A Forensics Approach to Blockchain

Master Thesis in Telecommunications Engineering

Álvaro Borreguero Beltrán

Barcelona 2019

Department of Telematics
Universitat Politèctinca de Catalunya
Spain
Date

Abstract

A Blockchain is essentially a distributed database of records, or public ledger of all transactions or digital events that have been executed and shared among participating parties. Each transaction in the public ledger is verified by consensus of a majority of the participants in the system. Once entered, information can never be erased. This allows participating entities to know for certain that a digital event happened by creating an irrefutable record in a public ledger. This participants are anonymous users which can be identified under a Bitcoin address.The increased use of this cryptocurrencies such as Bitcoin among private users and some businesses has opened a new avenue of research in the field of digital forensics involving cryptocurrencies.

The anonymity of Blockchain has made cryptocurrencies a de facto standard payment in the black market or deep web. Users who buy in this kind of services, protect their identity by means of the natural protection provided by cryptocurrencies. Because of this, the vast majority of payments which take place in TOR are made through this kind of payment methods.

This paper describes the steps that should take place in a crime investigation involving cryptocurrencies, from the alert of a crime being committed up until the support or refute of the hypothesis and proofs before a court.

One of the most important steps is to catch or identify the identity of cryptocurrency users or in this case "cybercriminals". In order to achieve this, it is necessary to try and exploit the infrastructures of both Bitcoin and TOR networks. If a positive coincidence is found when exploiting any of previous networks, relate or link a user to a specific Bitcoin address can help identify him in real life. After the identification has been successful, the next step will be to examine the user's personal data.

Once the identity has been discovered, we will make use of "deeper" Forensic analysis, which will help maintain valid any evidence needed in order to proof a criminal guilt before a court. This will be achieved by following a computer forensic process which prevents original data from being altered.

Acknowledgments

To both my Parents, my brother, Elena and my family, who were there to give me their unconditional and constant support in different ways, but all of them equally valid and appreciated. Which have been my main pillars during all this period of time, and whom without, I would not have been able to achieve this. To my best friends Carlos, Manuel and Javier, for their help in times of need, who joined me via Skype to work and keep in touch, who where there to speak during stressful moments or spent weekends working each one on his thesis but helping each other when required. To Josep Pegueroles, my tutor, who stood up to my indecision and many irrational proposals, and guided me to a better and more realistic approach. Finally, to SegInfoProv y P4UK who have accompanied me this last year and helped me advance and facilitated me meetings during my work time.

Personal Motivation

The main motivation of this project is to study one of the most relevant technologies today and it's possible security vulnerabilities, and once found, try and exploit them.

Blockchain is a technology fairly new to most of the population. It has not been yet implemented in our day to day, therefore, it's pertinent to study and understand which are it's vulnerabilities prior to its possible mass implementation.

One of the motivations that led me to this idea, was the study of a way of identifying the growing amount of cybercriminals that use cryptocurrencies as their standard method of payment. Since one of the main innovations of Blockchain transactions is anonymity, I believe it is important to identify those users who use this technology for illegal purposes.

Table of Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Background

We are living in an era of big technological innovations with infinite positive contributions, but as we know, every new IT innovation has a security requirement behind it. This requirements get more and more demanding as offensive tools evolve and increase in sophistication.

Since the creation of Bitcoin and later altcoins, and their respective Blockchains, cryptocurrencies have become a fancy and very useful way of payment for many users, which offers a fast and almost tax fee free transactions worldwide. The anonymity of the payments and the exclusion of trusted third parties needed to verify transactions made cryptocurrencies become a de facto standard payment in the dark web. This tool has been increasingly used by criminals to perform illegal transactions. This transactions majorly include:

- Drugs
  - http://kbvbh4kdddiha2ht.onion/
  - http://fzqnrlcvhkgbdwx5.onion/
  - http://newpdsuslmzqazvr.onion/
  - http://k4btcoezc5tlxyaf.onion/
  - http://mlj4iyalawb2ve2u.onion/
  - http://xdsa5xcrrrxxxolc.onion/
  - http://abyssopyps3z4xof.onion/products
  - http://mollyworh4524fop.onion/
- Guns
  - http://tuu66yxvrnn3of7l.onion/
  - http://drkseidwayn6uc5x.onion/catalogue.html
  - http://2kka4f23pcxgqkpv.onion/
  - http://q7hj46bbqorjthmq.onion/index.php/shopping/index/arms
  - http://gunsdtk47tolcrre.onion/
- Pornography

- Hackers
  - http://zv2pk3rt7lspysus.onion/
  - http://2ogmrlfzdthnwkez.onion/
  - http://hackerrljqhmq6jb.onion/
  - http://doxTORg7natnwyz5.onion/
- Hitman Contracts
  - http://darkmambawopntdk.onion/
  - http://yo4jmu6dsfaeekt3.onion/
  - http://hitman.TORpress2sarn7xw.onion/
- Money Laundering
  - https://cryptomixer.io/TOR/
  - http://blenderiocpxfema.onion/
  - http://coinpigih6i444lm.onion/
  - http://penguinsmbshtgmf.onion/
  - http://bitmixbizymuphkc.onion/en
- Ponzi Scheme (Investment with x% interest)

A report came out on 2011 where a user of Bitcointalk.org threatened the Bitcoin community with an injection of custom data in the Blockchain.[16]
In 2013 a user named Scintill alleged to have extracted data from two transactions in the Bitcoin Blockchain where *"some Hidden Wiki pages ("Jailbait", "Hard Candy") with links to pedo communities and stuff"* had been stored.[15]

- https://Blockchain.info/tx/dde7cd8e8f073a525c16c5ee4e4a254f847b7ad6babef257231813166fbef551
- https://Blockchain.info/tx/4a0088a249e9099d205fb4760c28275d4b8965ac9fd56f5ddf6771cdb0d94f38

Another main issue concerning security in the Blockchain is the possibility of theft or any other malicious attack that can be performed. A study has revealed that every day, around $9 Million are lost to cryptocurrency scams, hacks, thefts, frauds or phishing.[1]

Since the creation of Bitcoin there have been numerous thefts, it has been calculated that around 18% of the total amount of cryptocurrencies have been stolen.

[1]This amount only takes into consideration thefts that occurred in the first half of 2018. `https://news.Bitcoin.com/9-million-day-lost-cryptocurrency-scams/`

- https://www.ccn.com/biggest-theft-history-know-far-530-million-coincheck-hack
- http://money.cnn.com/2017/12/07/technology/nicehash-Bitcoin-theft-hacking/index.html
- https://www.investopedia.com/news/indian-exchange-coinsecure-hit-35-million-Bitcoin-theft
- https://coinjournal.net/uk-company-linked-to-the-theft-of-650000-Bitcoins-from-mt-gox
- http://time.com/money/5053744/hackers-steal-Bitcoin-nicehash
- https://www.reuters.com/article/us-crypto-currency-crime/about-1-2-billion-in-cryptocurrency-stolen-since-2017-cybercrime-group-idUSKCN1IP2LU
- https://www.coindesk.com/unconfirmed-report-5-million-bitstamp-Bitcoin-exchange
- https://medium.com/@MikeBacina/1b-lost-the-5-biggest-cryptocurrency-fails-of-2017-9862131e2bf7
- https://news.Bitcoin.com/9-million-day-lost-cryptocurrency-scams
- http://fortune.com/2018/02/14/Bitcoin-cryptocurrency-Blockchain-wallet-hack

The issues exposed above have taken experts to start investigating ways of tracking transactions in order to locate the responsible authors of these criminal activities. Although some crimes have been solved, the vast majority are still unsolved.The most renown case was Silk Road.

1.2 Scope of the Thesis

Given what has been indicated in the background, this project intends to explain at a high level, the process since a crime has been identified up to the presentation of the necessary evidences obtainable from the target's (criminal) PC. For that, we will have to discover and identify different methods to de-anonymize cybercriminals, which use Bitcoin and TOR as their preferred exchange and network respectively. Once the criminals have been identified, the existent frames in a client PC, where transaction or movements inside Bitcoin's Blockchain have taken place, will be analysed.

In order to achieve this, its crucial to completely identify the user that is behind a certain Bitcoin address. The following will be studied and explained: What is Bitcoin

and Blockchain and how do they work, what is TOR and how does it work, techniques to de-anonymize this kind of users and once identified, which are the relevant frames for the investigation and how to gain access to them.

1.3 State of the Art

We will summarize the state of the art of the three main aspects of this study. In this point, we will give a brief historical explanation of the technology and which is the current state of this.

The three main points of study of the project are:

- Cryptocurrencies
- Anonymous Communications
- Digital Forensics

1.3.1 Cyrptocurrencies

Digital currencies based on public key cryptography, cryptocurrencies, differ from other types of currencies as they are able to provide pseudonymity naturally. There are several notable examples of pseudonymous cryptocurrencies.

E-cash - Ecash was presented in 1983 by David Chaum, who introduced blind signatures and ecash [23], an anonymous payment system based on them. Coins were issued by a bank which were blindly signed and didn't belong to a specfici user. This made the coins anonymous. Ecash was implemented by "DigiCash" company in 1989. Despite some initial success the company went bankrupt in 1998 (one of the possible reasons is the wide adoption of credit cards for online payments).

Ecash was a centralized digital currency as there is one problem which is not easy to solve without introducing a trusted third party: double spending. In contrast to physical cash, electronic coins are very easy to copy. Thus the common solution for a merchant is to consult with the bank to make sure that a coin was not previously spent.

B-money - B-money was presented in 1998 by Wei Dai[24], as a design for distributed digital currency. It included several important ideas:

- Both parties hide their identities behind public keys.
- Anyone can create money by broadcasting the solution to a previously unsolved computational problem.

- In order to transfer money, a user needs to create a transaction with the receiver's public key and the amount, the user then signs and broadcasts the transaction.
- The information about who has how much money is kept in a distributed fashion on a set of servers.

.

However the protocol for B-money was never publicly published. This approach was a good initial idea, however there were some flaws still to be fixed.

Bitcoin - *Bitcoin was the first is a decentralized digital currency created by an unknown person or group of people under the name Satoshi Nakamoto and released as open-source software in 2009. It does not rely on a central server to process transactions or store funds. There are a maximum of 2,099,999,997,690,000 Bitcoin elements (called Satoshis, the unit has been named in collective homage to the original creator), which are currently most commonly measured in units of 100,000,000 known as BTC. There will only ever be 21 million Bitcoin (BTC) to ever be created.* - Bitcoin Wiki=`https://en.bitcoin.it/wiki/Bitcoin`

Blockchain or originally called "Block Chain", was invented in 2008 to serve as the public distributed ledger for Bitcoin. The distributed ledger[2] records and stores transactions made between two entities in a verifiable and permanent way. A block can't be erased once it has been verified and has entered the blockchain. The list of records are called *blocks*, in order to add a new block to the chain, miners[3] have to solve a difficult problem, and then verified by the rest, to successfully add a new block to the chain.

The invention of the Blockchain for Bitcoin made it the first digital currency to solve the double-spending problem without the need of a trusted authority or central server. This invention served as a guidance for several other Blockchains or cryptocurrencies that have appeared since then.

[2]A distributed ledger is a consensus of replicated, shared, and synchronized digital data geographically spread across multiple nodes. The primary advantage is the lack of central authority or centralized data storage.

[3](Miners are a individual or group of parties who compete to solve a difficult mathematical problem first. Once the problem has been solved, the entity who solves it shares it with the rest of miners to verify the validity of the solution. If the consensus agrees on the solution, the block is then added to the Blockchain. This process repeats itself every 10 minutes approximately, where all the miners compete to be the first to mine the block, as it will be the winner the one who will receive the reward.)

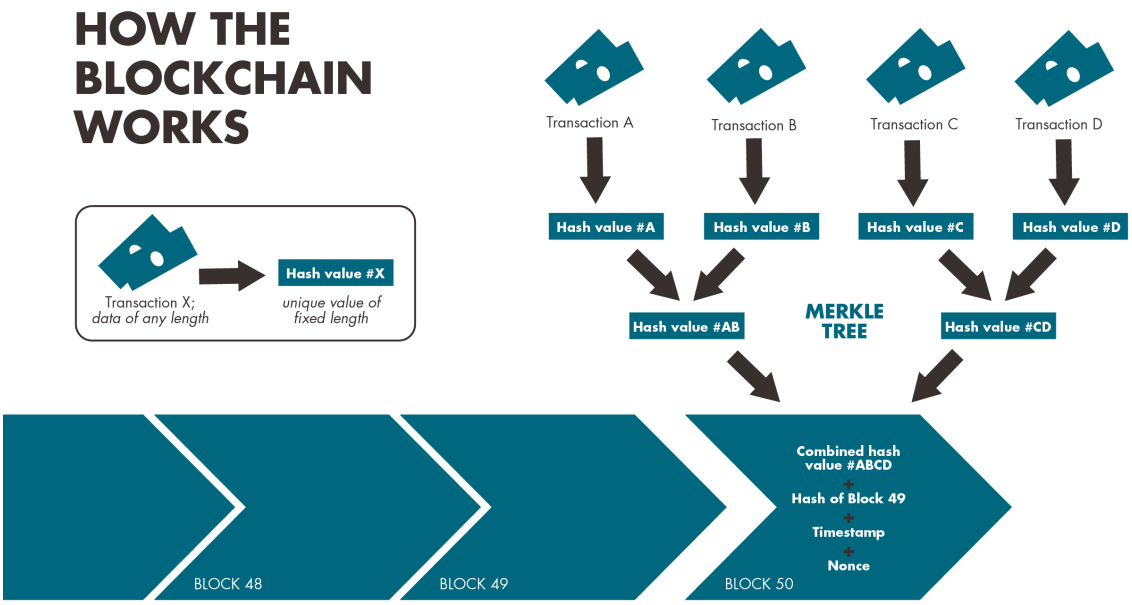

FIGURE 1.1. Blockchain Workflow - https://mn.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Blockchain_workflow.png

One of the main innovations behind Bitcoin is the solution of all the issues that came with the Double Spending process. The main points are the following:

- Details about the transaction are sent and forwarded to all or as many other miners as possible.
- A constantly growing chain of blocks that contains a record of all transactions is collectively maintained by all computers (each has a full copy).
- To be accepted in the chain, transaction blocks must be valid and must include proof of work (one block generated by the network every 10 minutes).
- Blocks are chained in a way so that, if any is modified, all following blocks will have to be recomputed.

When multiple valid continuations to this chain appear, only the longest branch is accepted and it is then extended further, whilst the rest are discarded.

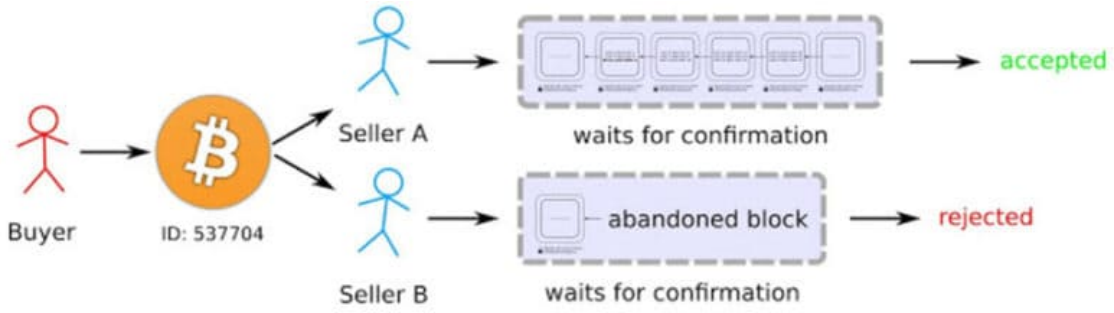

FIGURE 1.2. Double Spending - https://www.reflectionsofthevoid.com/2015/11/bitcoin-double-double-trouble.html

When Bob sees that his transaction has been included in a block, which has been made part of the single longest and fastest-growing Blockchain (extended with significant computational effort), he can be confident that the transaction by Alice has been accepted by the computers in the network and is permanently recorded, preventing Alice from creating a second transaction with the same coin.

Altcoins are the alternative cryptocurrencies launched after the success of Bitcoin. Generally, these new projects tend to fix existing flaws found in Bitcoin, try to optimize the procedure or the way it works, or in some cases give the technology a spin introducing new concepts (such as Solidity, an object-oriented programming language for writing smart contracts). The success of Bitcoin as the first peer-to-peer digital currency has given place to many other cryptocurrencies that have appeared in the past few years. Three of the most important altcoins identified which also work under a different Blockchain technology are:

- **Ethereum (Ethash)** - Ethereum is a decentralized platform that runs smart contracts. Smart contracts are elf-executing contracts where the terms of the agreement between buyer and seller have been written into lines of code. This enables developers to create markets, store registries of debts or promises, move funds in accordance with instructions given long in the past (like a will or a futures contract) and many other things that have not been invented yet, all without a middleman or counterparty risk.

- **Monero (Cryptonight)** - Unlike many cryptocurrencies that are derivatives of Bitcoin, Monero is based on the CryptoNight proof-of-work hash algorithm, which comes from the CryptoNote protocol. It offers a higher level of security and anonymity for users and their transactions. The Monero ledger, unlike

Blockchain, doesn't record the actual stealth addresses of parties to transactions, and the one-time created address that is recorded is not linked to the actual addresses of the parties as well.

All Monero transactions between two parties are mixed with other transactions that occur among unrelated parties (obfuscation). Monero also handles transactions by splitting the outgoing amounts into multiple parts and then treating each split part as a separate transaction. This way in a single transaction, there could be 5 senders parties and 7 receivers. The popularity of Monero is not growing just because of attention from black market, but also because of the large amount of people not fond of being watched by governments, hackers, and corporations.

- **Zcash (Equihash)** - Zcash uses a special proof to secure the network called zk-snark - or proof of construction. Unlike Bitcoin, where all the transactions are public, Zcash maintains a secure ledger of balances without revealing parties or amounts involved in transactions. This happens through the use of zero knowledge proofs.

Wallets
BItcoin Core
Electroneum
etc etc
Vulnerabilities

- **51% attack** - The "51% attack" can only be performed by an attacker, when he has more computing power then the rest of the network. If this occurs, he could be able to disconfirm others' blocks confirming only their own, and thus receive 100% of all new Bitcoins and block any transaction at their discretion. (However they won't be able to forward any transactions they want, because he will not have others' private keys to sign these transactions). Currently to conduct such attack in the Bitcoin network would require computing power many times more then the power of all TOP-500 rated supercomputers.

- **Denial-of-Service attacks (DoS)** - Sending a large number of "junk " data to the node that handles transactions may hinder its work. Bitcoin has built-in protection against attacks such as "denial of service" but today this type of attack becomes harder with each new attempt.

1.3.2 Anonymous Communications

TOR (The Onion Router) - is the most widely used anonymous communication network available online. TOR enables server-side anonymity through the design of

hidden services, also known as onion services. To achieve their anonymity goal, a hidden service client and operator establish a communication tunnel, known as a circuit, between each other over multiple intermediate routers.

Anonymity is maintained as long as the network relays are not controlled by an adversary who can use de-anonymization techniques to try and discover TOR users. Hidden services have also been subjected to active attacks in the wild. To ensure transaction anonymity, Bitcoin has become the most popular payment method by TOR hidden services. Unfortunately, this has contributed to the rise of illegal hidden services, such as Silk Road, which offers illicit merchandises and services.

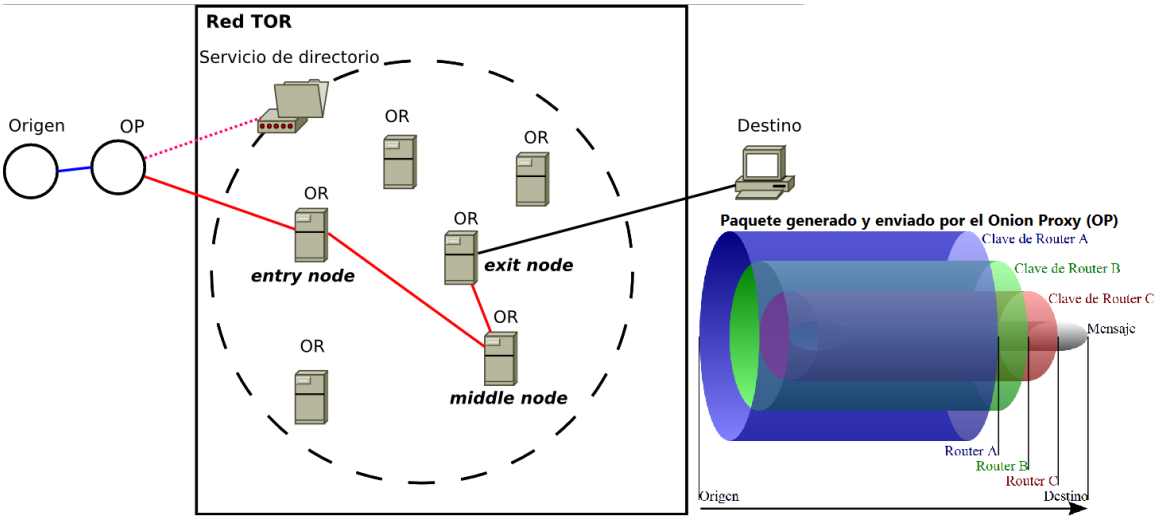

FIGURE 1.3. TOR structure - https://wikis.fdi.ucm.es/ELP/Archivo:Red_tor.png

Onion Routing - The TOR network utilizes Onion Routing as a way to anonymize communications. The term "onion routing" comes from the layered encryption used, resembling the layers of an onion. This allows Hidden Services to work as expected.

The messages sent between the sender and the receiver are transferred through a path of nodes called a "chain" or a "circuit". The nodes that form the chain are determined after the sender communicates with the directory node, who assigns three random nodes of the node pool. The message will travel hopping from node to node until it arrives at its destination, in a way that forbids any node to know whether the previous node is the sender or another node. Only the exit node knows its situation, as it sends the message directly to the receiver.

The message is transmitted using asymmetric key cryptography. First, the onion proxy communicates with the Onion Directory who indicates which will be the nodes in the chain. Then, the sender communicates with the nodes to negotiate the cipher key for each node. Once the sender knows each node encryption key, the Onion Proxy

wraps the message in an IP package with the original end host IP as the destination IP and the middle node as sender. This package is ciphered using the key negotiated with the last Onion Relay. The Onion Proxy repeats the process using the middle node (exit node as destination and entry node as origin) and lastly with the entry node. Only the sender has the three keys so that it can encrypt the data he sends in 3 layers. The final package has as many "layers" as Onion relays are involved in the chain (3 by default). The originator sends the message encrypted with the three layers of encryption to the entry node, who decrypts it with its key. The entry node then sends the message (with two encryptions left) to the middle node, who also decrypts the message and sends it to the exit node, who does the last decryption stage and finally sends the message to the destination. With this chain, the originator can communicate with the receiver anonymously. The chain can be used also from the receiver to the sender.

Hidden Services - When visitors connect to the TOR network, TOR resolves those .onion addresses and directs you to the anonymous service sitting behind that name. Unlike with other services, hidden services provide two-way anonymity. The server doesn't know the IP of the client, like with any service you access over TOR, but the client also doesn't know the IP of the server. This provides extra privacy since it's being protected on both sides.

Silk Road [4] is the most notable and re-known example of a successful symbioses of TOR Hidden Services and Bitcoin. Silk Road was an anonymous online market available as a TOR Hidden Service and launched in February 2011. Silk road was operated by a person under the pseudonym "Dread Pirate Roberts"[5]. In this market place the operator charge a fee to each seller as a rent for using the platform, it acted as a middle ware between both ends.

All operations were done through the SilkRoad escrow service: buyers' bitcoins were held by the SilkRoad until the order had been received. A mechanism was introduced which allowed sellers to opt for the value of bitcoins held in escrow to be fixed to their value in US$ at the time of the sale to mitigate against Bitcoin's volatility. Any changes in the price of bitcoins during transit were covered by Dread Pirate Roberts.The only way to access Silkroad was through TOR Hidden Services and buyers and sellers conducted all transactions with bitcoins.

[4]The name "Silk Road" comes from a historical network of trade routes started during the Han Dynasty (206 BC – 220 AD) between Europe, India, China, and many other countries on the Afro-Eurasian landmass.

[5]Silk Road was operated by the pseudonymous "Dread Pirate Roberts" (named after the fictional character from The Princess Bride), who was known for espousing libertarian ideals and criticizing regulation.

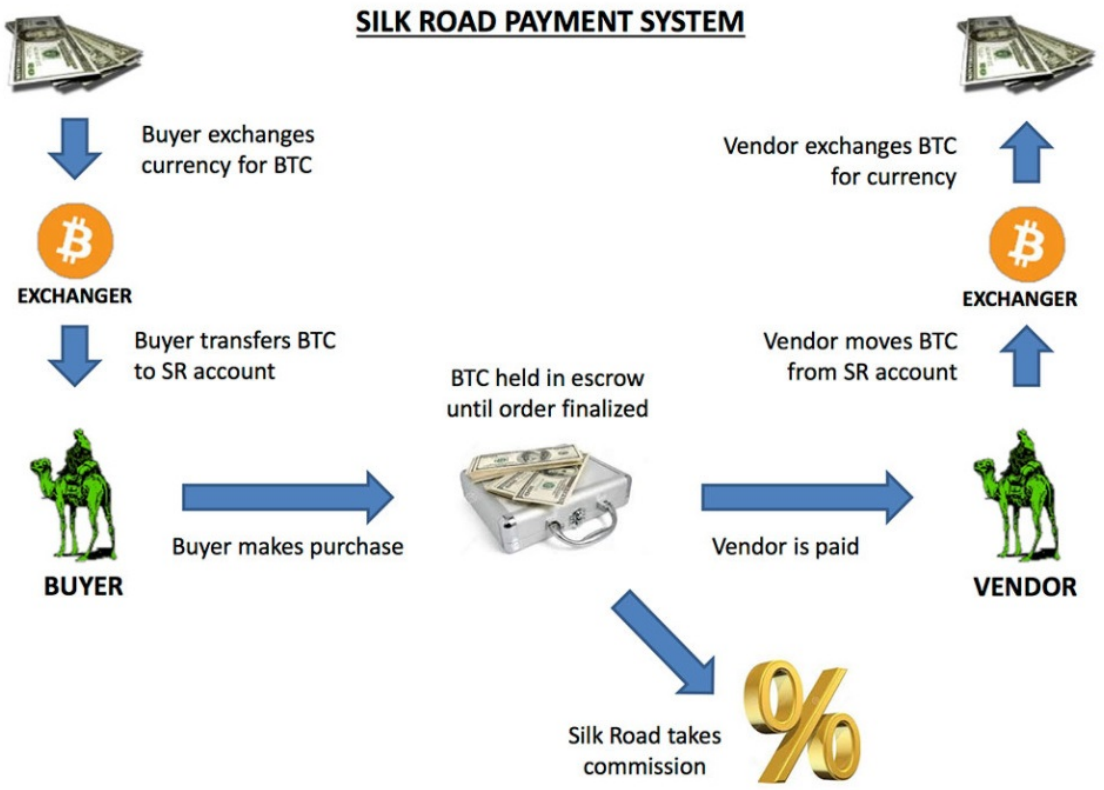

FIGURE 1.4. Silk road payment system - https://de.wikipedia.org/wiki/Silk_Road

After Silk Road was ceased by the FBI, several attempts of restoring the service appeared, most of them being closed short after their opening [11].[6]

The Invisible Internet Project - I2P is an anonymous network, similar to TOR, it is based on encryption layers but with the addition of a fourth layer. However, is strictly message based, IP (TOR supports IP, FTP, HTTP/s and other protocols). All communication is end to end encrypted, and even the end points ("destinations") are cryptographic identifiers (Hidden Services).

1.3.3 Digital Forensic Analysis

Computer Forensics*, or* ***Digital Forensics*** , is a fairly new field. Digital forensics includes the uncovering and describing the information contained on, or the state

[6]The latest version of the Silk Road (Silk Road 3.1) active today can be found under the following Hidden Service: `http://silkroad4n7fwsrw.onion/?road`.

or existence of, a digital artefact, this involves the preservation, acquisition, documentation, analysis, and interpretation of evidence from various storage media types. Digital artefacts include computer systems, hard drives, CDs, and other storage devices, as well as electronic documents and files like emails and JPEG images. The fast-growing field of computer forensics includes several branches related to firewalls, networks, databases, and mobile devices. Digital forensics technicians can find work with many types of organizations: government, accounting firms, law firms, banks, and software development companies. The most common is to support or refute a hypothesis before criminal or civil courts. Essentially, any organization that has a computer system may have a need for a digital forensics specialist.

Autopsy is one of the most widely used forensics tool, it is the graphical user interface (GUI) used in The Sleuth Kit. It is very simple to operate and use, automating many of the procedures. It helps to identify, sort and catalogue relevant pieces of forensic data in an ongoing investigation.

Some of the modules provide:

- Timeline Analysis - Advanced graphical event viewing interface.
- Hash Filtering - Flag known bad files and ignore known good.
- Keyword Search - Indexed keyword search to find files that mention relevant terms.
- Web artefacts - Extract history, bookmarks, and cookies from Firefox, Chrome, and IE.
- Data Carving - Recover deleted files from unallocated space.
- Multimedia - Extract EXIF from pictures and watch videos.
- Indicators of Compromise - Scan a computer using STIX.

When performing a digital forensic analysis, doing a **binary copy** is one of the most important things to do during an investigation. It is required to get the information but without accessing to it, to do so, a copy of all the 0 and 1 that the device contains has to be performed. To proof that the binary copy is identical to the original artefact, a hash of both files has to be computed. Once this hash function has been completed, in order to prove that the files have not been modified or altered, both hashes should be checked to prove they are equal.

Recovering deleted files is an essential part of the forensics analysis. This is called normally **file carving**. Once the binary copy is done, it can be used to recover files that has been deleted from the studied device. There are different tools to perform this part of the forensics analysis, one of the most used is the Linux tool “PhotoRec”.

PhotoRec is an effective file recovery program, which enables you to recover various file types, including multimedia, documents, archives and much more from a range of hard storage devices (hard disks, CD-ROMs, USB, memory cards etc.).

One of the tools that is sometimes used by forensics are those that allow the forensic engineer to get the **timeline** of the files in a specific directory. It consists on the creation of a document which lists the names of all the files located inside a certain directory and some useful information that can be used to filter the documents trying to encounter evidences. This information may include: size, format of the file, day of creation, day of the last modification, user that created the document and user which did the last modification.

1.4 Outline

The thesis can be divided into two different parts. Part 1 is composed of chapter 2 , which is an explanation of how the Blockchain works, with specific detail in Bitcoin. Also we will explain different attacks and de-anonymization techniques that can be conducted in the Bitcoin P2P network, we will then proof that by doing a Man in the Middle attack, a user can be identified in the network. Part 2 consists of chapter 3 and also gives a brief explanation of how TOR works. Different attacks and de-anonymization techniques will be identified, and we will also proof how to protect regular P2P network attacks by adding layers of encryption (Onion Routing). The third part is composed by Chapter 4, where we will explain which information is relevant and how to extract this information from a target's PC. Which will then be used to incriminate cybercriminals.

- In Chapter 2, We will also study and expose vulnerabilities and flaws that can be found in the Bitcoin network. We will also study and expose vulnerabilities and flaws that can be found in TOR's hidden services and network. The main objective is to try to discover the targets Bitcoin address and its IP address. We will also propose some requirements to perform the attacks identified.

- In Chapter 3, we give more insight in how to proceed with a Forensic investigation in a Bitcoin related scenario, once the user has been identified. We will assume the file is encrypted, and therefore, we will have to decrypt this file. This chapter will be a Proof of concept of the previous points, where we will extract the relevant information in the targets PC to perform a forensics study.

- 4 In Chapter 4, we will present the conclusions of the study as well as point out further study trends, where the paper could be developed.

1.5 Ethical Considerations

The different attacks explained to de-anonymize Bitcoin clients described in this Project are proposed in order to be carried out on live systems and explained as such, and are meant to successfully de-anonymize clients on real live systems. Consequently, this attacks are explained as if they were performed in such environments.

In the case of the attacks described to successfully de-anonymize users in real networks, those were only a theoretical proposition and therefore were never taken into practice. It was never part of the main scope of the project.

The proof-of-concept experiment carried out is intended to fully de-anonymize a real client. For ethical reasons no user outside our control was targeted or victim in any way, the experiments were targeted to myself.
All the wallets that were used to test the Forensic tool were of my own, and the brainwallet Bitcoin addresses created were double checked in order to reassure that they did not have a rightful owner or funds in them.

Chapter 2

DE-ANONYMIZATION OF USERS

The first part in the investigation will be to try and identify the criminal. As we know two possible networks where we could get a glimpse of the targets IP address, we will identify different attacks that may help us achieve this goal. Bitcoin network has a typical P2P structure, some of the attacks performed in a P2P network can be performed here too. TOR network is also based in a decentralized P2P network, however, the layer routing mechanism will require new methods to attack it successfully.

2.1 De-anonymization of clients in Bitcoin Network

Anonymity in the Bitcoin system is based on the following facts:

- Bitcoin address cannot be mapped to the real identity.
- Bitcoin transactions don't contain any personal information.
- The new transactions are spread radially, thus the sender's IP address will not be exposed.

However, Bitcoin is not fully anonymous. We can find more than one weakness that can help both criminals or special forces agents identify real users behind Bitcoin addresses.[14]

- The real-name authentication mechanism (exchange houses are the most common platform) helps Bitcoin service providers to find the addresses that ever deposited and withdrew.
- Bitcoin address exposed on the internet can be related to its owner (forums were people post their Bitcoin addresses to receive donations).
- The chain of transactions is transparent and traceable (https://www.blockchain.com/explorer).
- Gathering some or all inputs when sending Bitcoins to others, which may expose other addresses of the sender (Clusters or transaction graphs).

2.1.1 Tracing transactions history

Tracing the history of transactions can be used to identify the address. It should be remembered that Bitcoin is not completely anonymous means of payment. This is a very generalist theory, as we will see that by using advanced software we can develop tools able to perform this to a high degree of detail.

2.1.2 Sybil attack

The Sybil attack is an attack where the reputation system is corrupted or sabotaged by forging identities in P2P networks. The attacker will try to fill the network with nodes under his control, then, any new user that tries to connect to the Bitcoin network will connect to ones owned by the attacker. If this situation was to happen, other users will only be able to connect to an alternate network where newly created blocks will specifically by used for fraud or other illegal purposes. How can this happen:

- The attacker blocks transactions from other users, disconnecting you from the public network.
- The attacker only connects you to the blocks he created in a separate network.
- The attacker can see all of your transactions through the use of special programs.

By the nature of this attack, in order to perform it successfully, the amount of resources required is limited to very few. Therefore it can be estimated that governments or very large organizations are the only capable of performing this attacks. If any attacker was to perform a successful attack on the network, it would have the possibility to identify the IP address of every targeted node in the network, which may lead to the de-anonimzation of the addresses of Bitcoin users[1].

2.1.3 Through clusters and transaction graphs

The main idea of clusters and transaction graphs is to link addresses to a specific user. Even though anonymity is still present, it increases the amount of possible addresses for each different address found for a specific user. Thus, if a Bitcoin address happened to be de-anonymized, it could be checked with the cluster database for fast correlation. This is a very powerful tool proposed by Meiklejohn [1], in which users can be associated to one or more addresses by means of two heuristic models.

[1]By identifying and excluding Bitcoin nodes that belong to exchange, online wallet and other online websites where more than one address may use that Bitcoin network node, the attacker can then isolate nodes that belong to private users.

1st Heuristic model [10]: When a Bitcoin user receives or spend funds, the wallet uses or creates new addresses for each transaction. Remaining funds are kept in those addresses, therefore, if we see a transaction spends coins originating from multiple inputs, we can assume that the transaction was executed by one user. With this, we can assume that the user owns all the addresses related to the transaction inputs, as the transaction had to be signed using the private keys that matches all the public keys of all inputs.

E.g. If Alice sends money to Bob through addresses A and C, and later sends money to Charlie through addresses C and D, we can cluster addresses A, C and D under Alice's pseudonym.

2nd Heuristic model: Whenever a transaction has a single output, the output address is usually controlled by the same entity that owns the input address. The reasoning here is that the likelihood of a user owning the exact amount of bitcoins in a single UTXO[2] that wants to send to another address is extremely low. Typically when a transaction is made, a change address is created to return the remaining bitcoins to the sender. It is extremely hard to identify change addresses with high certainty, but if there is no change address we can suppose that the money is just being transferred between addresses owned the same user (or distinct UTXOs are being consolidated into a single larger UTXO).

[2]An unspent transaction output (UTXO) is an output of a blockchain transaction that has not been spent, generally used as an input in a new transaction.

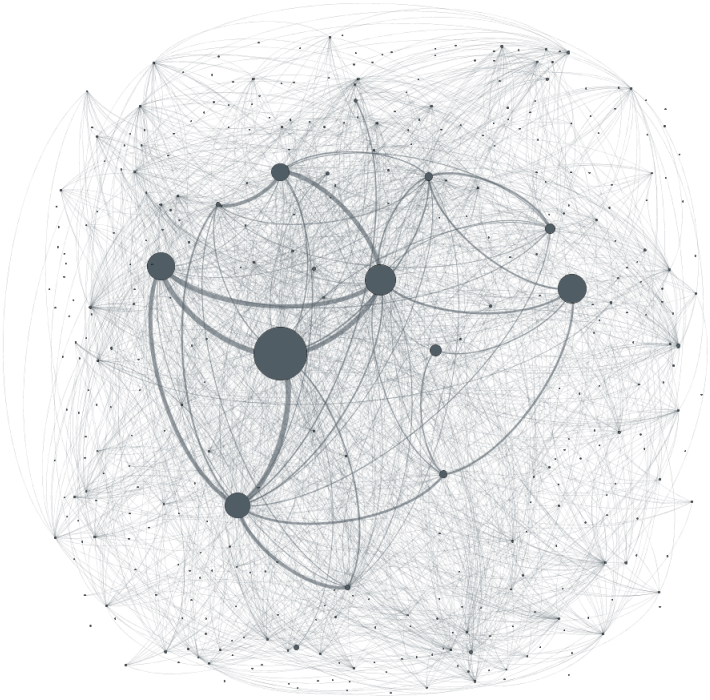

FIGURE 2.1. Transition graph or cluster address grouping

2.1.4 Through coin-mixing services

A cryptocurrency tumbler (commonly known as coin-mixer) is a service used to obfuscate funds in their current address by sending them to a new address. This process is achieved by the use of an intermediate entity, who mixes cryptocurrencies funds with others, from other users, so it is harder to trace them back to the initial user.

E.g. (For simplicity, each user only has one input) If Alice, Bob and Charlie (with respective addresses A, B and C) send their desired amount of Bitcoins that they want to obfuscate, to T (tumbler, address T). The Tumbler then sends the input amounts minus a transaction fee (kept by the tumbler as profit) to an account or accounts specified by the users, as outputs. Alice will then receive Bitcoins to addresses E and F, Bob will receive to address G and Charlie will receive to addresses H and I. Hence, the larger the amount of inputs and outputs involved in this process, the harder it is to trace coin-mixing service users.

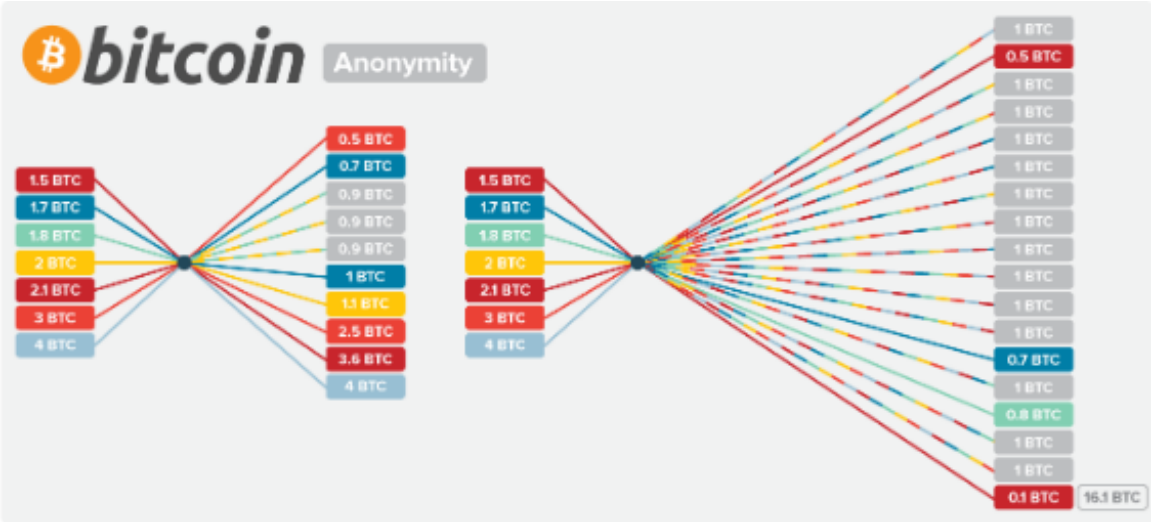

FIGURE 2.2. Bitcoin-mixing

We can try to isolate users mathematically, by calculating the mutual information of outgoing and incoming transactions. Let X be the discrete random variable with probability mass function Pi = Pr(X = i), where i represents each possible value that X may take. In this case, each i corresponds to an element of the anonymity set (a sender). We denote by H(X) the entropy of the system after the attack has taken place. For each sender belonging to the senders set of size N, the attacker assigns a probability pi . H(X) can be calculated as:

$$H(X) = -XN \quad i = 1 \quad Pi * log_2(Pi)$$

Let HM be the maximum entropy of the system we want to measure, for the actual size of the anonymity set: HM = $\log_2$ (N) , where N is the number of honest senders (size of the anonymity set). The information learned by the attacker with that specific attack can be expressed as HM - H(X). We divide by HM to normalize the value. We then define the anonymous mutual information provided by the system as:

$$I = 1 - HM - H(X)HM = H(X)HM$$

2.2 De-anonymization of clients in TOR Network

It is important to highlight that TOR is not the main point of study of this project, therefore it will only be explained from a theoretical point of view. Although it is not the focus, some basis will be left for future studies, in case this project wants to be

expanded from this perspective.

There is a large set of precautionary measures and best practices to make web browsing safer and more secure for users. Let's say that, for example, we send an HTTPs request to a server and someone intercepts that request. That person can not know what that message says, since it has been encrypted. We will assume that the level of security it is still not enough and want to take this to the next level, i.e. you don't even want anyone sniffing on your network, to know which server we are contacting to and if we are making any requests or not. This is where The Onion Router comes into play.

TOR Network has proven to successfully obfuscate its users real identity by adding extra layers of security. However, like most of things, it has been proven that it's not perfect, and that users can still be de-anonymized. Here we will present some of the most best known techniques or attacks that can be carried out in the TOR network, in order to exploit its vulnerabilities. Most of this attacks require the control of a large portion of its relays, which makes this kind of attacks improbable or unfeasible to perform (due to the high amount of resources required, unobtainable to many organizations).

By increasing the routing layers in the intermediate relays, some of the following attacks could be avoided (As I2P has implemented in its application). However this will have repercussions in the total communication delay between the user and the end point.

According to existing de-anonymizing techniques on the TOR network, we can sort these techniques into two groups from two different perspectives

- **Passive and active attacks** - The adversary can passively observe the network's traffic or actively manipulate traffic.
- **Single-end and end-to-end attacks** - The attacker can impose the network's anonymity by monitoring or controlling TOR circuits at either the enter relay or exit relay side, or at both edges of the circuit.

Based on their method and goal, attacks can be categorized into seven groups:

- Correlation Attacks (End-to-end, Passive Attack)
- Timing Attacks (End-to-end, Active Attack)
- Supportive Attacks (Not classified)
- FBI NIT exploit (Not classified)
- Differential Attack (Single-end, Passive Attack)

2.2.1 Correlation Attacks

Correlation attacks are well-known de-anonymization attacks. In this category of attacks, it is assumed that the attacker controls both the entry node and the exit node of the circuit. Traffic correlation attacks can be easily done by eavesdropping the outgoing and incoming traffic between the victim client and the first relay node (entry guard), as well as traffic reaching the final destination (hidden service, exit relay node. etc). Once the traffic is being monitored, statistical analysis is used to determine that they belong to the same circuit.

As such, TOR does not promote absolute anonymity. The user's address as well as the destination address of the monitored traffic are obtained by the attacker, who can successfully de-anonymize the target via correlation attacks. Although the attacker doesn't necessarily need to take control of either the guard and exit relays, it just needs to be able to operate the traffic running through those two nodes. For this attack, we assume that the attacker controls one or more very fast exit routers which see a significant fraction of the traffic exiting the TOR network, thus it gets access to pseudonyms of the users (ex. cookies, logins).

"The way we generally explain it is that TOR tries to protect against traffic analysis, where an attacker tries to learn whom to investigate, but TOR can't protect against traffic confirmation (also known as end-to-end correlation), where an attacker tries to confirm a hypothesis by monitoring the right locations in the network and then doing the math. And the math is really effective. There are simple packet counting attacks (Passive Attack Analysis for Connection-Based Anonymity Systems) and moving window averages (Timing Attacks in Low-Latency Mix-Based Systems), but the more recent stuff is downright scary, like Steven Murdoch's PET 2007 paper about achieving high confidence in a correlation attack despite seeing only 1 in 2000 packets on each side (Sampled Traffic Analysis by Internet-Exchange-Level Adversaries)." [13]

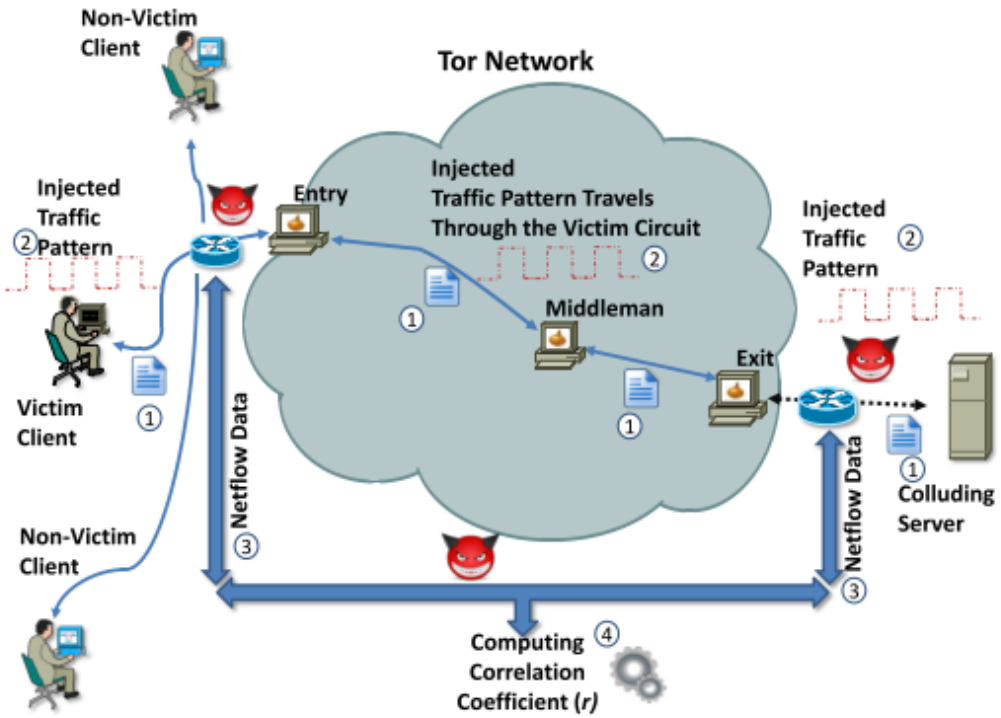

FIGURE 2.3. Traffic & Timing Correlation Attack - https://securityaffairs.co/wordpress/30202/hacking/tor-traffic-analysis-attack.html

At the time (early 2014), TOR relays could easily confirm their suspicion by adding an arbitrary value to the packet and check for it on the other end to reach the level of certainty. This was quickly patched, but correlation attack is still not prevented.

2.2.2 Timing Attacks

Timing attacks are another form of de-anonymizing attacks. During a timing attack an adversary manipulates both the entry and the exit relay of a targeted client. By correlating flow patterns in traffic flowing from the entry node to traffic flowing to the exit node, the adversary can determine which server a client is communicating with.

The aim of the attack is to uncover the autonomous system (AS) which contain the target and possibly also the identity of the target. The target can be either an OP, an OR or a hidden service.

The attack is composed of three steps.

- First, a server is needed that is colluding with the adversary. This server may

be controlled by the adversary (which will make the attack easier to execute), but this is not strictly necessary.

- Second, several network bandwidth probing nodes are required.
- Third, maps containing ingress and egress routers of AS's are needed. [3]

The adversary places the bandwidth probing nodes close to ingress and egress routers at the boundary of the AS's. The most useful location for the probing nodes is determined using the acquired network maps.

The adversary must then get the target client to connect to the colluding server. Once the connection is established, the server will vary the bandwidth of the connection to the client, leaving a distinct pattern along the path of the connection. This is the pattern that will trace back to the OP of the client.

The bandwidth probing nodes at the AS boundaries will probe ingress and egress routers in order to find the bandwidth pattern used by the colluding server. If the pattern is detected on a router, the corresponding AS is probably part of the path from client to server. Using this technique, the adversary can trace the route back to the client.

2.2.3 Supportive Attacks

This attacks do not directly aim to de-anonymize TOR users or disrupt the TOR network but rather are helpful to perform a de-anonymization attack or a disruptive attack at a later point in time.

In the Sybil attack, an adversary must control a major amount of virtual identities (relays) in the network in order to obtain a large influence in the network. The effectiveness of many attacks on TOR depends on the consensus weight of the attacker, which is the amount of traffic an attacker can observe. As the consensus weight grows, a number of other TOR attacks become easier to execute. Examples of attacks that are easier in combination with a Sybil attack are the fingerprinting and correlation attacks.

Besides simplifying other attacks, the Sybil attack poses a major risk on the usage of the TOR network and therefore on the anonymity of its users. The effectiveness of TOR depends on the reliability given to the TOR relays. Less users means a decrease in the overall anonymity of the network. The remaining users will continue using the network with a lower anonymity, presenting better opportunities for attacks. This problem can be exploited by adding malicious relays and strategically affecting the reliability of anonymous communications to increase the odds of an adversary compromising user anonymity.

[3] Ingress traffic is traffic in the TOR network that originates outside of it. Egress traffic originates inside the TOR network and is sent outside the network.

However, since Sybil relays typically behave and appear similarly, there are some heuristics that can be used to detect a Sybil attack to some extends. Relays that are part of a Sybil attack often join and leave the network simultaneously, they have common configuration parameters, and may frequently change their identity fingerprint to manipulate TOR's distributed hash table.

2.2.4 FBI NIT exploit

The NIT (Network Investigative Technique) is a form of malware employed by the FBI since at least 2002. It is a drive-by script (written in JS) which is downloaded by the target (or more than one target) computer. The malware is sent via TOR using Javascrit features, consisting of three components:

- The exploit sent over the TOR browser and network.
- The payload sends back to the attacker personal information regarding the targets IP direction and configuration.
- The server that support the infrastructure which hosts the NIT exploit but modifies each copy sent to include a unique identifier. This way it can be used to mass identify users in the network.

2.2.5 Differential Attack

Consider a user which periodically checks some Web server or a web service that instructs the user's browser to periodically re-establish streams or any web sites with auto-refresh contents. The aim of the attacker is to find at least one of the guard nodes of a pseudonymous user (identified by a cookie or a login credential) that uses such a service for several days. Note that this attack does not require a single long-lived circuit or session. It just requires that a TOR client is connected to the TOR network for non-negligible amount of time within the span of a month (as long as the guards are still valid).[3]

In this attack, the attacker has control over a significant number of exit relays in the TOR network. The reason behind only controlling exit nodes is that these relays are the ones that can sniff traffic between the server and the sender. Assuming that a user visits a Web server that causes recurrent connections to occur. One of the specifications in TOR, is that connections last 10 minutes, therefore after this time, the browser creates a new one for the user. Ten minutes after the first connection, the users TOR client will have to build a new circuit. Given a sufficient number of exit nodes controlled by the attacker, the attacker can determine the type of user behind each connection.

2.3 De-anonymization requirements

Due to a lack of time and resources, none of the attacks explained above were able to be performed to proof them. As explained, in most of the attacks, for the attacker to successfully de-anonymize a user, he must control a significant amount of the network nodes. We will calculate the minimum requirements a user or organization should require in order to try and achieve a positive attack.

For both networks, we will consider that being in control of at least 10% of the network's nodes, will be enough to identify the targeted user.

Regarding Bitcoin P2P network, according to https://bitcoinist.com/bitcoin -nodes-10k-reachable/ the amount of bitcoin nodes in the network is 64768. Therefore, we would require to control at least 6477 Bitcoin nodes.

On the other hand, according to `https://metrics.torproject.org/rs.html#aggregate/all` the amount of reachable TOR relays in the network is 6628. If we were to perform the attack in the TOR network, we would require to control 663 relays.

2.3.1 Budget requirement

As mentioned before, the set up we would build up, would try and require the minimum specifications. For it, we have decided to propose two possible combinations. The first one would be to use Raspberry Pi's Model B, while the second one would require a Desktop PC. The only specification that has been taken into consideration is the Hard Drive space. The actual Bitcoin Blockchain occupies around 250GB, so we require a 500GB or above microSD for the Raspberry and at least a 500GB HDD for the desktop PC.

The following table has calculated the required budget to build both set ups:

BUDGET		**Raspberry Pi price**	**PC price**
BITCOIN NODES	10%	€66,00	€302,48
64768	6477	€427.468,80	€1.959.102,46
		Raspberry Pi price	**PC price**
TOR RELAYS	10%	€66,00	€302,48
6628	663	€43.758,00	€200.544,24

TABLE 2.1. Budget for a 10% network control.

We have disregarded any concern for the RAM memory necessary to run any desired script or application that may analyze the network (such as Wireshark).

2.3.2 Time requirement

We will use KALI LINUX as the preferred OS for all the machines, both Raspberrys and PCs. The average time required to install KALI in a machine is around 30 minutes.

As said before, the actual size of the Bitcoin Blockchain is around 250GB. Assuming a download speed of 10MB/s, the time required to fully download the Blockchain in a machine would take be 7 hours (in a normal case scenario (such as the set up used for the proof of concept), the download speed varies and the total time required drops to 9 hours). To optimize the installation time, whilst one machine is downloading the blockchain, KALI will be installed in the other machines, such as in an assembly line.

Therefore, the required time to fully complete the set up installation will be the following:

$$Bitcoin : 0,5h + (7 * 6477)h = 45338,5h$$

This means that a single user would spend 1890 days, without stopping, to complete the configuration process (A unique internet connection to download all the Blockchains). We can not download multiple Blockchains simultaneously as this will affect the overall download speed.

For the TOR de-anonymization attack, the time required is considerably shorter. The average time required to configure a TOR relay in a Raspberry Pi, once the OS has been installed, takes around 45 minutes. According to TOR requirements, a relays must be active for 72 hours before it comes online and accessible by users. To optimize the installation time, whilst one machine is in the 3 day acceptance period, other machines will be configured.

Therefore, the total time required for this set up will be the following:

$$TOR : (1,25 * 663)h + 72h = 900,75h$$

This means that a single user would spend 38 days, without stopping, to complete the configuration process.

A very important part that has not been calculated is the human power required. A single person cannot monitor all the incoming and outgoing traffic through all the nodes unless it has a very good script configured, that analyzes the network waiting for a coincidence to appear.

We have also disregarded the electricity consumption required to power up all the nodes 24/7, as well as the maintenance needed to keep all the nodes up and active or the internet connection required.

This process can also be done by other simple network attacks, however most of them require very specific situations. Probably the most effective attack would be a

Man In The Middle. In an eavesdropping attack, the attacker would secretly relay the connection between the target and the network node. The attacker would decide to monitor the communication in search for any particular piece of information, or alter it. In this particular case, an attacker could link a Bitcoin user to a specific Bitcoin address, or he could alter the information in the message to send the Satoshis to a different address.

In a real case scenario, this attack would imply that the attacker might already have some kind of knowledge of who the criminal might be, as the Man In The Middle attack requires both entities to exist in the same network simultaneously.

Chapter 3

BITCOIN FORENSICS EXAMINATION

Due to the time and economic requirements calculated to identify a user in both the Bitcoin and TOR network, we will suppose a theoretical scenario where we do control such set up. In this scenario where we control 952 Bitcoin nodes, we have decided to perform a sybil attack. We have been monitoring the network until we detected that the criminals Bitcoin address (the criminals Bitcoin address that was seen buying guns in the Black Market is the following: 34ZUdU5mWqT4frvQC9YMebNhrS8iVYg2Xo) has connected to one of our nodes. We will make him download in the background a script that will send us its IP address (similar to the NIT exploit attack in TOR network).

We found out that the address 34ZUdU5mWqT4frvQC9YMebNhrS8iVYg2Xo was linked to the following IP address: 2.152.6X.XXX. The job now is to investigate if the Bitcoin address that we intercepted, corresponds to one of the addresses the target has under his possession. We have joined with a Security organization to enter the targets house and obtain any possible evidence that can incriminate him (link the user to the Bitcoin address intercepted).

In order to be able to present valid evidences or proofs before criminal or civil courts, it is important to take into consideration the processes involved in Digital Forensics.

3.1 Forensic Procedures

3.1.1 Investigation Initiation

For businesses of any size, it is important for the business to secure the data for forensic analysis. Prior to a new forensic investigation, it's crucial that the company keeps unaltered all the possible files or systems that may be relevant for the investigation. This way, all the evidences will give a clearer image of "who and what" was involved in the crime.

The most effective methods to ensure legal admissibility while preparing to engage a forensic analyst, which will be carried out in this lab, include the following:

- Drive Imaging
- Hash Values
- Data Recovery

- Computer system history
- Chain of Custody

3.1.2 Drive Imaging

Before investigators can begin analyzing the evidence rescued from a crime scene, it needs to imaged first. Imaging a drive is a forensic process in which an analyst creates a bit-by-bit duplicate[1] of a drive. This forensic image of all digital media helps retain evidence for the investigation. When analyzing the image, it should be kept in mind that even formatted drives can preserve important recoverable data. In the best cases, file carving techniques can recover all the deleted files.

As a rule, investigators should exclusively operate on the forensic image created and never perform forensic analysis on the original media. In fact, once a system has been compromised, it is advisable to do nothing with the system other than isolating it to prevent any incoming or outgoing connections. Limiting actions on the original computer is important, especially if evidence needs to be taken to court. Once finished the evidence manipulation, the forensic investigators should be able to prove that the information handled has not been tampered in any way by presenting cryptographic hash values, digital time stamps or legal procedures followed.

By utilizing the bit-stream image, the Computer Forensics Examiner takes no risk of contaminating the original evidence. It creates a bit-stream image by attaching the original computer media to a write protection device that ensures no writes can take place to the original media while the bit-stream image is created.

3.1.3 Hash Values

When an investigator images a machine for analysis, the process generates cryptographic hash values (In this case, the files will be hashed with MD5). The purpose of a hash value is to verify the authenticity and integrity of the image as an exact duplicate of the original media.

Hash values are crucial, especially when presenting evidence into court. If the information extracted was to be altered even by the smallest bit of data, the checksum would generate a completely new hash value. When a new file is created or an existing file on the computer is edited, it generates a new hash value for that file. After the evidence study has finished, and all the relevant data has been extracted and manipulated, a hash of the image is computed. If the output hash values do not match the expected values, the court may induce that the evidence has been tampered, and therefore the results may have been manipulated.

[1] Bit-by-Bit duplicate (also referred to as mirror image backup) involves the backup of all areas of a computer hard disk drive or another type of storage media (all the metadata is extracted from the target media as it is important to be able to recuperated deleted files). Such backup replicates with exactitude all sectors on a given storage device. Therefore, all files and ambient data are copied.

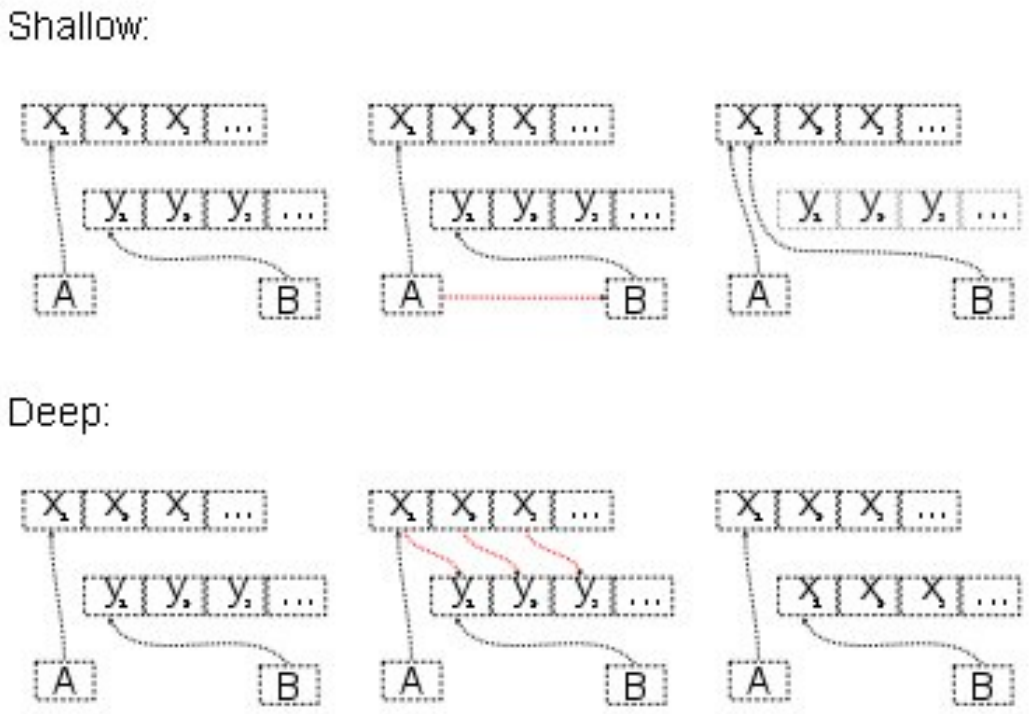

FIGURE 3.1. Bit-by-Bit copy

3.1.4 Data recovery (*File/Data Carving*)

File carving is a process used to recover files metadata or ambient data that are stored in a drive or other data storage device. This method tries to recover files without knowing or having accessible its metadata. It is important to notice that the software used to perform the file carving process, is executed in read-only access to avoid writing over the drive. This way, you can ensure that the image is not being altered. For a safer process, it is advisable to perform this technique in the forensic image created earlier in an investigation.

File carving techniques are most commonly used to recover files from from the unallocated[2] space in a drive. Many file systems do not zero-out the data when they delete files. Therefore, file carving techniques exploit this to analyze the file structure and content of the raw bytes found in the drive. File carving is the process of recon-

[2]Unallocated space refers to the area of the drive which no longer holds any file information, however it potentially contains complete and partial files that can remain untouched for long periods of time. The residual data cannot be viewed by an ordinary computer user, but can be discovered and extracted.

structing files by scanning the raw bytes of the disk and reassembling them. This is usually done by examining the header and footer of a file.

3.1.5 Computer system history

When investigating a crime, it is critical to have an idea of when applications or files were executed. Extracting information from the drives metadata or log files, we can create a timeline of the usage of the drive. This output can help analyze documents preserved from computers, servers, or other executable files existent in the memory image by comparing the date and time they were last accessed or created. This can help correlate any suspicious activity in the drive or system.

In cyberattacks were there has been a breach in a system, it is crucial in the investigation to quickly scan relevant information. Creating a timeline analysis can help identify relevant activity faster.

Timelines are useful to analyze the picture of how a computer was used. By looking at the high-level activity for the past month, it may be able to identify what directories had activity. This helps to determine what user accounts and applications are used. Timelines are also useful to analyze all of the places that had activity in a given time range when intrusion activity occurred (assuming that the file system time stamps were not modified by the attacker) and identify all of the places that the intruder placed files.

3.1.6 Chain of Custody

The chain of custody collects all the relevant information occurred during an investigation. From the media collection, its transfer or its manipulation by the forensic analyst, all the actions regarding all the media found or obtained, this artefact should document all this movements and capture signatures and dates upon media handoff. Once we have created a binary copy of the media, this artefact demonstrates that the image has been under known possession since the time it was created.

Any misalignment or gap during the possession of a record (or media), any situation where the evidence has been in an unsecured location or the loss of an evidence can be problematic (in the case of a loss, this can be disastrous for the investigation). Any lapse in chain of custody nullifies its legal value, and thus the analysis. Investigators may still analyze the information but the results are not likely to hold up in court.

In order to protect the chain of custody and make it as authentic as possible, a series of steps must be taken into consideration during the forensic analysis process. The following established procedure should be followed according to the chain of custody for electronic evidence:

- Save original material: Work always on copies of the digital evidence, by creating a Bit-by-Bit clone of it. This way we create a complete duplicate of the evidence and ensure that the original evidence is preserved and kept unmodified.
- Photos of physical evidence: Photos of physical evidence establish the chain of custody and make it more authentic.
- Take screenshots of digital evidence content: Taking screenshots is an effective way of establishing the chain of custody.
- Document date, time, and any other information of receipt. Recording the timestamps of who has had the evidence allows investigators to build a reliable timeline of where the evidence was prior to being obtained. In the event that there is a hole in the timeline, further investigation may be necessary.
- Perform a hash test analysis. Performing a hash test ensures that the data we obtain from the previous bit-by-bit copy procedure is not corrupt and reflects that the original evidence has not been tampered.

3.2 Tools

- BitcoinCore wallet.[3]
- TFM Alvaro Forensic Tool (**FIGURE 3.2)**
  - MD5 checksum + Bit-to-bit copy.
  - File Carving.
  - Timeline.
  - Extract wallet private key.
  - Password Crack.
  - MD5 Checksum.
- 2 computers with KALI distribution.
- 8GB Toshiba thumb drive. (**EVIDENCE A.7**)
- 500GB Hard Disk Drive. (**EVIDENCE ??**
- 1TB Hard Disk Drive.
- AMD CPU FX-8350.

[3]Bitcoin Core is free and open-source software that serves as a bitcoin node and provides a bitcoin wallet which fully verifies payments. It is considered to be bitcoin's reference implementation and is the most used implementation by a large margin.

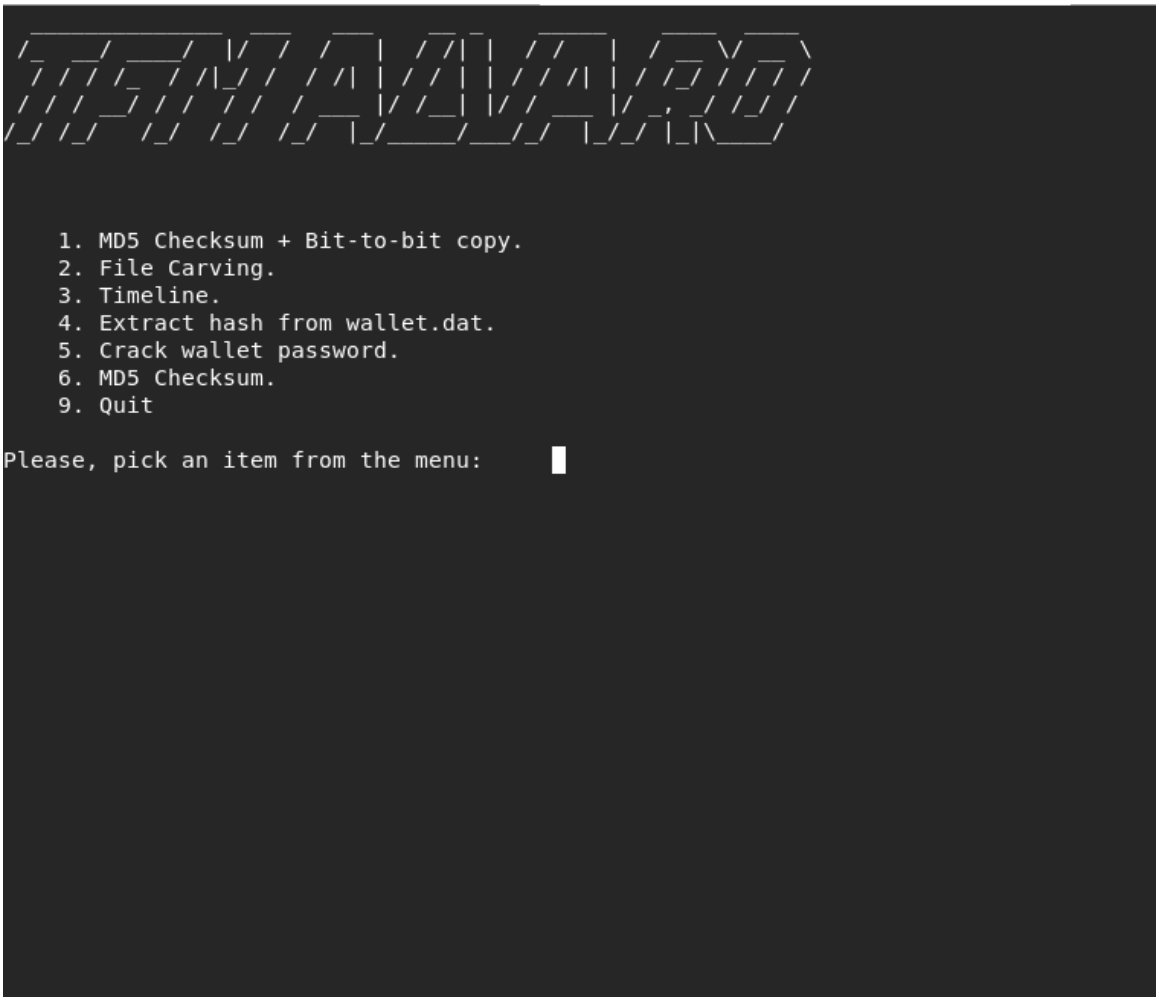

FIGURE 3.2. TFM Tool

3.3 Lab Setup

Remember that during the de-anonymization process, we identified that this address addr=34ZUdU5mWqT4frvQC9YMebNhrS8iVYg2Xo was linked to the following IP address: 2.152.6X.XXX.

Assuming the user works on a Linux distribution, more exactly KALI, we will set up the bitcoin folder where all the relevant data files can be found. The software used to create the Bitcoin wallet will be Bitcoin Core (thick wallet) version 0.17.0.1.

In this case we have created the directory inside the Documents folder: /home/hallvardr/Docum 0.17.1/.

Once we open the folder we can find the following files inside the bitcoin folder:

```
hallvardr@kali:~/Documents/bitcoin-0.17.1$ ls
banlist.dat  bitcoin.conf  debug.log          lib           wallet.dat
bin          bitcoind      dump_setup.txt     mempool.dat   wallet1.dat
bitcoin-cli  blocks        fee_estimates.dat  peers.dat     wallets
bitcoin-qt   chainstate    hash.txt           share
bitcoin-tx   db.log        include            test_bitcoin
hallvardr@kali:~/Documents/bitcoin-0.17.1$
```

FIGURE 3.3. Bitcoin Folder

- banlist.dat
- bitcoind.pid
- chainstate
- blocks - Stores Bitcoin blocks, in network format, dumped to disk raw. They are only needed for re-scanning missing transactions in a wallet, reorganizing to a different part of the chain, and serving the block data to other nodes that are synchronizing.
- fee_estimates.dat - Statistics used to estimate fees and priorities. Saved just before program shutdown, and read in at startup.
- peers.dat - Storage for peer information to make a reconnect easier. This file uses a bitcoin-specific file format, unrelated to any database system.
- wallet.dat
- debug.log - Bitcoin's verbose log file. Automatically trimmed from time to time.

- mempool.dat
- wallets

Our focus in this exercise will be on the wallet.dat, found in /wallets, since this is the file that contains all the private key information stored from the clients Bitcoin wallet. It includes all of the information needed to access a Bitcoin wallet and its associated funds. It is important to notice there might will be other .dat files, as these can be manual backups created by the user.

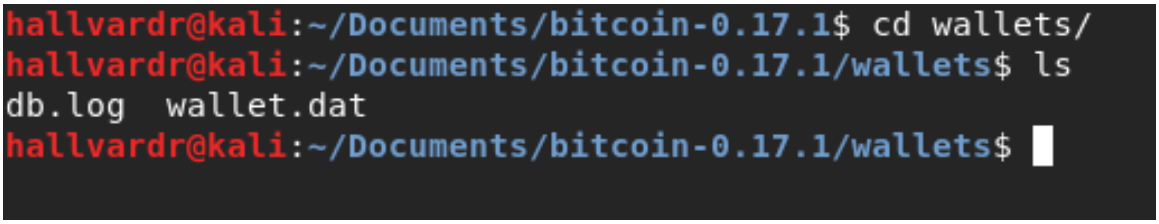

FIGURE 3.4. Wallets

The information that can be found in the wallet.dat file is the following:

- Keypairs for each of your addresses
- Transactions done from/to your addresses
- User preferences
- Default key
- Reserve keys
- Accounts
- Version number
- Key pool

All this information can be identified in **FIGURE 3.6**. As the wallet is new, the addresses have not performed any actions, the transaction done field will not appear or appear as empty. Therefore this field will not be identifiable.

There are four classes of keys:

- Used keys, which have unredeemed transaction outputs linked to them (and thus contain funds).

- Used keys, which only have redeemed transaction outputs linked to them (worthless, but the addresses may be used still by others).
- Unused keys (whose corresponding addresses may have been published).
- Reserve keys (whose corresponding addresses have typically never been divulged, but may have been divulged at the time the export is imported again).

A Bitcoin wallet works with asymmetric or public-key cryptography, which generates the public-key-and-private-key pair required for users to create transactions between them. In this case to transfer Bitcoins between them and register the information related to the transaction in the Blockchain. Both public and private keys, are large integer numbers represented using a separate Wallet Import Format (WIF)[4] consisting of letters and numbers. In theory, each public key should only be used once to preserve the anonymity, as we know that addresses can be used to track down users. While the address is public, the private key should be kept secret to avoid money theft and identity usurpation.

If we do a dump of the wallet.dat file, we can observe the fields identified before:

```
$ bitcoin-cli dumpwallet /media/hallvardr/_backup/TFM/
results/dump.txt
```

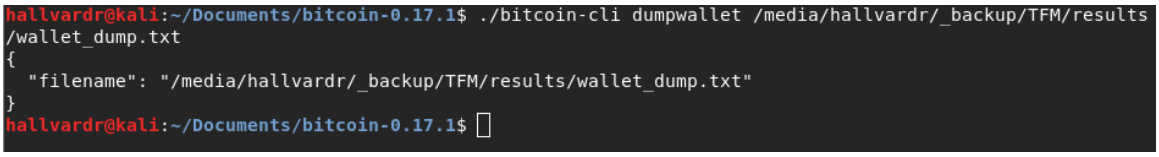

FIGURE 3.5. bitcoin-cli dumpwallet

[4]Wallet Import Format (WIF, also known as Wallet Export Format) is a way of encoding a private ECDSA key so as to make it easier to copy. The procedure for this encoding can be found in:**Appendix B**

```
# Wallet dump created by Bitcoin v0.17.1
# * Created on 2019-05-12T20:16:05Z
# * Best block at time of backup was 551198 (0000000000000000000007631aa59eb1dbb5f0818b017642b1329db2fba564f5d5),
#   mined on 2018-11-23T14:50:48Z

# extended private masterkey: xprv9s21ZrQH143K4Ug68EYjHo9XgtdUxdy7wzqiAvPZQQuHMeBpg3Jt8YgWW324QgR2AFwxMZno61H4GZ257ToKZXewa1kccY6FNSo1BNzkwbY

KwqUZfZbmFZCu38d7icdfZ3b3FpnRh3EZCGcvpAA4rZHib2aRBSg 2019-05-12T09:43:58Z change=1 # addr=3Fn64h9MYXRqCxov2k7wRZcEsm6vpoRvNt hdkeypath=m/0'/0'/130'
Kz23Lboqx4aDjqgF1Tq7pnx7MWgF33FyFqY6SWkkNyH5BqkntVDZ 2019-05-12T09:43:58Z change=1 # addr=35PJLt2oQHkXpUJCUh8FcxQdEWose8ZnRj hdkeypath=m/0'/0'/445'
L3qnnDNp6k47zcjhgspunbRDnp9MF9yqo1SjknZ61Y8uQCjqpLjW 2019-05-12T09:43:58Z change=1 # addr=36XggKBtpsCuQoeQdRxNzKh9J8n5toG92j hdkeypath=m/0'/0'/487'
L2U5Gt6C2qXgVcRBNKjYs1KvSjf1yL3dHsuznpdpLCTN3dpxkujx 2019-05-12T09:43:58Z change=1 # addr=3NhkzppL9cUUBq4sorMzvpN5HFzjoPMV3x hdkeypath=m/0'/0'/561'
Ky1Lf3iJphMreov3p85AGdGuiUuAnTgZ9tix7U56qWiBM4DyMLt1 2019-05-12T09:43:58Z change=1 # addr=37wt9CA5SDtkeGPPeYKZ3YZ2VmnNqfFnLu hdkeypath=m/0'/0'/392'
L1xPiLAtaJGNFPe9xwrRE2baMCH8DRrikLqSXeLfGqPCugXemGLg 2019-05-12T09:43:58Z change=1 # addr=34ZUdU5mWqT4frvQC9YMebNhrS8iVYg2Xo hdkeypath=m/0'/0'/102'
Kzpdec26tPWqeTPCKEwNKokYJ8Q2DP79epkjzxVahATLAM3VyBdE 2019-05-12T09:43:58Z change=1 # addr=3CkLnZ6iK6jaqqL3K9Z6n7yQwqgZ9Cm2Dz hdkeypath=m/0'/0'/142'
L3siGVv9XmJ2MkdWDauaP5Hn7sXL4iRQfrrsYkU71up4YbSp8Lpx 2019-05-12T09:43:58Z change=1 # addr=34SySg6rmpS4X2ywQPESbCF5KcPVH6U1dF hdkeypath=m/0'/0'/494'
L3AqEm2eLARb6hKHoERzxhzPAoXpYrJQuuLD15xr5uxW3whEvLZa 2019-05-12T09:43:58Z change=1 # addr=3JRAETw9jqqLbiAKt44tdcncF6QGmTCjfG hdkeypath=m/0'/0'/558'
KzRw87UyVXYfvBevgQWHfR8FW3QjqmQJR8Byz86U9KdFigKbnsXj 2019-05-12T09:43:58Z change=1 # addr=3L7NEWxGzmZwi5zSV9sKTGxwHtwVYmL1Ux hdkeypath=m/0'/0'/94'
Kyc9u1Z1pHfLet6H3KoQEEpLH3kZSLYzvGNXxZrHyK3RLutoA5tk 2019-05-12T09:43:58Z change=1 # addr=32tr76CryocQ8h2kJMqkLfTDkmDi3LHPg8 hdkeypath=m/0'/0'/331'
Kwo6aKCo9sCfyXwi4o7Akd9rjSdpVZ4c6zvENt8oFT7VVHLbRQky 2019-05-12T09:43:58Z change=1 # addr=3F6XmwxpQJLUMpQYsAKtXbihsCu2LXpNCP hdkeypath=m/0'/0'/293'
L3La4dahztauXqobWdaWzKbeREEUyMSn4fSVM7y4LkPhpXoSLwyM 2019-05-12T09:43:58Z change=1 # addr=3FBrynsJdAJX6cmfVKGQXCF7yrUrnnsRHa hdkeypath=m/0'/0'/125'
L4VWRPabWfwTCDCwLgpEgpq7Ze4HvvxdJ6jnxE2Ur4eKB8gdz1co 2019-05-12T09:43:58Z change=1 # addr=33Urj71FbsHtCDoB5LLo8ky6QoKzTwcSQK hdkeypath=m/0'/0'/131'
L1PyPKCP7Hnq9JBcUb1RiRSLS7dnMYdoH1K4TdMSgfexiSDXhmJY 2019-05-12T09:43:58Z change=1 # addr=3GjsSgGPKQVUbo6ndGvG2bdL6uktaqxcxB hdkeypath=m/0'/0'/182'
L5n1bzyNvwc5DKmSzd5GuauKRRRHZSzpgNT8natgkjaMFudrYmo8 2019-05-12T09:43:58Z change=1 # addr=38QQT2QATCcdwtdA6TVHbX3oFme65dKNsn hdkeypath=m/0'/0'/248'
KyLbczyYV45qB22wnqtojcqqmftaAXjjq8i4PHUr79V97AR73pvi 2019-05-12T09:43:58Z change=1 # addr=3L5776dSjsD3vkpVB4P7zJy7smPW93j4Kx hdkeypath=m/0'/0'/337'
Kyr4iLy5ZzSE6raTrG533ZMeWjcL3fQ4Tj6vi2HbKJEGrv5s3Eom 2019-05-12T09:43:58Z change=1 # addr=375rmNLrbRoMLZdyWao2dNRyYwmiDzaz64 hdkeypath=m/0'/0'/54'
KzSa3PJw9PdBrXAu1TFntZYc4cQS5uXF33pLkMGBqEjAVXGwcaSK 2019-05-12T09:43:58Z change=1 # addr=3LsNA3FVVjv4X5BMcSPjb42QJy36qyDByg hdkeypath=m/0'/0'/290'
L3t1sD4xA38G7H3n67LkmrhKcvArxUrSPCr74u1N3jApmmM6WotF 2019-05-12T09:43:58Z change=1 # addr=3AkLBYHfPKLy7cvgPytUQr1EmKy5spcYeY hdkeypath=m/0'/0'/275'
Kyct6e1k7tUEUwEm1J8EsaKvFTYVxwoUvZ62V76QfS7pXkGu52yp 2019-05-12T09:43:58Z change=1 # addr=34DnvjgxvCsJvxe4xsPMXFx2bc5so13bQn hdkeypath=m/0'/0'/496'
KztvMuyvCWzGFdS3rtpygDSKVMXKzHCNuE6VncEiDTXU311mns9v 2019-05-12T09:43:58Z change=1 # addr=3BdBkYv5oTdW4MhGD1c8FdkqEUDuFPnGWC hdkeypath=m/0'/0'/171'
KxzRB2nYhvQfjY6zNJMiSL2f5h2hyk5DUUxrRnTR9uXpx7Ft71WX 2019-05-12T09:43:58Z change=1 # addr=3DoxCEoWBNKiZ8Lxr1QvZ3qvU8dxuXNc8M hdkeypath=m/0'/0'/202'
Kx7uJvN6QuSzk7JDsdwnsFTKGmUiCc47yTiu5qLXBcWMrUL9osGJ 2019-05-12T09:43:58Z change=1 # addr=34riK99MMZv9psggoFzwdkwUgZ55sRjucW hdkeypath=m/0'/0'/235'
```

FIGURE 3.6. bitcoin-cli dumpwallet

We can identify all the fields indicated before.

For the complexity of the exercise we will also assume that the wallet.dat file has been encrypted by the user. The wallet.dat file will be encrypted using the following command:

```
$ bitcoin-cli encryptwallet popatop
```

```
hallvardr@kali:~/Documents/bitcoin-0.17.1$ ./bitcoin-cli encryptwallet popatop
wallet encrypted; Bitcoin server stopping, restart to run with encrypted wallet. The keypool has been fl
ushed and a new HD seed was generated (if you are using HD). You need to make a new backup.
hallvardr@kali:~/Documents/bitcoin-0.17.1$ sudo apt install gnome-screenshot
```

FIGURE 3.7. bitcoin-cli encryptwallet

Where popatop will be the password with which we will encrypt the wallet.dat file.

Wallet encryption uses AES-256-CBC to encrypt the private keys that are held in a wallet. The keys are encrypted with a master key which is entirely random. This master key is then encrypted with AES-256-CBC with a key derived from the passphrase using SHA-512. Although the underlying code supports multiple encrypted copies of the same master key (and thus multiple passphrases) the client does not yet have a method to add additional passphrases. Therefore once discovered the wallet's passphrase, it will be valid for all the possible master keys. - Bitcoin Wiki - `https://en.bitcoin.it/wiki/Wallet_encryption`

If we try to open the file again, we will see the wallet.dat file is now encrypted and therefore we can not access it without prior decryption.

```
hallvardr@kali:~/Documents/bitcoin-0.17.1$ ./bitcoin-cli dumpwallet /media/hallvardr/_backup/TFM/results
/wallet_dump.txt
error code: -13
error message:
Error: Please enter the wallet passphrase with walletpassphrase first.
hallvardr@kali:~/Documents/bitcoin-0.17.1$
```

FIGURE 3.8. bitcoin-cli decryptwallet

At runtime, the client loads the wallet as it normally would, however the keystore stores the keys in encrypted form. When the passphrase is required, it must first be entered with the walletpassphrase RPC command. This will change the wallet to "unlocked" state where the unencrypted master key is stored in memory.

This "unlock" state will temporarily decrypt the wallet, the time the wallet will be accessible is determined by the value set in the RPC command (In this case, this value will be set to 120 seconds). Once the password has been validated, we will perform the same action as before to dump the wallet file.

```
$ bitcoin-cli walletpassphrase popatop 120
$ bitcoin-cli dumpwallet /media/hallvardr/_backup/TFM/
results/wallet_dump.txt
```

```
hallvardr@kali:~/Documents/bitcoin-0.17.1$ ./bitcoin-cli walletpassphrase popatop 120
hallvardr@kali:~/Documents/bitcoin-0.17.1$ ./bitcoin-cli dumpwallet /media/hallvardr/_backup/TFM/results
/wallet_dump.txt
{
  "filename": "/media/hallvardr/_backup/TFM/results/wallet_dump.txt"
}
hallvardr@kali:~/Documents/bitcoin-0.17.1$
```

FIGURE 3.9. bitcoin-cli dumpwallet encrypted

After unlocking the wallet, we will be able to dump the information contained in it again (**FIGURE 3.10**). This will prove the validity of the password, and will also allow us to investigate the wallet. Investigating the wallet will be the last part of the proof-of-concept, since it contains the information needed to prove a criminal guilty before a court.

```
# Wallet dump created by Bitcoin v0.17.1
# * Created on 2019-05-12T20:16:05Z
# * Best block at time of backup was 551198 (00000000000000000007631aa59eb1dbb5f0818b017642b1329db2fba564f5d5),
#   mined on 2018-11-23T14:50:48Z

# extended private masterkey: xprv9s21ZrQH143K4Ug68EYjHo9XgtdUxdy7wzqiAvPZQQuHMeBpg3Jt8YgWW324QgR2AFwxMZno61H4GZ257ToKZXewa1kccY6FNSo1BNzkwbY

KwqUZfZbmFZCu38d7icdfZ3b3FpnRh3EZCGcvpAA4rZHib2aRBSg 2019-05-12T09:43:58Z change=1 # addr=3Fn64h9MYXRqCxov2k7wRZcEsm6vpoRvNt hdkeypath=m/0'/0'/130'
Kz23Lboqx4aDjqgF1Tq7pnx7MWgF33FyFqY6SWkkNyH5BqkntVDZ 2019-05-12T09:43:58Z change=1 # addr=35PJLt2oQHkXpUJCUh8FcxQdEWose8ZnRj hdkeypath=m/0'/0'/445'
L3qnnDNp6k47zcjhgspunbRDnp9MF9yqo1SjknZ61Y8uQCjqpLjW 2019-05-12T09:43:58Z change=1 # addr=36XggKBtpsCuQoeQdRxNzKh9J8n5toG92j hdkeypath=m/0'/0'/487'
L2U5Gt6C2qXgVcRBNKjYs1KvSjf1yL3dHsuznpdpLCTN3dpxkujx 2019-05-12T09:43:58Z change=1 # addr=3NhkzppL9cUUBq4sorMzvpN5HFzjoPMV3x hdkeypath=m/0'/0'/561'
Ky1Lf3iJphMreov3p85AGdGuiUuAnTgZ9tix7U56qWiBM4DyMLt1 2019-05-12T09:43:58Z change=1 # addr=37wt9CA5SDtkeGPPeYKZ3YZ2VmnNqfFnLu hdkeypath=m/0'/0'/392'
L1xPiLAtaJGNFPe9xwrRE2baMCH8DRrikLqSXeLfGqPCugXemGLg 2019-05-12T09:43:58Z change=1 # addr=34ZUdU5mWqT4frvQC9YMebNhrS8iVYg2Xo hdkeypath=m/0'/0'/102'
Kzpdec26tPWqeTPCKEwNKokYJ8Q2DP79epkjzxVahATLAM3VyBdE 2019-05-12T09:43:58Z change=1 # addr=3CkLnZ6iK6jaqqL3K9Z6n7yQwqgZ9Cm2Dz hdkeypath=m/0'/0'/142'
L3siGVv9XmJ2MkdWDauaP5Hn7sXL4iRQfrrsYkU71up4YbSp8Lpx 2019-05-12T09:43:58Z change=1 # addr=34SySg6rmpS4X2ywQPESbCF5KcPVH6U1dF hdkeypath=m/0'/0'/494'
L3AqEm2eLARb6hKHoERzxhzPAoXpYrJQuuLD15xr5uxW3whEvLZa 2019-05-12T09:43:58Z change=1 # addr=3JRAETw9jqqLbiAKt44tdcncF6QGmTCjfG hdkeypath=m/0'/0'/558'
KzRw87UyVXYfvBevgQWHfR8FW3QjqmQJR8Byz86U9KdFigKbnsXj 2019-05-12T09:43:58Z change=1 # addr=3L7NEWxGzmZwi5zSV9sKTGxwHtwVYmL1Ux hdkeypath=m/0'/0'/94'
Kyc9u1Z1pHfLet6H3KoQEEpLH3kZSLYzvGNXxZrHyK3RLutoA5tk 2019-05-12T09:43:58Z change=1 # addr=32tr76CryocQ8h2kJMqkLfTDkmDi3LHPg8 hdkeypath=m/0'/0'/331'
Kwo6aKCo9sCfyXwi4o7Akd9rjSdpVZ4c6zvENt8oFT7VVHLbRQky 2019-05-12T09:43:58Z change=1 # addr=3F6XmwxpQJLUMpQYsAKtXbihsCu2LXpNCP hdkeypath=m/0'/0'/293'
L3La4dahztauXqobWdaWzKbeREEUyMSn4fSVM7y4LkPhpXoSLwyM 2019-05-12T09:43:58Z change=1 # addr=3FBrynsJdAJX6cmfVKGQXCF7yrUrnnsRHa hdkeypath=m/0'/0'/125'
L4VWRPabWfwTCDCwLgpEgpq7Ze4HvvxdJ6jnxE2Ur4eKB8gdz1co 2019-05-12T09:43:58Z change=1 # addr=33Urj71FbsHtCDoB5LLo8ky6QoKzTwcSQK hdkeypath=m/0'/0'/131'
L1PyPKCP7Hnq9JBcUb1RiRSLS7dnMYdoH1K4TdMSgfexiSDXhmJY 2019-05-12T09:43:58Z change=1 # addr=3GjsSgGPKQVUbo6ndGvG2bdL6uktaqxcxB hdkeypath=m/0'/0'/182'
L5n1bzyNvwc5DKmSzd5GuauKRRHZSzpgNT8natgkjaMFudrYmo8 2019-05-12T09:43:58Z change=1 # addr=38QQT2QATCcdwtdA6TVHbX3oFme65dKNsn hdkeypath=m/0'/0'/248'
KyLbczyYV45qB22wnqtojcqqmftaAXjjq8i4PHUr79V97AR73pvi 2019-05-12T09:43:58Z change=1 # addr=3L5776dSjsD3vkpVB4P7zJy7smPW93j4Kx hdkeypath=m/0'/0'/337'
Kyr4iLy5ZzSE6raTrG533ZMeWjcL3fQ4Tj6vi2HbKJEGrv5s3Eom 2019-05-12T09:43:58Z change=1 # addr=375rmNLrbRoMLZdyWao2dNRyYwmiDzaz64 hdkeypath=m/0'/0'/54'
KzSa3PJw9PdBrXAu1TFntZYc4cQS5uXF33pLkMGBqEjAVXGwcaSK 2019-05-12T09:43:58Z change=1 # addr=3LsNA3FVVjv4X5BMcSPjb42QJy36qyDByg hdkeypath=m/0'/0'/290'
L3t1sD4xA38G7H3n67LkmrhKcvArxUrSPCr74u1N3jApmmM6WotF 2019-05-12T09:43:58Z change=1 # addr=3AkLBYHfPKLy7cvgPytUQr1EmKy5spcYeY hdkeypath=m/0'/0'/275'
Kyct6e1k7tUEUwEm1J8EsaKvFTYVxwoUvZ62V76QfS7pXkGu52yp 2019-05-12T09:43:58Z change=1 # addr=34DnvjgxvCsJvxe4xsPMXFx2bc5so13bQn hdkeypath=m/0'/0'/496'
KztvMuyvCWzGFdS3rtpygDSKVMXKzHCNuE6VncEiDTXU311mns9v 2019-05-12T09:43:58Z change=1 # addr=3BdBkYv5oTdW4MhGD1c8FdkqEUDuFPnGWC hdkeypath=m/0'/0'/171'
KxzRB2nYhvQfjY6zNJMiSL2f5h2hyk5DUUxrRnTR9uXpx7Ft71WX 2019-05-12T09:43:58Z change=1 # addr=3DoxCEoWBNKiZ8Lxr1QvZ3qvU8dxuXNc8M hdkeypath=m/0'/0'/202'
Kx7uJvN6QuSzk7JDsdwnsFTKGmUiCc47yTiu5qLXBcWMrUL9osGJ 2019-05-12T09:43:58Z change=1 # addr=34riK99MMZv9psggoFzwdkwUgZ55sRjucW hdkeypath=m/0'/0'/235'
```

FIGURE 3.10. bitcoin-cli dumpwallet

3.4 Proof of Concept, Accessing a Bitcoin folder

3.4.1 Thumb Drive Study

By this point the criminal has now been identified and we have gained access to its computer and personal devices. To create the most real case scenario possible, we will follow steps that should be taken in a Forensic Analysis scenario. In order to achieve this, we will have to preserve the Chain of custody and integrity of the files.

We have reached the targets home. We have spotted a large amount of pen drives inside a plastic bag (**EVIDENCE A.1**) in one of the cupboards (**Evidences A.2 and A.6**). In order to preserve the Integrity of the data, we have created a binary copy of each of them. Analyzing the images of the thumb drives we have found that one of them (**Evidence A.7**) seems to contain two files that might be related to Bitcoin wallets (**FIGURE 3.11**). By the name we think that the files are backups from a Bitcoin wallet.

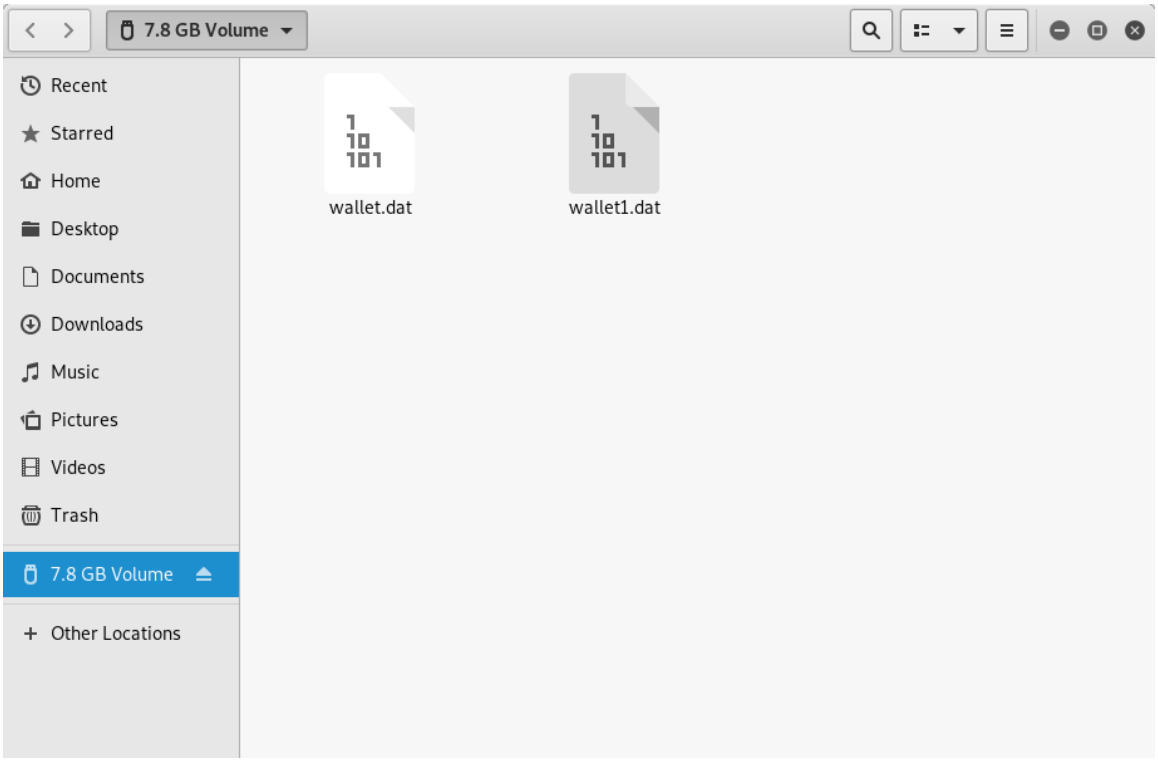

FIGURE 3.11. Pen Drive data files

We will work with this pen drive. First of all, in order to create a binary copy of it, we must look up the name of the drive the thumb drive is connected to.

```
$ sudo fdisk -l
```

```
Disk /dev/sdf: 7.2 GiB, 7759462400 bytes, 15155200 sectors
Disk model: TransMemory
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xd7570d50

Device     Boot Start      End  Sectors  Size Id Type
/dev/sdf1        2048 15155199 15153152  7.2G  b W95 FAT32
hallvardr@kali:~/Documents/TFM/TFM$
```

FIGURE 3.12. Identify Pen drive's mount point

Since we have the pen drive's drive location (**FIGURE 3.12**), we will start the tool created for this laboratory.

```
$ python TFM_Alvaro.py
```

The first option will create a binary copy of the pen drive. This will make a bit-for-bit duplicate of the pen drive in the folder where we will keep all of the results (TFM/results). This image will be the core of the task, since it is the main element of study. We will have to select the input file or drive and select the output where we would like to save the copy of the image. Once the binary copy has been completed successfully, the script will automatically compute the checksum of the file. This checksum which is a MD5 hash of the .dd image, will be necessary to compare it with a checksum done at the end of the study to prove the data inside the image has not been altered.

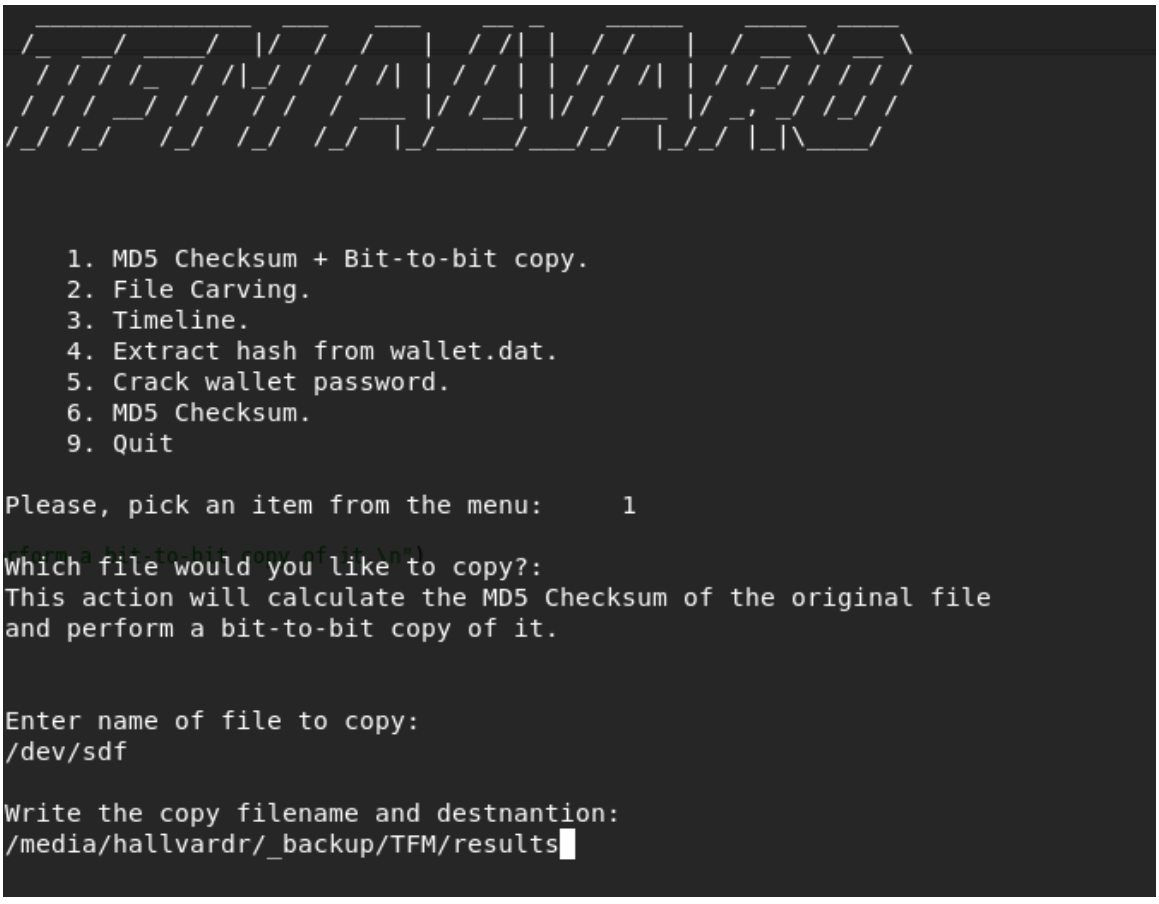

FIGURE 3.13. Create Pen Drive's binary copy and compute the Checksum

The input file will be /dev/sdf and the output file will be: /media/hallvardr/_backup/TFM/results. The image will be called copy_image.dd (**FIGURE 3.13**).

The output of the checksum function will be a file called: hashoriginal.txt. The hash can be found in **FIGURE 3.14**.

FIGURE 3.14. Initial Image hash

As said before, this hash is crucial for the investigation and must be kept unaltered.

Once the binary copy has been made, the next step will be to recover files from the pen drive. Even though with the two files that have been found, a good study could be performed, the user might have saved the wallet passwords in a text document which would facilitate the investigation. Therefore, we will carve the image copy_image.dd to restore any deleted files. (Due to the size of the file, it wasn't possible to upload them to the Github repository)

The input file will be /media/hallvardr/_backup/TFM/results copy_image.dd and the output file will be: /media/hallvardr/_backup/TFM/results/recovery/recovery (**FIGURE 3.15**).

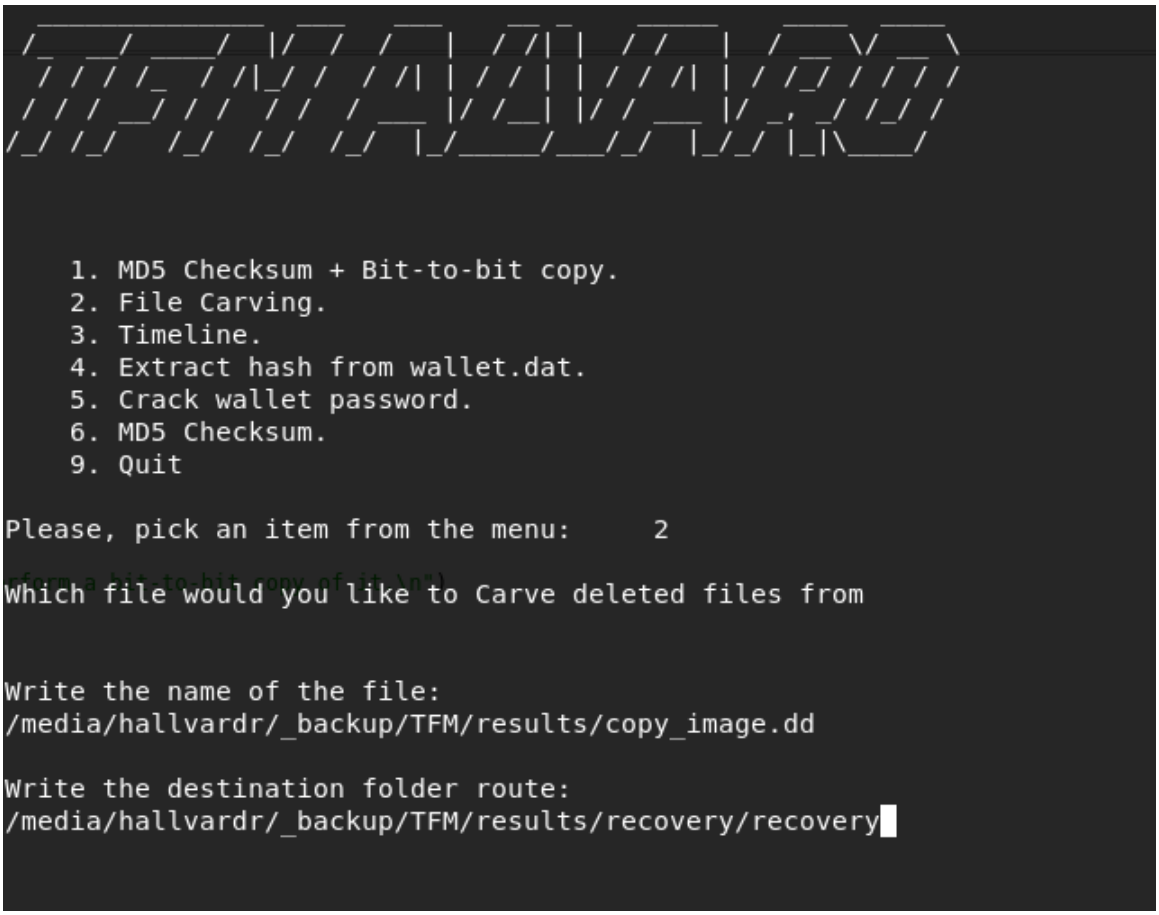

FIGURE 3.15. File carving

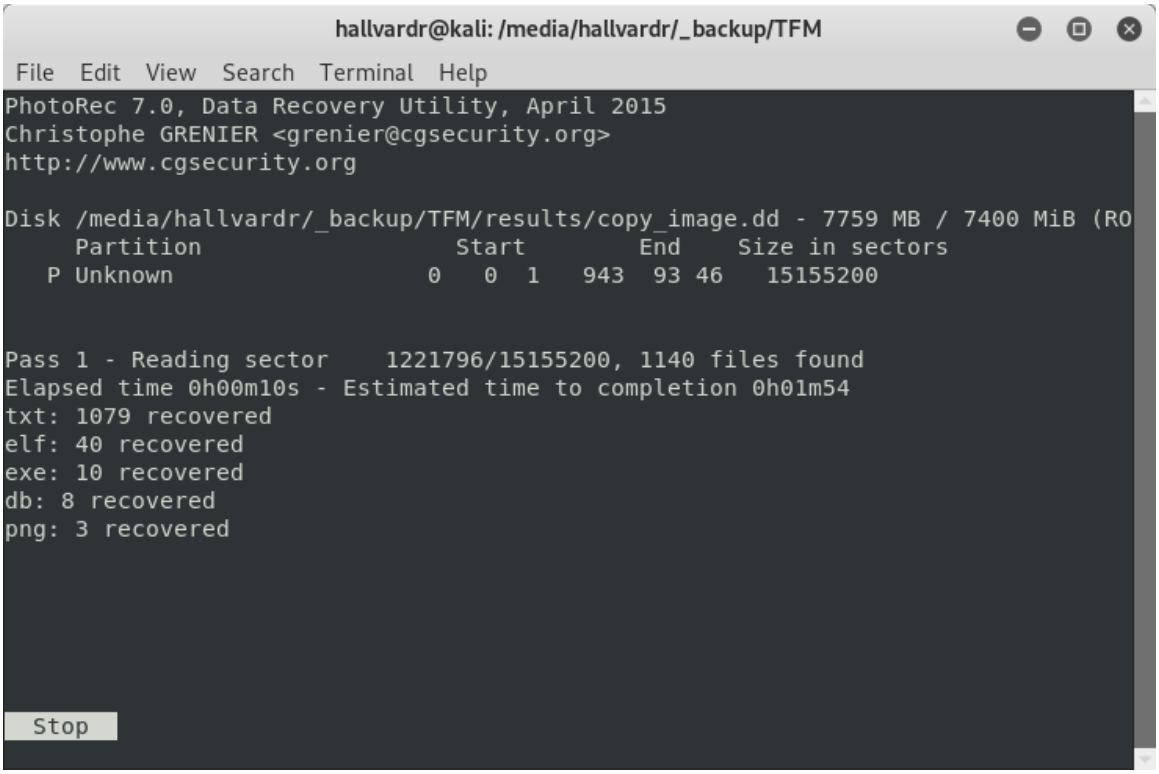

FIGURE 3.16. Pen Drive files recovery

Once the File carving process has finished, we can take a look at all the recovered files. (**FIGURE 3.17**). The recovery procedure has created 7 folders with recovered files from the historical movement of files that have taken place in the removable storage media found in the users home.

In cases like this, it is recommendable to use use other programs that help identify files. Some of this tools are OCR and eDiscovery. In this lab they weren't necessary as the pen drive didn't contain any file that might have been helpful. However in cases similar to this one, where there are a large amount of recovered files, eDiscovery could be helpful to filter by specific key words. (Due to the large amount of files, it wasn't possible to upload them to the Github repository)

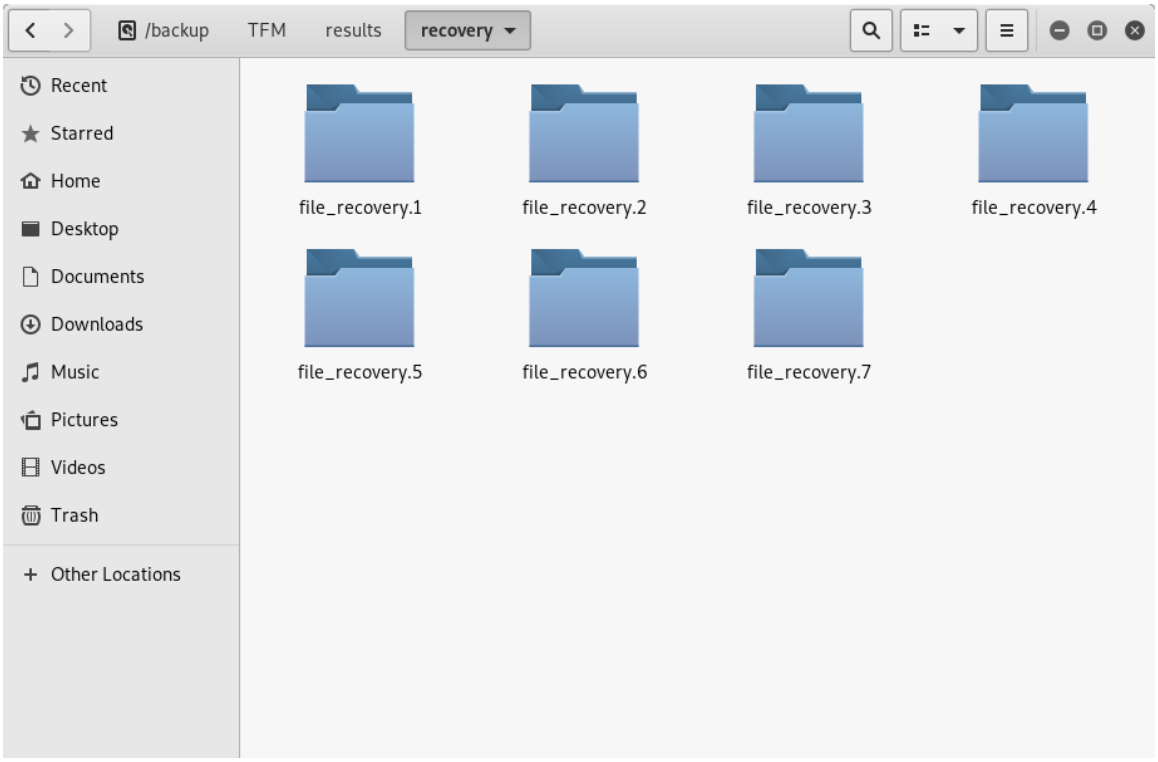

FIGURE 3.17. Recovery Files

Now we will analyze the recovery files, hoping to find any file that will help in the investigation.

One of the files seemed to contain information related to bitcoin. After looking at them, we can see some typical files of the Bitcoin wallet, but none seem to be helpful for this investigation (**FIGURE 3.18**). The other files obtained from the recovery don't seem to be related to Bitcoin or cryptocurrencies.

```
/home/hallvardr/Documents/TFM/TFM/results.1/f0244616.elf         244616-244631
/home/hallvardr/Documents/TFM/TFM/results.1/f0244632.elf         244632-244767
Sector 244768
   26644 drwxr-xr-x     0      0         0 13-Apr-2019 19:38 .
       0 drwxr-xr-x     0      0         0 13-Apr-2019 19:38 ..
   26645 -rwxr-xr-x     0      0        37 20-Jan-2019 19:04 banlist.dat
       0 -rwxr-xr-x     0      0         0  6-Feb-2019 19:58 bitcoin.conf
   26646 -rwxr-xr-x     0      0   3030928 20-Jan-2019 19:02 bitcoin-cli
   27386 -rwxr-xr-x     0      0  36748032 20-Jan-2019 19:02 bitcoin-qt
   36358 -rwxr-xr-x     0      0   3594160 20-Jan-2019 19:02 bitcoin-tx
   37236 -rwxr-xr-x     0      0  11083856 20-Jan-2019 19:02 bitcoind
   39943 -rwxr-xr-x     0      0        58  7-Apr-2019 20:56 db.log
   39944 -rwxr-xr-x     0      0       274 14-Feb-2019 20:47 hash.txt
   39945 -rwxr-xr-x     0      0   2755612  7-Apr-2019 22:23 peers.dat
/home/hallvardr/Documents/TFM/TFM/results.1/f0244768.fat         244768-244769
/home/hallvardr/Documents/TFM/TFM/results.1/f0244778.txt         244778-244783
/home/hallvardr/Documents/TFM/TFM/results.1/f0244784.elf         244784-250703
```

FIGURE 3.18. Recovery log Bitcoin data

In this case, recovering files wasn't helpful, however, in most cases, most of the controversial information is often eliminated to hide proof or evidences of a user's implication in a crime. File carving is a great method for recovering files and fragments of files when directory entries are corrupt or missing.

In certain cases related to child pornography, law enforcement agents are often able to recover more images from the suspect's hard disks by using file carving techniques. Other situations were law enforcement organization can benefit from file carving techniques are those related to terrorist warlords. During Osama Bin Laden's campus raid, U.S. Navy Seals were able to obtain hard disks and removable storage media. Forensic experts then used file carving techniques to squeeze every bit of information out of this media.

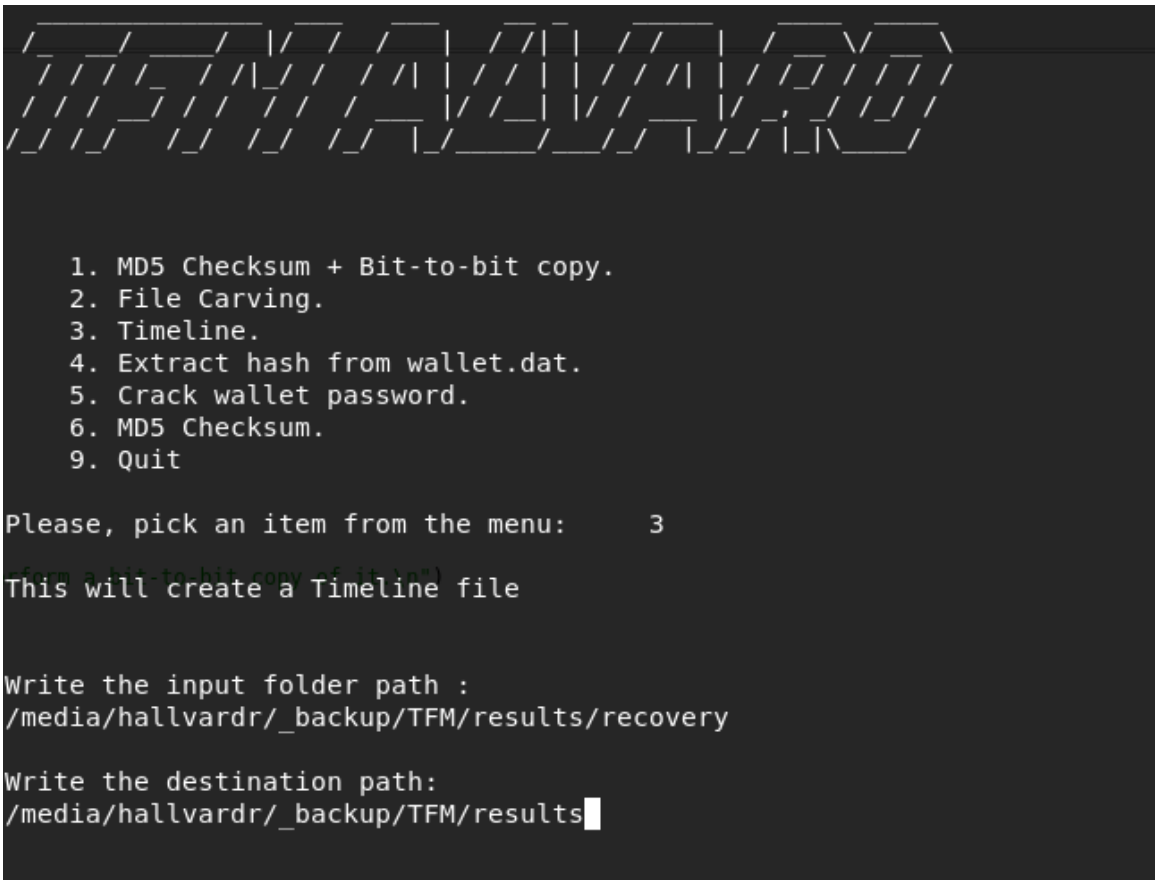

FIGURE 3.19. Timeline creation

The Timeline analysis (**FIGURE 3.19**) will give us the following fields from each file in the destination path:

- File Name
- Path
- Size
- Modified Time
- Access Time
- Created Time
- Owner

- Group
- Mode

The output of this is a .CSV file. The reason for this lays in the amount of fields obtained. By using a spreadsheet it is easier to browse through the information than in any other file extension. This process will help us identify when a file has been altered and by whom. As explained earlier, this is key to identify any kind of activity that has taken place in the system or computer paths where the study is being held.

In order to copy the files from the image created from the thumb drive, we will need to mount the image. To mount the image correctly, we should use the following commands:

```
$ sudo mkdir -p /mnt/tmp

$ fdisk -l /media/hallvardr/\_backup/TFM/results/
copy_image.dd

$ sudo mount -o ro,loop,offset=1048576 //media/hallvardr/
\_backup/TFM/results/copy_image.dd /mnt/tmp

$ sudo mount | grep /media/hallvardr/\_backup/TFM/results/
copy_image.dd

$ cd /mnt/tmp/

$ cp wallet.dat wallet1.dat /media/hallvardr/\_backup/
TFM/results

$ sudo umount /mnt/tmp
```

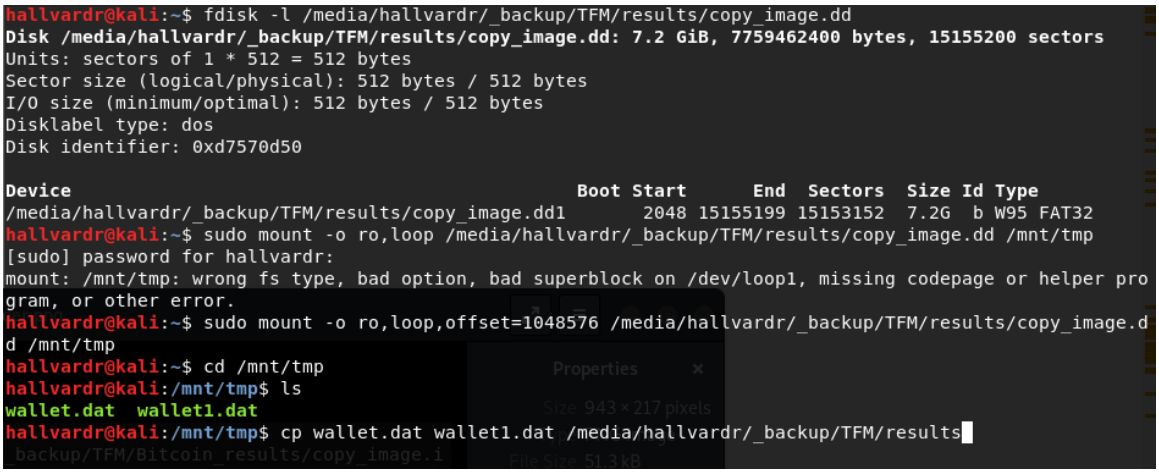

FIGURE 3.20. Mount image and copy data

We now have the wallet backup copied in a folder, where we can work with it without the fear of altering the information in it **FIGURE 3.20**. As explained before, this files contain the information regarding the private key and the Bitcoin addresses in it, amongst other things.

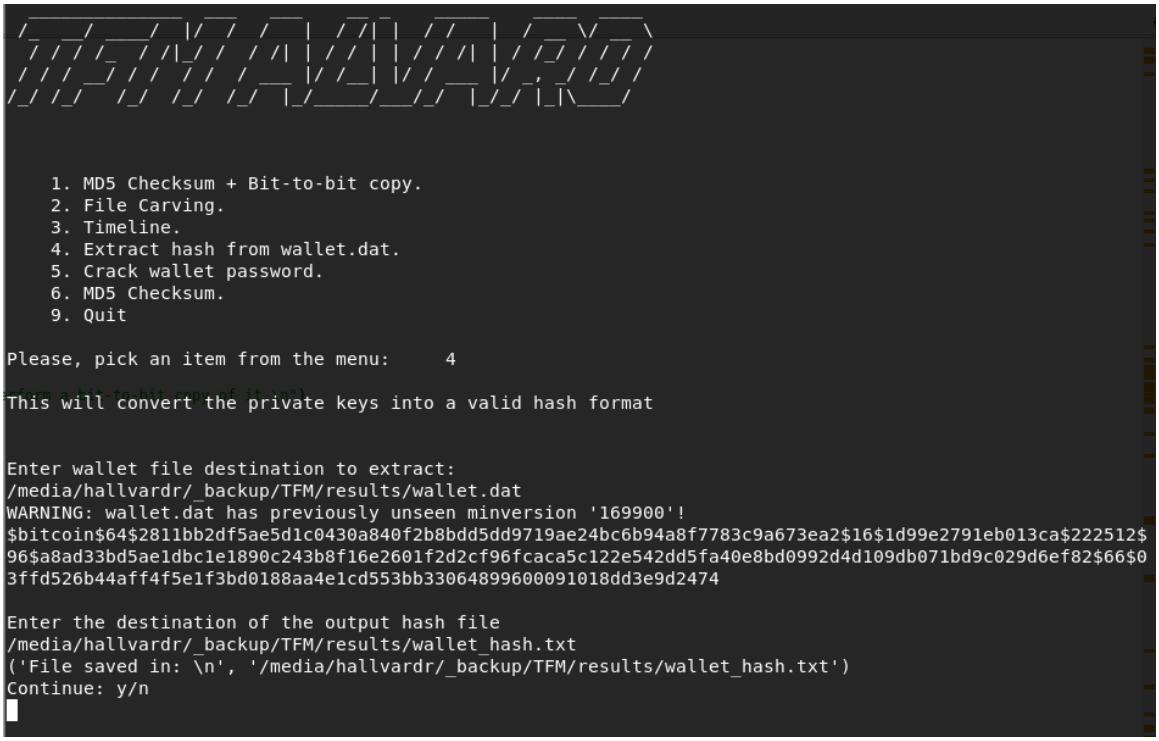

FIGURE 3.21. Wallet hash extraction

The output of this script will give us a hash, which is readable by the main password cracking tools. The input file will be the wallet backup copied from the image which is stored in: /media/hallvardr/_backup/TFM/results/wallet.dat and destination of the output hash file: /media/hallvardr/_backup/TFM/results/wallet_hash.txt.

The hash is the following:

$bitcoin$64$2811bb2df5ae5d1c0430a840f2b8bdd5dd9719ae24bc6b94a8f7783c9a673ea2$16$1d99e2791eb013ca$222512$96$a8ad33bd5ae1dbc1e1890c243b8f16e2601f2d2cf96fcaca5c122e542dd5fa40e8bd0992d4d109db071bd9c029d6ef82$66$03ffd526b44aff4f5e1f3bd0188aa4e1cd553bb33064899600091018dd3e9d2474

Once the wallet private key hash has been extracted, it is time to crack the password. In order to crack it and obtain the password behind it, we will call a system command line which will execute hashcat with the following options.

- -a 0 - (attack mode, Straight)
- -m 11300 - (Mode, Bitcoin/Litecoin)

- output wallet hash
- rockyou dictionary
- -O - (Optimized-kernel-enable)
- -w 3 - (Workload Profiles, High)

```
    1. MD5 Checksum + Bit-to-bit copy.
    2. File Carving.
    3. Timeline.
    4. Extract hash from wallet.dat.
    5. Crack wallet password.
    6. MD5 Checksum.
    9. Quit

Please, pick an item from the menu:     5
Which type of attack wopuld you like to execute?
 (1) Bruteforce
 (2) Dicctionary
 (3) Exit
2

Select dicctionary to use:
/home/hallvardr/Documents/TFM/TFM/attached/Rockyou.txt
[sudo] password for hallvardr:
hashcat (v5.1.0) starting...

OpenCL Platform #1: The pocl project
====================================
* Device #1: pthread-AMD FX(tm)-8350 Eight-Core Processor, 17988/17988 MB allocatable, 8MCU

/usr/share/hashcat/OpenCL/m11300-optimized.cl: Optimized OpenCL kernel requested but not needed - fallin
g back to pure OpenCL kernel
Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Applicable optimizers:
* Zero-Byte
* Single-Hash
* Single-Salt
* Slow-Hash-SIMD-LOOP
* Uses-64-Bit

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

* Device #1: build_opts '-cl-std=CL1.2 -I OpenCL -I /usr/share/hashcat/OpenCL -D LOCAL_MEM_TYPE=2 -D VEN
DOR_ID=64 -D CUDA_ARCH=0 -D AMD_ROCM=0 -D VECT_SIZE=2 -D DEVICE_TYPE=2 -D DGST_R0=0 -D DGST_R1=1 -D DGST
_R2=2 -D DGST_R3=3 -D DGST_ELEM=4 -D KERN_TYPE=11300 -D _unroll'
Dictionary cache built:
* Filename..: /home/hallvardr/Documents/TFM/TFM/attached/Rockyou.txt
* Passwords.: 932102
* Bytes.....: 8904979
* Keyspace..: 932102
* Runtime...: 1 sec
```


FIGURE 3.22. Wallet hash crack process

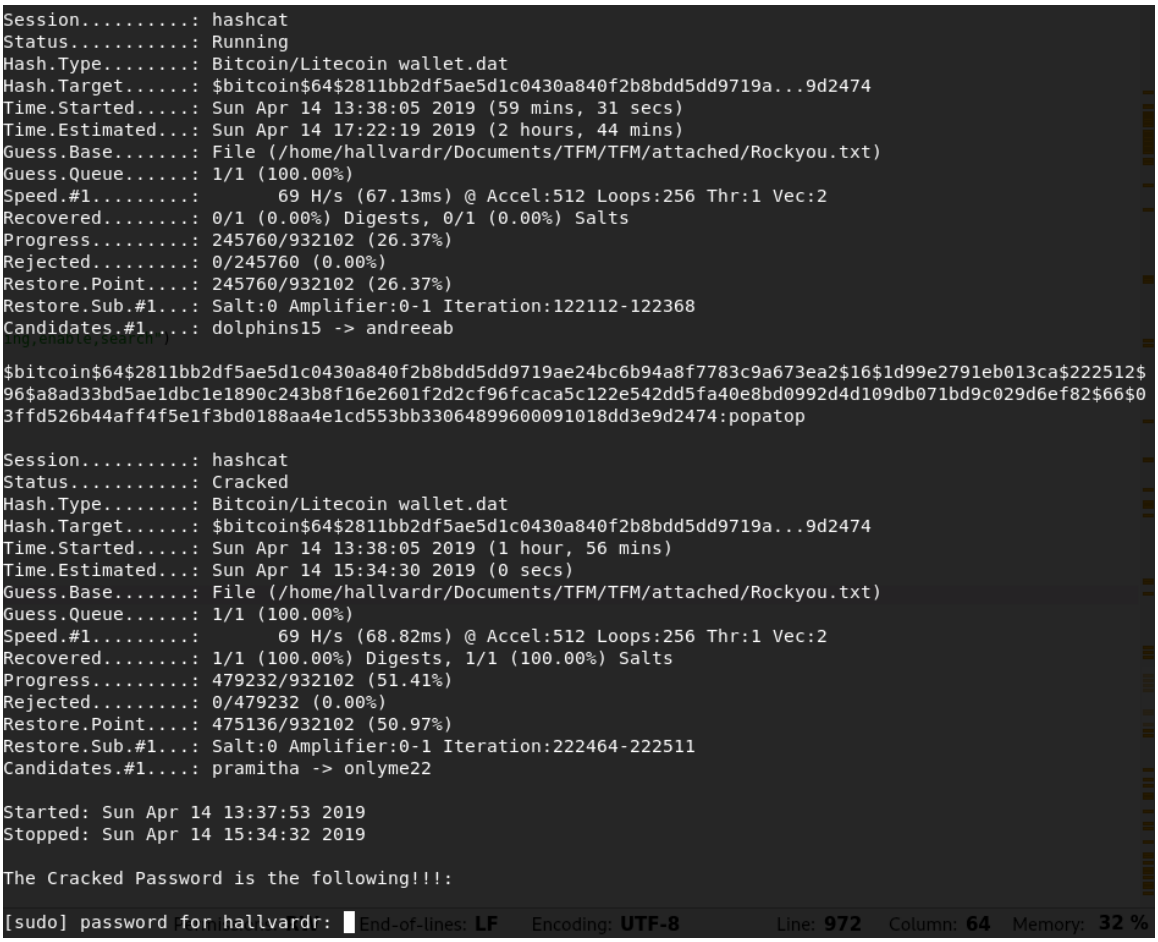

FIGURE 3.23. Password crack Process

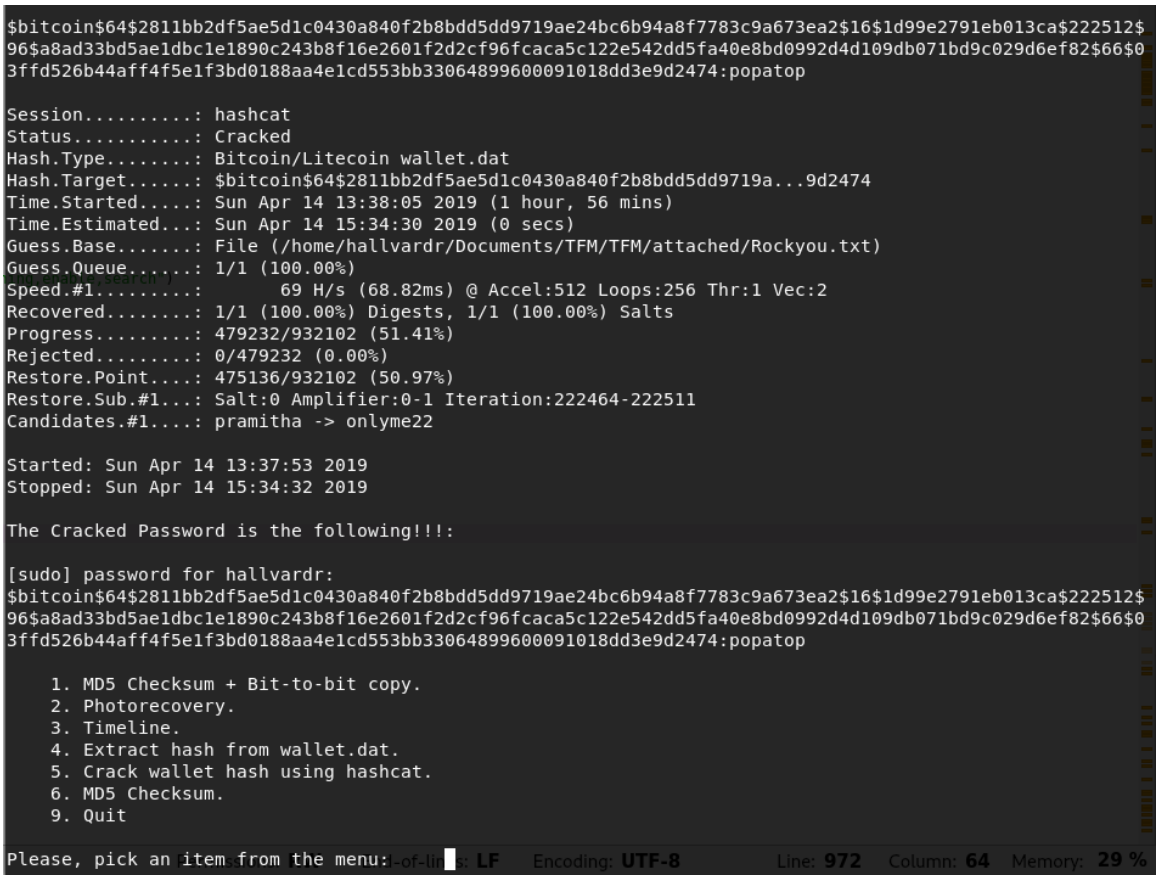

FIGURE 3.24. Password Successfully cracked

We have successfully cracked the password, we can see the password used to encrypt the wallet was: popatop. With this, we can decrypt the wallet and dump the information stored in it (as shown in the lab setup).

After successfully obtaining the wallets password, we can conclude with the work with the thumb drive image. In order to prove that no information has been altered during the investigation, we have to hash the image again. The output file can be found in the following path: /media/hallvardr/_backup/TFM/results/ output_hash.txt (**FIGURE 3.25**).

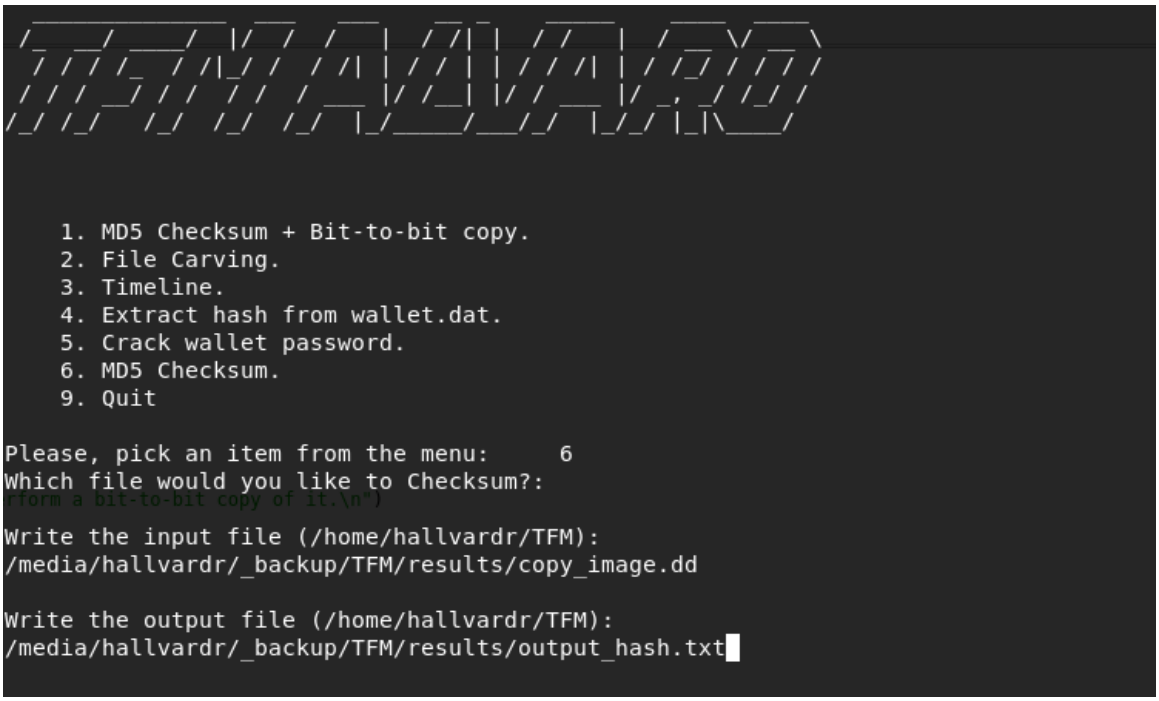

FIGURE 3.25. Image after-process hash

In order to corroborate the integrity of the data, we must compare the original hash from the thumb drive image, with the output hash once the investigation is completed. In this case we can see that both hashes are equal (see **FIGURE 3.14** and **FIGURE 3.26**)

FIGURE 3.26. Hash obtained after investigation has been completed

This means that the information used for the investigation has not been tampered or altered in any way. This point is crucial for the development of the crime report which must be presented before the court. Any slight alteration between both hashes would end up in the total invalidation of the proofs and all the work put into the investigation.

All the outputs obtained during the process described in the laboratory, have been saved in the results folder, which can bee seen in **FIGURE 3.27**. Due to size issues, both copy_image.dd and recovery, were not able to be uploaded to the Github repository.

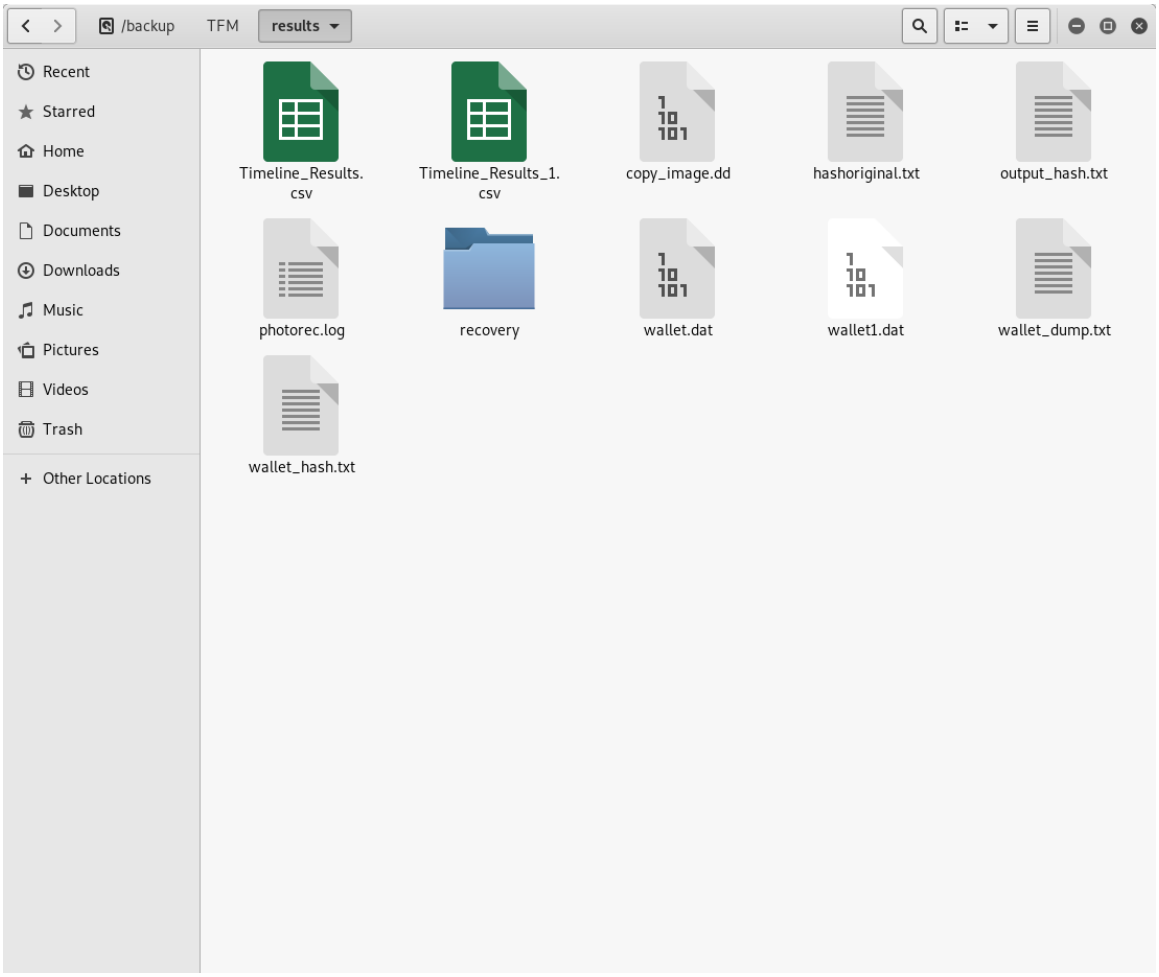

FIGURE 3.27. Results Folder

3.4.2 Hard Disk Drive Study

The next step will be to study the Bitcoin folder found in the users Hard Disk Drive (**Evidence A.6**). Although we presume the files found in the pen drive correspond to the ones that will be found in the HDD, we must be certain of it. Once we are able to confirm that the files are the same, and the that the wallet corresponds to that one, we can confirm that wallet corresponds to that Bitcoin Core account. Also, during the period between the backup was stored in the thumb drive and the current investigation of the computer, the target user may have performed any other transactions. Identifying additional transactions made by the user, may incriminate him in other cases yet undiscovered, or use programs such as bit clusters to try and link it to other addresses.

For this part of the process, we will have to repeat the same steps that we

performed for the files contained in the thumb drive. The first one will be to create a binary copy of the hard drive and create a hash of it.

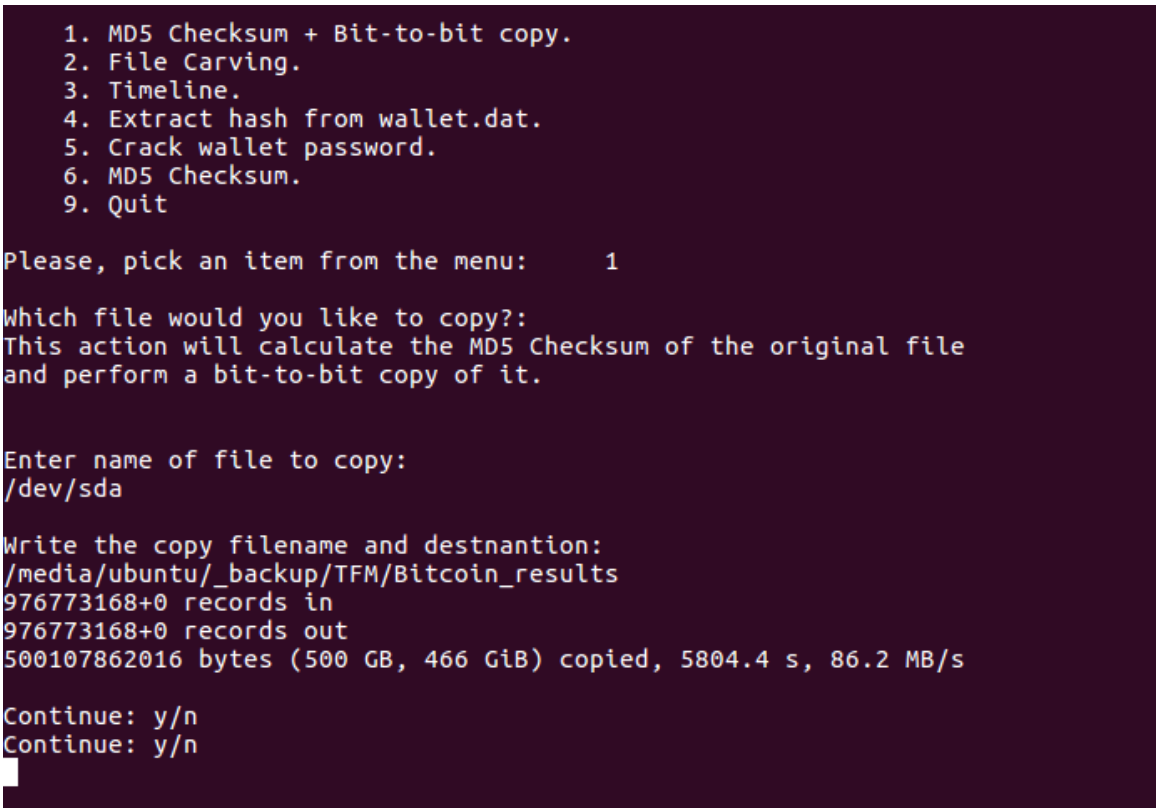

FIGURE 3.28. HDD binary copy

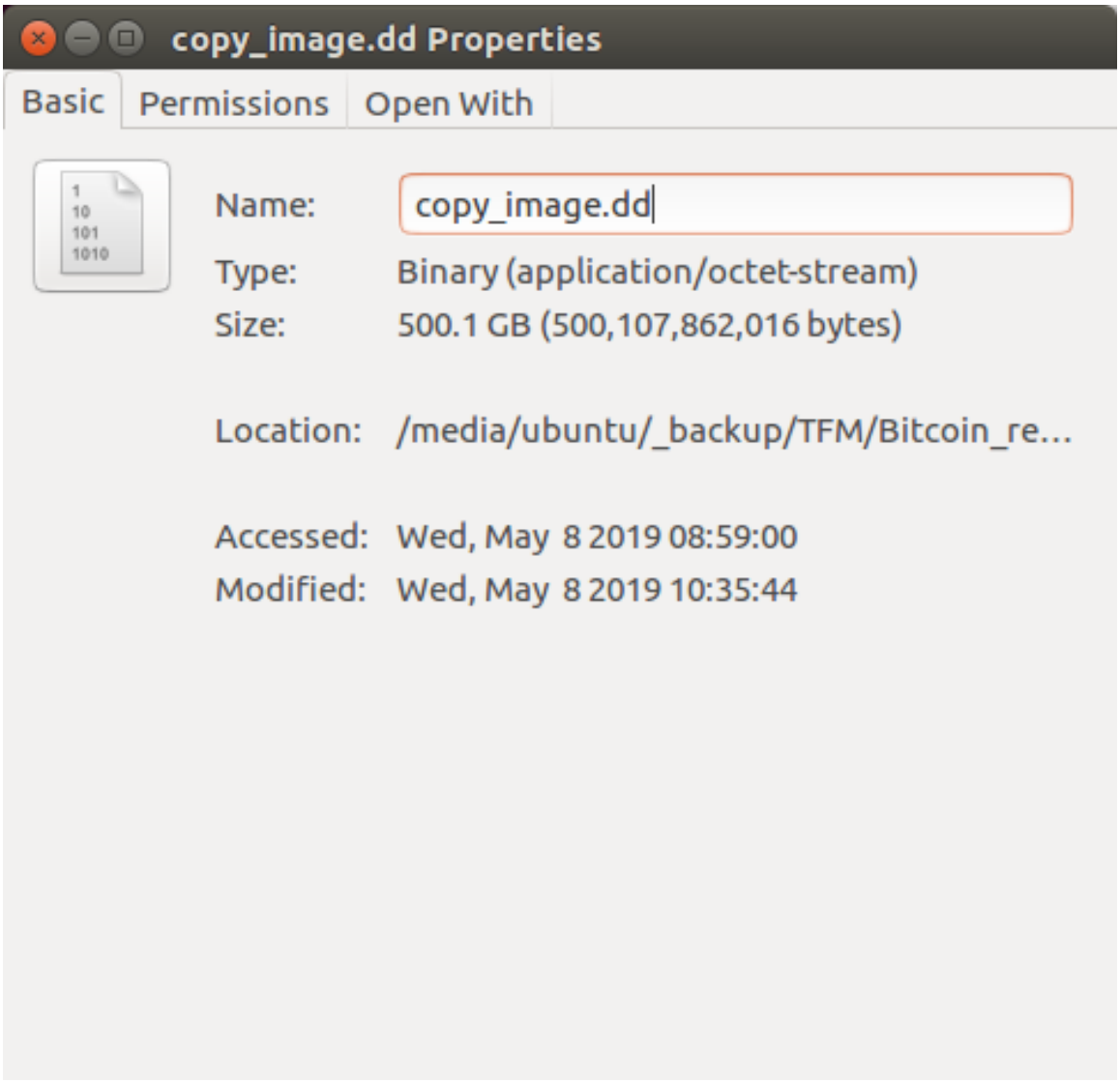

FIGURE 3.29. HDD image size

The output of the checksum function will output a file called hashoriginal.txt, however we changed the name of the file to copy_btc_hash.txt. The hash can be found in **FIGURE 3.30**.

FIGURE 3.30. Hash bitcoin folder image

FIGURE 3.31. HDD File carving

Due to a lack of space in the Hard Disk Drive where the investigation results were being stored, we have skipped this step. We will assume that since the user was actively using the wallet, he had not deleted any files related to it. A mistake during any deletion of a wallet can easily end up with the permanent loss of all the addresses that were stored in that wallet.

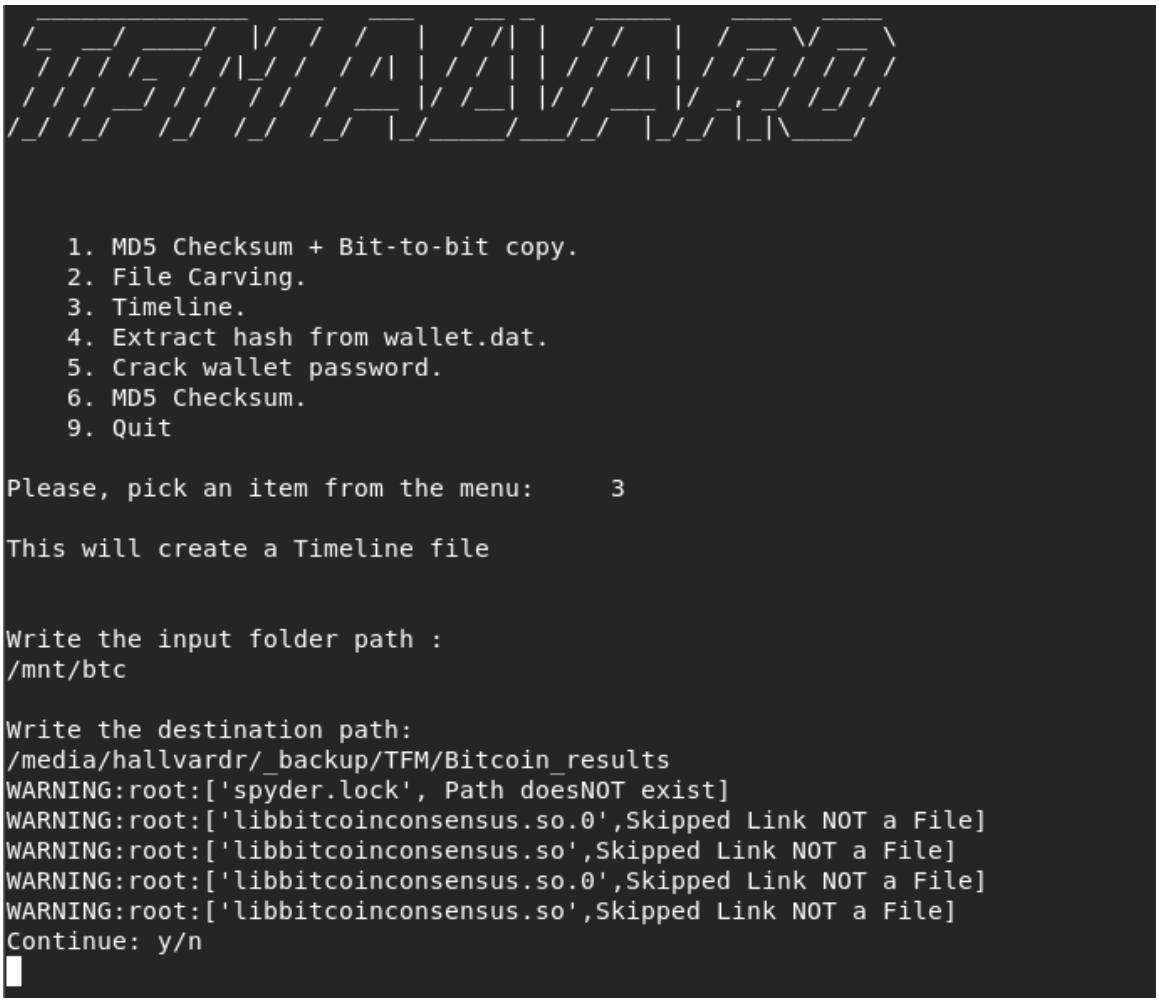

FIGURE 3.32. HDD Timeline

In order to copy the files from the image created from the Hard Disk Drive, we will need to mount the image. To mount the image correctly, we should use the following commands:

```
$ sudo mkdir -p /mnt/btc
```

```
$ sudo fdisk -l /media/hallvardr/_backup/TFM/Bitcoin_results/
HDD_copy.dd
```

```
$ sudo mount -o ro,loop,offset=1048576 /media/hallvardr/
_backup/TFM/Bitcoin_results/HDD_copy.dd  /mnt/btc/
```

```
$ cd /mnt/btc/
```

```
$ cp wallet.dat wallet1.dat /media/hallvardr/_backup/
TFM/Bitcoin_results/

$ sudo umount /mnt/btc
```

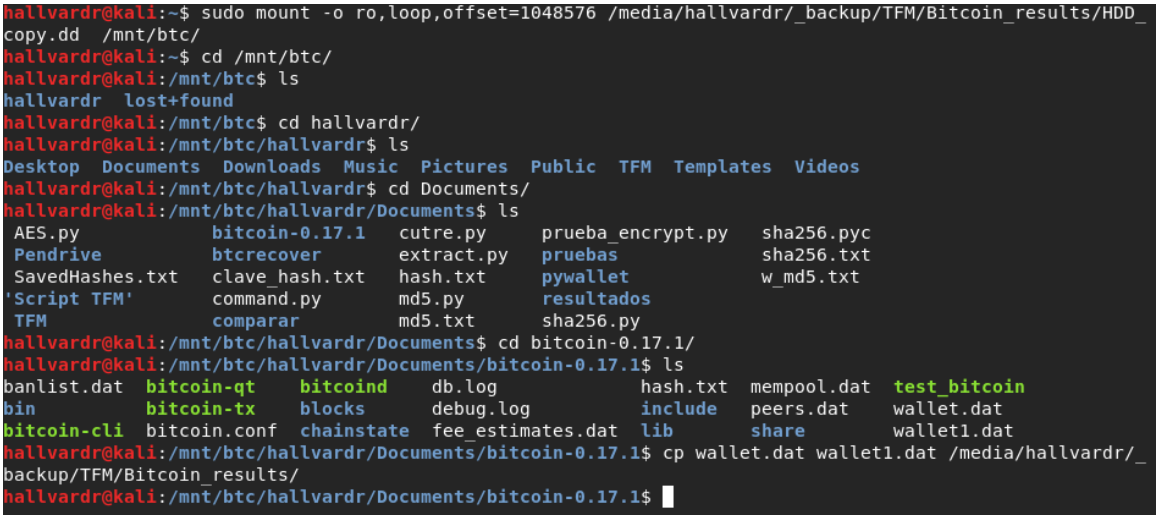

FIGURE 3.33. Copy wallets found in HDD

We have now got access to the users PC, mounted the image and have navigated to the Bitcoin folder where all the relevant information has been found. We have taken a look at the files and found only two wallet files created, both wallet.dat files correspond in name with the ones found inside the thumb drive.

We will copy the wallet files to /media/hallvardr/_backup/TFM/Bitcoin_results. The next step will be to try and dump the information stored inside the wallet. Note that if we are not able to successfully decrypt, "unlock", the wallet with the password obtained from the previous exercise, it will mean that both wallets belong to different Bitcoin accounts.

For this last exercise we will need a second computer with the Bitcoin Blockchain set up in it. This is necessary, since in order to dump the information stored in the wallet, we need to run Bitcoin's daemon process and have access to the Blockchain. For this, we have set up a second computer, also working with KALI Linux, where we have downloaded Bitcoin's Blockchain.

If we were to dump the wallets information from the targets bitcoin folder, we would automatically invalidate the whole investigation. The reason for this lies in that, in order to dump the information stored in the wallet, we have to execute Bitcoin's daemon, which will look up for any new blocks in the blockchain and start

updating any missing information. This will mean that the original hash of the Bitcoin folder and the output hash will not be identical. Hence, two different hashes will not stand before any court, and will make useless any work done in the investigation.

Going back to the lab setup, the necessary commands to dump the wallets information will be the following:

```
$ bitcoind -daemon

$ bitcoin-cli walletpassphrase popatop 120

$ bitcoin-cli dumpwallet dump.txt
```

FIGURE 3.34. Wallet dump found in HDD

If the process of dumping the wallet is satisfactory, then we can confirm that the information found in the pen drive was valid and corresponded to that Bitcoin account. This will allow us to skip steps 5 and 6 of the tool, since we have already performed them with success earlier in the investigation.

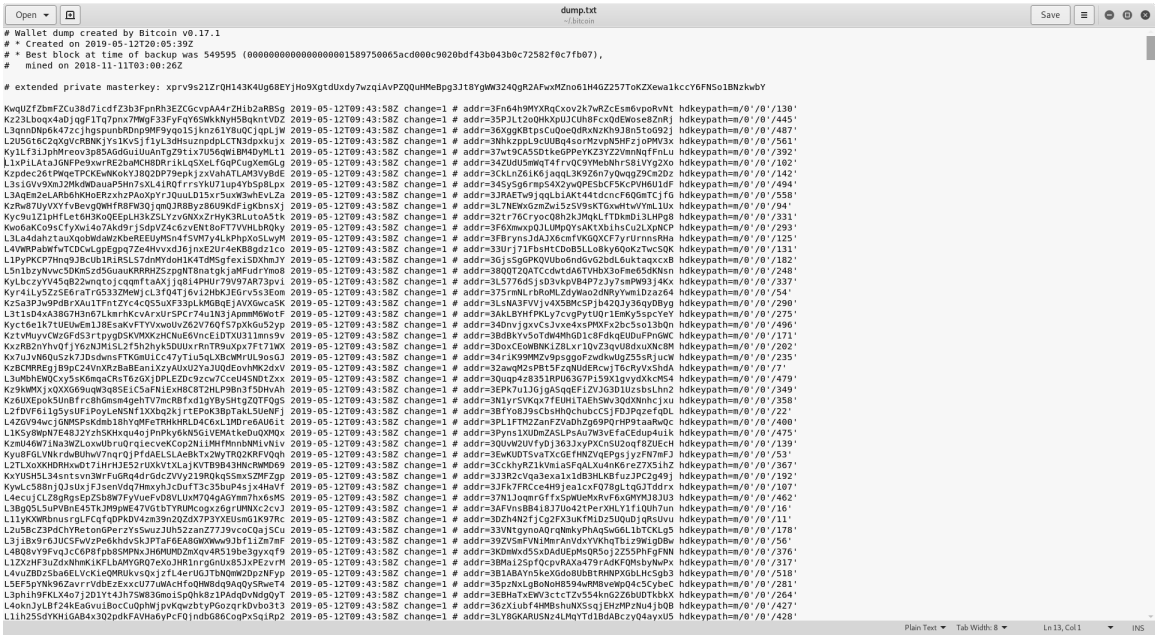

FIGURE 3.35. HDD Bitcoin wallet dump

As we can see, the wallet has been dumped successfully and the bitcoin addresses correspond to the ones in the lab set up. Therefore we can determine that indeed both wallets corresponded to the same Bitcoin account and that the address found can be related to the target. During the de-anonymization process, we identified that this address addr=34ZUdU5mWqT4frvQC9YMebNhrS8iVYg2Xo was linked to the following IP address: 2.152.6X.XXX. As we can identify in the wallet dump imag, the Bitcoin address identified corresponds to one of the addresses in the targets wallet. If we had to requisition the criminal's Satoshis or Bitcoins, we can find inside the wallet dump the private key associated to each address that belongs to the wallet. In this case, the private key that corresponds to the address is:L1xPiLAtaJGNFPe9xwrRE2baMCH8DRrikLqSXeLfGqPCugXemGLg.

This will conclude with a successful investigation and a valid defense before the court, making the evidences provided valid and incriminatory.

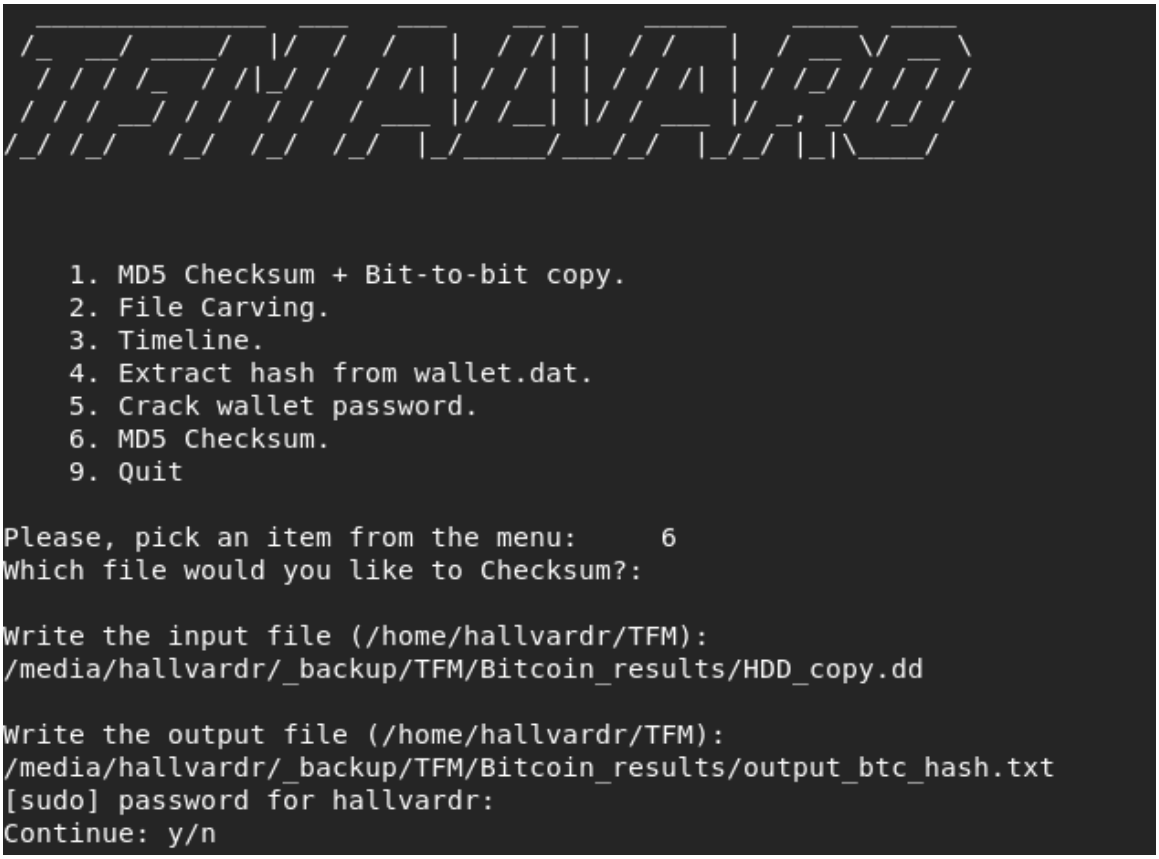

FIGURE 3.36. Hash HDD

After we have finished working with the image, we will hash it again to corroborate the file has not been tampered in any way. We must compare the original hash from the hard disk drive image, with the output hash once the investigation is completed. In this case we can see that both hashes are equal (see **FIGURE 3.30** and **3.37**).

```
470cc07a84d868ddfaaa6f1e52d1eca6  /media/hallvardr/_backup/TFM/Bitcoin_results/HDD_copy.dd
```

FIGURE 3.37. HDD image hash

This means that the information used for the investigation has not been tampered or altered in any way. This point is crucial for the development of the crime report which must be presented before the court. Any slight alteration between both hashes would end up in the total invalidation of the proofs and all the work put into the investigation.

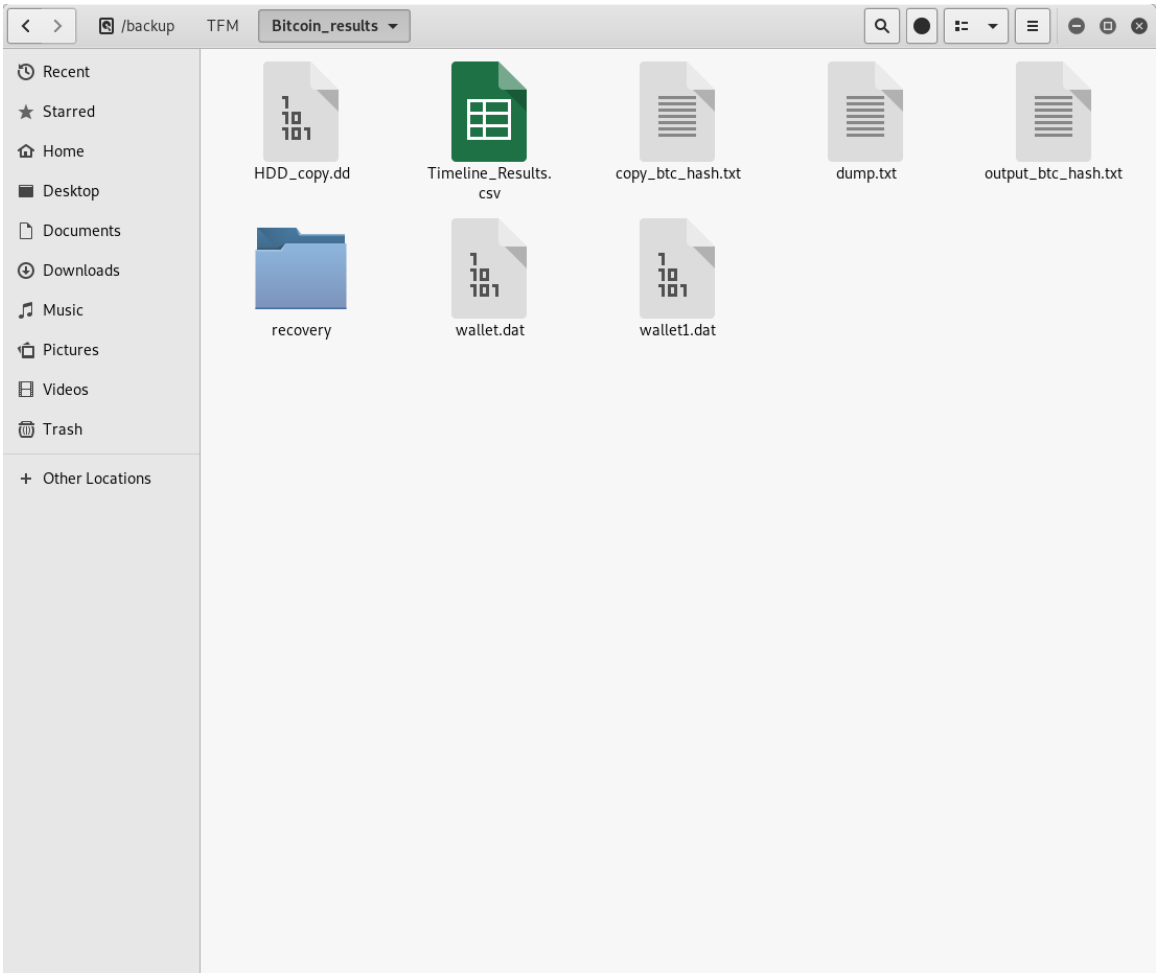

FIGURE 3.38. Bitcoin results folder

All the outputs obtained during the process described in the HDD laboratory part, have been saved in /_backup/TFM/Bitcoin_results, as well as in the Github repository. Due to size issues, both HDD_copy.dd and recovery, were not able to be uploaded to Github.

The last step in an investigation is to present a chain of custody of all the evidences found and used during the process. We have simulated a typical Digital Forensic chain of custody table, where all the evidences use have been detailed. Taking into consideration the item model or any possible marks that can help identify it. Also, it is important to note down the date the item was borrowed for investigation, by whom and where the item was found.

Description of Evidence		
Item #	Quantity	Description of Item (Model, Serial #, Condition, Marks, Scratches)
1	2	Toshiba Pendrive, 1420145A858RARG31S & 1420145A855RARG31S, Good Condition, One has WS written on it.
2	1	Kingston SSDNow 300V 120GB, SV300S37A, Good Condition.
3	1	Seagate Barracuda 7200 .12 HDD 500GB, 6VMAK3TH, Good Condition, Dust.
4	5	Unknown Pendrives 8GB, No S/N, Ubuntu 16.04, KALI and MINT written on them.

Chain of Custody				
Item #	Date/Time	Requested by	Received by	Comments/Location
1	14/04/2019 / 10:00:23	Álvaro Borreguero	Álvaro Borreguero	Found in a plastic bag on top of the table.
2	14/04/2019 / 10:00:23	Álvaro Borreguero	Álvaro Borreguero	Inside PC.
3	14/04/2019 / 10:00:23	Álvaro Borreguero	Álvaro Borreguero	Inside PC.
4	14/04/2019 / 10:00:23	Álvaro Borreguero	Álvaro Borreguero	Found in a plastic bag on top of the table.

FIGURE 3.39. Chain of Custody

The items corresponding to those in the Chain of Custody can be found in **Appendix A**.

All the work performed during the proof of concept, all the images, files, screenshots or any piece of information obtained, is crucial for the chain of custody. It is important to maintain the chain of custody to preserve the integrity of the evidence and prevent it from contamination. Any contamination or minimal change occurred during the investigation, can alter the state of the evidence. If not preserved, the evidence presented in court might be challenged and ruled inadmissible.

Suppose that we obtain metadata for a piece of evidence. However, we are unable to extract meaningful information from it, that will help refute the investigation hypothesis. The fact that there is no meaningful information within the metadata does not mean that the evidence is insufficient, there is always relevant information laying around. The chain of custody in this case helps show where the possible evidence might lie, where it came from, who created it, and the type of equipment that was used.

It is possible to have the evidence presented in court dismissed if there is a missing link in the chain of custody. In this example, if any of the media used to extract data were to go missing, the court with high certainty would invalidate the investigation resources presented. It is therefore important to ensure that a wholesome and meaningful chain of custody is presented along with the evidence at the court.

3.4.3 Hash cracking

Cracking these wallets can be fairly hardware-intensive, especially when using really long wordlists. Since bitcoin hash is harder to compute than other common hashes such as MD5 or SHA-256, it is recommended to have a GPU Rig setup for the task. The performance obtained by this hardware build, is much higher than the one a CPU

could achieve. The following tables compare the performance of some of the most common GPUs at the time and the CPU used for the laboratory, and the hashrate obtained for bitcoin hash cracking.

	NVIDIA	**NVIDIA**	**AMD**	**AMD**
Device	**GTX1080**	**TITAN**	**Vega 64 (Air)**	**FX-8350**
Hash/s Bitcoin/Litecoin wallet.dat	**4439**	**8938**	**4779**	**89**
Length of Password	**Time required in hours**			
1	3,879753E-06	1,926854E-06	3,603729E-06	1,935081E-04
2	2,405447E-04	1,194650E-04	2,234312E-04	1,199750E-02
3	1,491377E-02	7,406827E-03	1,385274E-02	7,438452E-01
4	9,246537E-01	4,592233E-01	8,588696E-01	4,611840E+01
5	5,732853E+01	2,847184E+01	5,324991E+01	2,859341E+03
6	3,554369E+03	1,765254E+03	3,301495E+03	1,772791E+05
7	2,203709E+05	1,094458E+05	2,046927E+05	1,099131E+07
8	1,366299E+07	6,785638E+06	1,269095E+07	6,814610E+08
9	8,471056E+08	4,207095E+08	7,868386E+08	4,225058E+10
10	5,252055E+10	2,608399E+10	4,878400E+10	2,619536E+12
11	3,256274E+12	1,617207E+12	3,024608E+12	1,624112E+14
12	2,018890E+14	1,002669E+14	1,875257E+14	1,006950E+16
13	1,251712E+16	6,216545E+15	1,162659E+16	6,243088E+17
14	7,760613E+17	3,854258E+17	7,208487E+17	3,870715E+19
15	4,811580E+19	2,389640E+19	4,469262E+19	2,399843E+21
16	2,983180E+21	1,481577E+21	2,770942E+21	1,487903E+23
17	1,849571E+23	9,185776E+22	1,717984E+23	9,224997E+24
18	1,146734E+25	5,695181E+24	1,065150E+25	5,719498E+26
19	7,109752E+26	3,531012E+26	6,603932E+26	3,546089E+28
20	4,408046E+28	2,189228E+28	4,094438E+28	2,198575E+30

TABLE 3.1. Time required to bruteforce a password of 62 possible characters.

	NVIDIA	NVIDIA	AMD	AMD
Device	**GTX1080**	**TITAN**	**Vega 64 (Air)**	**FX-8350**
Hash/s Bitcoin/Litecoin wallet.dat	**4439**	**8938**	**4779**	**89**
Length of Password	**Time required in hours**			
1	0,0000059	0,0000029	0,0000055	0,0002934
2	0,0005529	0,0002746	0,0005136	0,0275780
3	0,0519752	0,0258131	0,0482774	2,5923346
4	4,8856659	2,4264344	4,5380772	243,6794507
5	459,2525984	228,0848383	426,5792602	22905,8683645
6	43169,7442527	21439,9747972	40098,4504578	2153151,6262672
7	4,0579559598E+06	2,0153576309E+06	3,7692543430E+06	2,0239625287E+08
8	3,8144786022E+08	1,8944361731E+08	3,5430990825E+08	1,9025247770E+10
9	3,5856098860E+10	1,7807700027E+10	3,3305131375E+10	1,7883732904E+12
10	3,3704732929E+12	1,6739238025E+12	3,1306823493E+12	1,6810708929E+14
11	3,1682448953E+14	1,5734883744E+14	2,9428414083E+14	1,5802066394E+16
12	2,9781502016E+16	1,4790790719E+16	2,7662709238E+16	1,4853942410E+18
13	2,7994611895E+18	1,3903343276E+18	2,6002946684E+18	1,3962705865E+20
14	2,6314935181E+20	1,3069142680E+20	2,4442769883E+20	1,3124943513E+22
15	2,4736039070E+22	1,2284994119E+22	2,2976203690E+22	1,2337446903E+24
16	2,3251876726E+24	1,1547894472E+24	2,1597631468E+24	1,1597200088E+26
17	2,1856764123E+26	1,0855020803E+26	2,0301773580E+26	1,0901368083E+28
18	2,0545358275E+28	1,0203719555E+28	1,9083667165E+28	1,0247285998E+30
19	1,9312636779E+30	9,5914963818E+29	1,7938647136E+30	9,6324488383E+31
20	1,8153878572E+32	9,0160065989E+31	1,6862328307E+32	9,0545019080E+33

TABLE 3.2. Time required to bruteforce a password of 94 possible characters.

Bruteforce attacks are the simplest to put into practice, as the machine tries every single combination with the amount of characters set. This attack is the most complete, however is the most time and resource consuming, as it's not configured with any type of intelligence at all. In most cases, bruteforce attacks are left for passwords that are known to be formed with very few characters.

Due to the stated above, hash cracking experts relay on dictionary attacks. Most dictionaries are crucial in hash cracking, as they have registered most of the common words, as well as already used passwords. The most important hacker teams focused in hash cracking, own very extent dictionaries, with billions of words in it. Most of this words can be found in language dictionaries, extracts from the Wikipedia, but the biggest contributor, are leaks from other hackers that have performed SQL attacks to websites databases and have extracted the passwords stored in them.

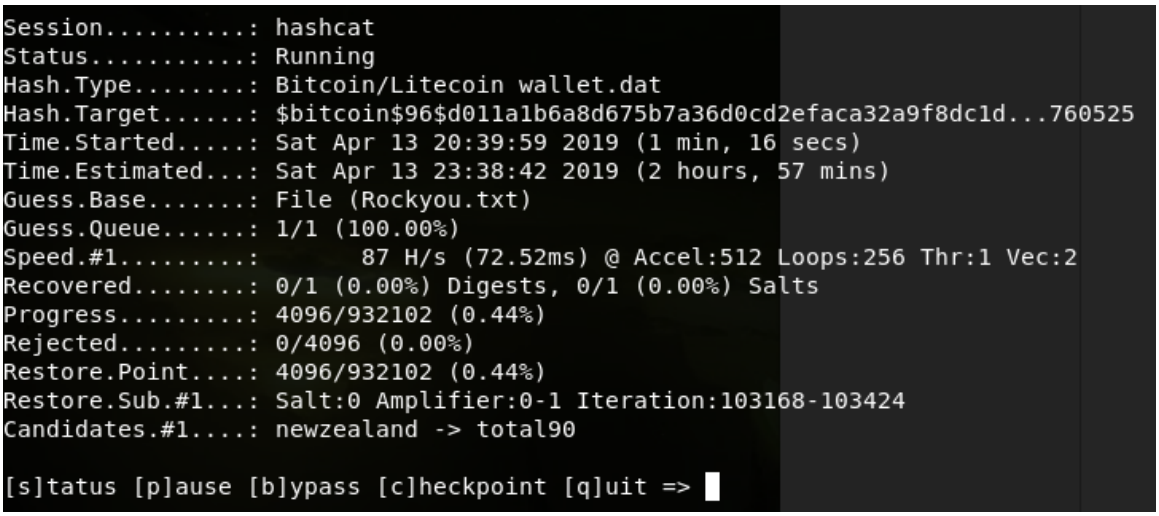

FIGURE 3.40. AMD CPU hashing computational power testing

As we can see, the hashrate obtained with the laboratory setup is quite low. In a real case scenario, where passwords are arbitrary and in most cases can not be found in most dictionaries, forensic examiners have to make use of hashing rules. This rules create new combinations of words by mixing existent words in a dictionary, in a specific way determined by the rule specification. Lets suppose a dictionary with 900000 entries in it. We decide to use a rule that tries all possible combinations with a degree of two words. This is, for every single word, we will add at the end of it each word in the dictionary.

$$Total\ combinations : 900000 * 900000 = 810000000000\ combinations$$

The total time required to complete all the possible combinations in both cases will be the following:

$$No\ rule : 900000/87 = 10344,83\ seconds$$

$$Using\ rule : 810000000000/87 = 9310344828\ seconds$$

As we can see, using extra methods to achieve better results imply spending a higher amount of time, which in many times is crucial to advance in an investigation.

	NVIDIA	NVIDIA	AMD	AMD
GPU	**GTX1080**	**TITAN**	**Vega 64 (Air)**	**FX-8350**
Hash/s Bitcoin/Litecoin wallet.dat	**4439**	**8938**	**4779**	**89**
Number of words	**Time Required in seconds**			
Crackstation - 1493677782	336489,7	167115	312550,27	16782896,43
Rockyou - 932102	209,9801757	104,2852987	195,041222	10473,05618

TABLE 3.3. Time required to run over two well known Password Cracking Dictionaries.

In **TABLE 3.3** we can observe the difference in computational power that can be achieved by using GPUs instead of CPUs to crack hashes. This output can be increased by creating GPU Rigs to crack hashes.

FIGURE 3.41. Mining Rig setup - https://www.flickr.com/photos/bitcoin-crypto/32107355744

Chapter 4

Conclusions

Although we have identified several ways of de-anonimizing users in both the standard network and TOR network, it is still very cost efficient and difficult to identify them. Exploiting regular internet failures can give us fairly nice results with very little effort, however, the kind of users we are trying to identify do not operate in the regular network.

Attacks on the Bitcoin network are easy to perform, as most of them come from P2P network vulnerabilities, however the amount of resources required to successfully de-anonymize a user are accessible to very few organizations. The 51% attack is probably the most common, and which is only applicable to Blockchain networks.

The Onion Router network provides the most robust anonymity, thanks to its layer based encryption mechanism. But this is still not 100% safe from attacks. Similar to Blockchain attacks (and P2P attacks), in order to fully de-anonymize a user, the attacker needs to own a large amount of TOR relays to obtain any valid output. The FBI is probably the best know attacker in this network, since it has already performed successful attacks in the past.

Once we have gained access to the users PC, extracting the information needed can seem fairly easy. However, if the criminal has a minimum concern for security, most of the relevant files will have been eliminated or encrypted. For this reason it is important to take into consideration techniques such as file carving, although, the most important aspect of all is probably password (hash) cracking. If we want to make sure we are able to crack any hashes we might encounter, it is important to own a very good dictionary. In order to get onto a decent dicctionary, it is necessary to continuously update it with any password leaks, add words from different languages or develop good rules to form any possible combination with the words sTORed in it. This are some of the ideas to keep a good dicctionary up to date. A ver good example is Crackstation's dictionary, however, most hackers have their own one built up for their needs.

It is important to notice that once you have completely and successfully identified a Bitcoin address to a specific user, this can be very helpful to link any future crimes. By using a transition or cluster graph, any transaction made by this users (this includes all the Bitcoin addresses linked to this user's wallet), can be useful to correlate him to other criminals (such as Coin mixing services, ilegal goods marketplace, hitman services and so on). By the creation of a "map", slowly but steadily, a security department can get to link criminals to specific Bitcoin addresses.

There are some companies that work in restoring users Bitcoin wallets or addresses (such as Btcrecover, recoverbitcoin, bitcoinrecovery, walletrecovery, etc.), which support several wallet applications. The main objective of this companies is to recover the password a Bitcoin user lost or forgot and therefore can not access it's Bitcoins anymore.

However, this companies centre themselves in password recovery, they don't follow any Forensic process. If there were a case, in which a user had been stolen, this services would not be useful if the user was to denounce the criminal who illegally took the cryptocurrencies.

Finally, this study was performed for a user that operates with Bitcoin and uses Bitcoin Core wallet. It could be extended by focusing rather on the multiple wallet applications available. It would be possible to develop scripts that could operate for mobile applications or web services. Another option would be to centre the study on other cryptocurrencies, such as Ethereum or Monero. This last one could be of more interest since it is gaining higher adoption in the Deep Web thanks to its natural transaction obfuscation. Remember that there are more than a thousand existent cryptocurrencies, so there is not a unique option.

Appendices

Appendix A

Source Code and Evidences

The Script code and all the outputs can be found in the following GitHub link: `https://github.com/Hallvardr/TFM`

The following pictures have been taken from the evidences found in the targets home.

Figure A.1. Pen drives bag found in targets home

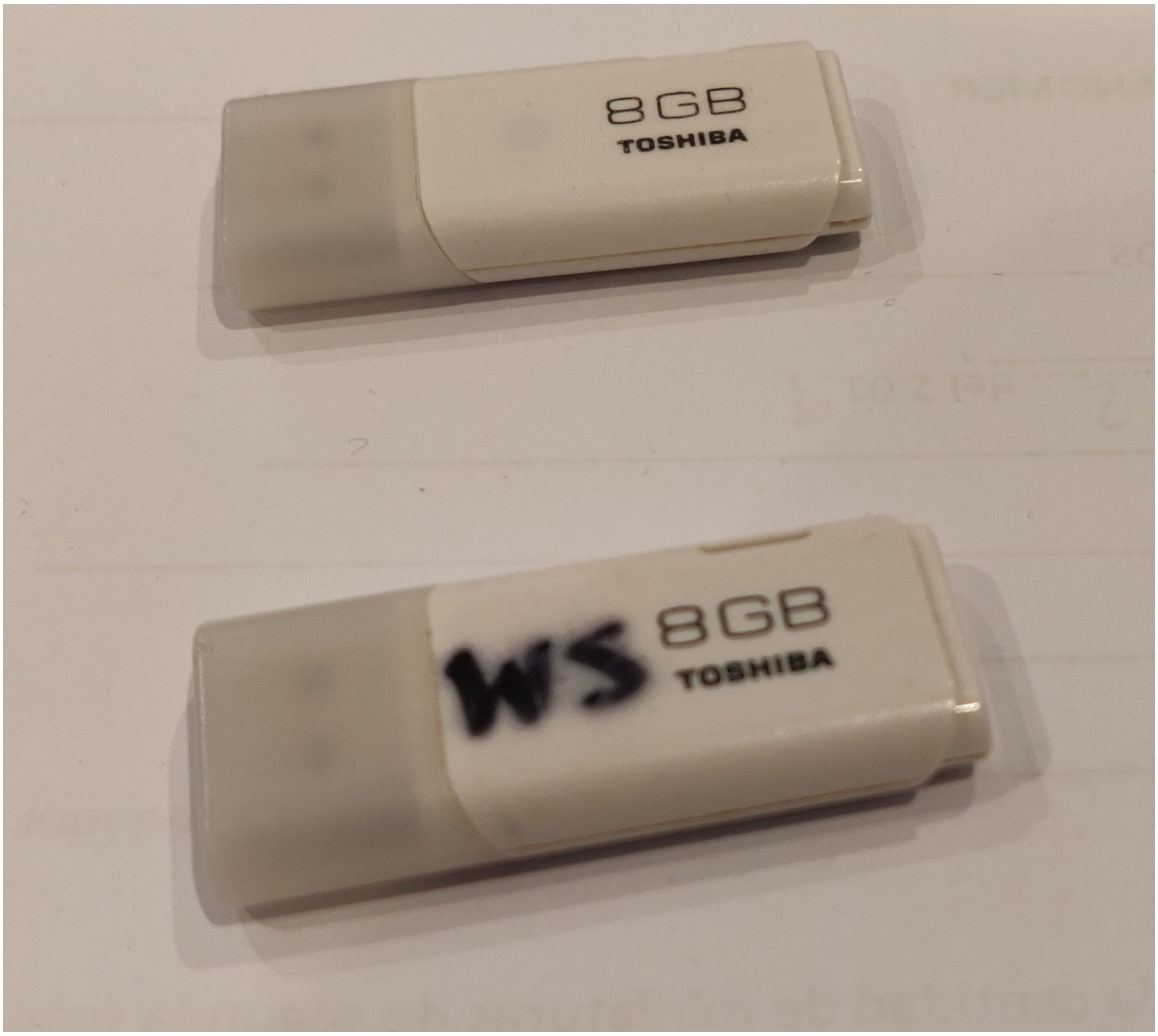

FIGURE A.2. Item 1, front

FIGURE A.3. Item 1, back

FIGURE A.4. Item 2, Kingston SSD (root files)

FIGURE A.5. Item 3, Seagate HDD (home files)

FIGURE A.6. Item 4, unknown pendrives

FIGURE A.7. Pendrive with wallet.dat files in it

Appendix B

WIF Encoding

Private key to WIF

- 1 - Take a private key:

  0C28FCA386C7A227600B2FE50B7CAE11EC86D3BF1FBE471BE89827
  E19D72AA1D

- 2 - Add a 0x80 byte in front of it for mainnet addresses or 0xef for testnet addresses. Also add a 0x01 byte at the end if the private key will correspond to a compressed public key:

  800C28FCA386C7A227600B2FE50B7CAE11EC86D3BF1FBE471BE898
  27E19D72AA1D

- 3 - Perform SHA-256 hash on the extended key

  8147786C4D15106333BF278D71DADAF1079EF2D2440A4DDE37D747
  DED5403592

- 4 - Perform SHA-256 hash on result of SHA-256 hash

  507A5B8DFED0FC6FE8801743720CEDEC06AA5C6FCA72B07C499644
  92FB98A714

- 5 - Take the first 4 bytes of the second SHA-256 hash, this is the checksum

  507A5B8D

- 6 - Add the 4 checksum bytes from point 5 at the end of the extended key from point 2

  800C28FCA386C7A227600B2FE50B7CAE11EC86D3BF1FBE471BE8982
  7E19D72AA1D507A5B8D

- 7 - Convert the result from a byte string into a base58 string using Base58Check encoding. This is the Wallet Import Format

  5HueCGU8rMjxEXxiPuD5BDku4MkFqeZyd4dZ1jvhTVqvbTLvyTJ

WIF to private key

- 1 - Take a Wallet Import Format string

  5HueCGU8rMjxEXxiPuD5BDku4MkFqeZyd4dZ1jvhTVqvbTLvyTJ

- 2 - Convert it to a byte string using Base58Check encoding

  800C28FCA386C7A227600B2FE50B7CAE11EC86D3BF1FBE471BE8982
  7E19D72AA1D507A5B8D

- 3 - Drop the last 4 checksum bytes from the byte string

  800C28FCA386C7A227600B2FE50B7CAE11EC86D3BF1FBE471BE8982
  7E19D72AA1D

- 4 - Drop the first byte (it should be 0x80). If the private key corresponded to a compressed public key, also drop the last byte (it should be 0x01). If it corresponded to a compressed public key, the WIF string will have started with K or L instead of 5 (or c instead of 9 on testnet). This is the private key.

  0C28FCA386C7A227600B2FE50B7CAE11EC86D3BF1FBE471BE89827E
  19D72AA1D

WIF checksum checking

- 1 - Take the Wallet Import Format string

  5HueCGU8rMjxEXxiPuD5BDku4MkFqeZyd4dZ1jvhTVqvbTLvyTJ

- 2 - Convert it to a byte string using Base58Check encoding

  800C28FCA386C7A227600B2FE50B7CAE11EC86D3BF1FBE471BE8982
  7E19D72AA1D507A5B8D

- 3 - Drop the last 4 checksum bytes from the byte string

  800C28FCA386C7A227600B2FE50B7CAE11EC86D3BF1FBE471BE8982
  7E19D72AA1D

- 3 - Perform SHA-256 hash on the shortened string

  8147786C4D15106333BF278D71DADAF1079EF2D2440A4DDE37D747D
  ED5403592

- 4 - Perform SHA-256 hash on result of SHA-256 hash

  507A5B8DFED0FC6FE8801743720CEDEC06AA5C6FCA72B07C49964492
  FB98A714

- 5 - Take the first 4 bytes of the second SHA-256 hash, this is the checksum

  507A5B8D

- 6 - Make sure it is the same, as the last 4 bytes from point 2

  507A5B8D

Appendix C

Commands Used

```
$ sudo apt-get install python-pyfiglet

$ bitcoind -daemon

$ bitcoin-cli encryptwallet popatop

$ bitcoin-cli walletpassphrase popatop 120

$ bitcoin-cli dumpwallet /media/hallvardr/_backup/
TFM/results/wallet\_dump.txt

$ bitcoin-cli -datadir=/home/hallvardr/Documents/
bitcoin-0.17.1 stop

$ python TFM_Alvaro.py

$ sudo mkdir -p /mnt/tmp

$ fdisk -l /media/hallvardr/_backup/TFM/results/
copy_image.dd

$ sudo mount -o ro,loop,offset=1048576 /media/hallvardr/
_backup/TFM/results/copy_image.dd /mnt/tmp

$ sudo mount | grep /media/hallvardr/_backup/TFM/
results/copy_image.dd

$ cd /mnt/tmp/

$ cp wallet.dat wallet1.dat /media/hallvardr/_backup/
TFM/results/

$ sudo umount /mnt/tmp

$ sudo mkdir -p /mnt/btc

$ fdisk -l /media/hallvardr/_backup/
TFM/Bitcoin_results/bitcoin_image.dd
```

```
$ sudo mount -o ro,loop,offset=1048576 /media/hallvardr/
_backup/TFM/Bitcoin_results/bitcoin_image.dd /mnt/btc

$ sudo mount | grep /media/hallvardr/_backup/
TFM/Bitcoin_results/bitcoin_image.dd

$ cd /mnt/btc/

$ cp wallet.dat wallet1.dat /media/hallvardr/_backup/
TFM/Bitcoin_results

$ sudo umount /mnt/btc
```

Appendix D

Bitcoin Brainwallet addresses used

Used addresses:

- Address A: 18MxiJrPcTVHSVmQJTRmYY6JYP3gGCxmCD
- Address B: 1KWXjyV3JAedCUmttr6e4x5fUKfJybzmw2
- Address C: 1Nc56SToGYnM3rUFkb6jkNcMvmST7fdiSE
- Address D: 1FiKrcvU7fa2ukvhchUvoxzv9aeQRksMJB
- Address E: 1eXpcNAda3VSpFKThPYXH7XD9BMtnRgro
- Address F: 1JfGtyFXLV2g8e9Zjikidzcq9AwyGVmsvU
- Address G: 18xxAK2tooecE9Mzyi2LzfBfFAJ4xJiZiW
- Address H: 1ECPYEfD1CuUhgQxDDkiUX21iHnGk2vzxu
- Address I: 1ACS8DMFXRpXC75vLrPykLnnqtQBoyVjun
- Address T: 1JG6z9FC6eCPmEJxbPqENUKCwn2y6ietxW

Brainwallet compressed addresses, private keys are:

- A: al pan pan y al vino vino
- B: esta es una contraseña de prueba
- C: tres tristes tigres comen trigo en un trigal
- D: la he liado parda
- E: un dia vi una vaca vestida de uniforme
- F: murcia es la mejor tierra del mundo
- G: never use password as your password
- H: 123456 is the worst possible password
- I: use rockyou.txt for a fast check
- T: this is a bad passphrase

References

[1] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy†, Geoffrey M. Voelker & Stefan Savage. *A Fistful of Bitcoins: Characterizing Payments Among Men with No Names. Internet Measurement Conference.* ACM, 2013 (cit. on pp. 71, 76). *URL: https://cseweb.ucsd.edu/ smeiklejohn/-files/imc13.pdf.*

[2] Satoshi Nakamoto. *BITCOIN: A-PEER-TO-PEER ELECTRONIC CASH SYSTEM. 2008. URL: https://bitcoin.org/bitcoin.pdf.*

[3] A. Biryukov, I. Pustogarov, and R.-P. Weinmann. "TorScan: *Tracing Long-Lived Connections and Differential Scanning Attacks". 17th European Symposium on Research in Computer Security (ESORICS 2012).* Vol. 7459. Lecture. Notes in Computer Science. Springer Berlin Heidelberg, 2012.

[4] Adrià Gil Sorribes, *A forensic look at Blockchain.* 2017.

[5] Bitcoin talk, *https://bitcointalk.org/.*

[6] Bitcoin Wiki, *https://en.bitcoin.it.*

[7] Blockchain Explorer, *https://www.blockchain.com/explorer.*

[8] Coinsutra, *https://coinsutra.com/bitcoin-double-spending/.*

[9] Medium, *https://medium.com/swlh/how-does-bitcoin-blockchain-mining-work-36db1c5cb55d.*

[10] Dmitry Ermilov, Maxim Panov, Yury Yanovich. *Bitfury, Automatic Bitcoin Address Clustering.* https://bitfury.com/content/downloads/clustering_whitepaper.pdf.

[11] Blockonomi Silk Road, *https://blockonomi.com/history-of-silk-road.*

[12] Gawker Silk Road, *https://gawker.com/the-underground-website-where-you-can-buy-any-drug-imag-30818160.*

[13] TOR Poject, *https://blog.TORproject.org/traffic-correlation-using-netflows.*

[14] QingChun ShenTu, JianPing Yu, *Research on Anonymization and de-anonymization in the Bitcoin System.* 2015.

[15] Data extraction, *URL: https://Bitcointalk.org/index.php?topic=191039 .msg1980099#msg1980099.*

[16] Data injection, *URL: https://Bitcointalk.org/index.php?topic=11381.0*

[17] Crypto Exchange, Brute Forece SHA256, *https://crypto.stackexchange.com/questions/1145/how-much-would-it-cost-in-u-s-dollars-to-brute-force-a-256-bit-key-in-a-year.*

[18] Christoph Kinkeldey, Jean-Daniel Fekete, Petra Isenberg. *BitConduite: Visualizing and Analyzing Activity on the Bitcoin Network. EuroVis 2017 - Eurographics Conference on Visualization, Posters Track, Jun 2017, Aire-la-Ville, Switzerland. pp.3, 2017.* <hal-01528605>.

[19] REID F., HARRIGAN M.: *An Analysis of Anonymity in the Bitcoin System. Springer New York, New York, NY, 2013,* pp. 197–223.doi:10.1007/978-1-4614-4139-7_10. 2.

[20] A. Kwon, M. AlSabah, D. Lazar, M. Dacier, and S. Devadas, *"Circuit fingerprinting attacks: passive de-anonymization of tor hidden services," in USENIX Security Symposium. USENIX Association, 2015, pp. 287– 302.*

[21] Center of Computer Forensics: *URL: http://www.computer-forensics.net/index.html*

[22] SANS Python taxonomy *URL: https://www.sans.org/reading-room/whitepapers/incident/grow-forensic-tools-taxonomy-python-libraries-helpful-forensic-analysis-33453*

[23] DigiCash. *URL:http://www.chaum.com/projects/eCash/ecash.html*

[24] A. Back. Wei Dei's "b-money" protocol. Cypherpunks Archives. *URL:http://cypherpunks.venona.com/date/1998/12/msg00194.html*

[25] InfoSecInstitute *URL: https://resources.infosecinstitute.com/*