unknown cases. The advantage is the versatility of this tool, which can be effective in detecting new things, although its application outside ransomware is not yet developed. [5]

Encryption pattern detection method

- **Entropy Analysis :**
  Entropy is a measure of disorder in a data set. Encrypted files often have higher entropy than unencrypted files. Detection tools can measure the entropy of a file to determine whether it has encryption characteristics.
- **Byte Distribution Analysis :** Some encryption algorithms may generate non-uniform byte distributions in encrypted files. Byte distribution analysis can help identify patterns characteristic of certain

types of encryption.

- **Finding repetitive sequences :** Finding repetitive sequences : Some encryption algorithms generate blocks of encrypted data that have repetitive sequences. Detection tools can look for these patterns to identify encrypted files.
- **Block size analysis :** Encryption algorithms use different block size schemes to encrypt data. Analyzing the block size in a file can help identify the type of encryption algorithm used.
- **Study of specific headers :** Some encryption software adds specific headers to encrypted files to indicate the type of algorithm used. Analyzing these headers can provide useful information about the encryption applied.
- **Pattern Absence Analysis :** Some encryptions completely erase patterns and structures from the original files. Detecting the absence of patterns can be an indicator of the presence of encryption.
- **Metadata analysis :** Analyzing file metadata can reveal information about the encryption methods used, even if the contents of the file itself remain unreadable.

7.4 Elcomsoft Tools

Elcomsoft is a company specialized in the development of computer security software and data recovery solutions.

They specialize in highly useful forensic tools, especially for law enforcement professionals.

Elcomsoft Encrypted Disk Hunter

Elcomsoft Disk Hunter is designed to identify encrypted volumes created by software such as BitLocker, TrueCrypt, VeraCrypt, and other popular encryption tools.

It is designed to analyze file systems and detect the presence of these encrypted volumes, even if they are not currently mounted or if they are hidden.

It conducts a thorough search for signs of full disk encryption. If no obvious signs of disk encryption are found, Encrypted Disk Hunter will then check the system driver chain for TrueCrypt, VeraCrypt, and PGP WDE drivers.

For resume :

- Search for specific signs of encryption or traces left by common encryption software.
- Search for indicators or specific metadata that could indicate the presence of encrypted volumes (hidden or mounted).
- Search for the presence of drivers associated with encryption software

Here's a detailed example of usage : [14]

Elcomsoft Disk Decryptor

Even if one has obtained an encrypted file, finding the key to decrypt it can be challenging. However, there are tools available that can aid in decrypting such files.

The best decryptor at the moment is forensic Disk Decryptor Tool from Elcomsoft.

The Elcomsoft Disk Decryptor is a forensic tool used in the field of computer security and digital investigation. It is designed to decrypt encryption systems such as TrueCrypt([4]), BitLocker([1]), PGP , etc. This tool can assist security experts and investigators in accessing data protected by these encryption methods by using specialized techniques and appropriate decryption methods.

Using one of the three search methods offered, it will be able to retrieve encryption keys, as well as identify the algorithms that facilitated the recovery of these keys.

- By analyzing the hibernation file
- By analyzing a memory dump file
- By performing a FireWire attack

For more details, we invite you to consult the sources used for Forensic [11].

7.5 Evolution

The evolution of encrypted file identification methods depends on the advancement of cryptography.

7.6 IA integration

The current integration of AI and machine learning to analyze file behavior and detect specific cryptographic patterns could be further developed. This might allow for identifying encrypted files based on more complex, reliable, and faster characteristics.

7.6.1 post-quantique/quantique

The advent of quantum computers will revolutionize the field of cryptography. Encryption methods could pose challenges to identification techniques.

Research in post-quantum cryptography is progressing. New encryption algorithms resistant to quantum attacks could emerge, making it more difficult to identify encrypted files using these new algorithms.

7.7 Conclusion

So, to answer our problem on how to determine whether a computer includes encrypted files or not, we first explored the way in which a file can be encrypted. In order to better understand how identifiers.

We identified the two main encryption algorithms (symmetric and asymmetric) and how encryption software uses them. Following that, we analyzed different methods to identify the presence of encrypted files. Initially, we began with a straightforward approach using file extensions, and then progressed to more complex identification using software that searches for encryption traces. This involved searching for encrypted partitions through the identification of volumes and containers or by examining drivers.

Finally, there's a consideration regarding the potential evolution of encrypted file identification, involving the integration of AI (machine learning) and quantum or post-quantum methods.

Given the constant evolution of encryption technologies, there is a continual need to adapt methods and tools to ensure precise and effective detection.

Références

[1] BitLocker - Software BitLocker

[2] GPG (GnuPG) - Software GPG (GnuPG)

[3] VeraCrypt - Software VeraCrypt

[4] TrueCrypt- TrueCrypt - SourceForge.net

[5] Machine learning - Crypto-ransomware detection using machine learning models in file-sharing network scenarios with encrypted traffic

[6] Davies S.R, Macfarlane R, Buchanan W.J Comparison of Entropy *Calculation Methods for Ransomware Encrypted File Identification (2022)* - https://arxiv.org/abs/2210.13376

[7] Encrypted Files Extentions - popular encrypted files extentions - Lists of encrypted files extentions

[8] Cipher.exe documentation cipher.exe Documentation

[9] TcHunt (2017) - https://github.com/antagon/TCHunt-ng

[10] MAGNET Encrypted Disk Detector (2022) - https://www.magnetforensics.com/resources/encrypted-disk-detector

[11] Forensic Disk Decryptor - Elcomsoft Forensic Tools

[12] Peter Shor - Shor's algorithm (1994) - Shor's algorithm

[13] Vasileios Mavroeidis, Kamer Vishi, Mateusz D. Zych, Audun Jøsang - The Impact of Quantum Computing on Present Cryptography

[14] Elcomsoft Disk Hunter - Elcomsoft Disk Hunter - Elcomsoft Disk Hunter exemple

[15] AES *(Advanced Encryption Standard) is an encryption algorithm that takes a 128-bit data block and an encryption key (128, 192, or 256 bits). It works in several repeated steps (rounds) to mix and confuse the data, thus ensuring security.* - link AES

[16] TPM *Trusted Platform Module is an international standard for a secure cryptoprocessor, a dedicated microcontroller designed to secure hardware via embedded cryptographic keys.* - link TPM

[17] Salage/salting *method of increasing security by chopping and adding additional.* - link salting

[18] PBKDF2*used to reduce vulnerability to brute force attacks* - link PBKDF2

[19] CBC - IBM-CBC

[20] XTS - IBM-XTS

[21] Twofish*is a symmetric block cipher algorithm* - link Twofish

[22] Serpent*block cipher algorithm* - link Serpent

[23] Cold Boot Attacks - link CBA

[24] OpenPGP - link OpenPGP

[25] Web of Trust - link Web of Trust