TALLINN UNIVERSITY OF TECHNOLOGY

School of Information Technologies

Juan Manuel Delgado García 194336IVCM

FORENSIC ANALYSIS OF PRIVACY-ORIENTED CRYPTOCURRENCY WALLETS

Master's Thesis

Supervisor: Dr. Hayretdin Bahsi

Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Juan Manuel Delgado García.

14.05.2021

Abstract

The increasing use of privacy-oriented cryptocurrencies due to the privacy and anonymity features these offers, allow cybercriminals to commit illegal transactions that have raised the concern of the law enforcement agencies because they are harder to trace back than Bitcoin. However, Bitcoin remains the most traded and used cryptocurrency, so there is a considerable number of forensic studies related to it, while privacy-oriented currencies have fewer studies. The present research focuses on the forensic analysis of cryptocurrency wallets Zcash and Dash with the purpose to elaborate a technical guide that supports the investigator work, showing what forensic artefacts can be obtained and be helpful during an investigation. To achieve the purpose of the study, methods such as memory acquisition, disk acquisition and network traffic acquisition have been analysed. From these analyses, valuable forensic artefacts were obtained, like the transaction IDs, mnemonic phrase, and private keys.

Abstranke

Privaatsusele orienteeritud krüptovaluutade suurenev kasutamine privaatsuse ja anonüümsuse funktsioonide tõttu, mida need pakuvad kasutajatele ebaseaduslike tehingute sooritamiseks, on tekitanud õiguskaitseasutustele muret, kuna neid on raskem jälgida kui Bitcoin. Bitcoin on endiselt enim kaubeldav ja kasutatav krüptoraha. Seega on sellega seotud märkimisväärne arv kohtuekspertiise, samas kui privaatsusele orienteeritud valuutade kohta on vähe uuringuid. Käesolev uuring keskendub krüptoraha rahakottide Zcash ja Dash kohtuekspertiisi analüüsile, eesmärgiga töötada välja uurija tööd toetav tehniline juhend, mis näitab, milliseid kohtuekspertiisi esemeid on võimalik hankida ja mis võivad olla uurimise käigus kasulikud. Uuringu eesmärgi saavutamiseks on analüüsitud selliseid meetodeid nagu mälu omandamine, ketta hankimine ja võrguliikluse omandamine. Nende analüüside põhjal saadi väärtuslikke kohtuekspertiisi artefakte, nagu tehingu ID-d, mnemooniline fraas ja privaatvõtmed.

Abbreviations

ASCII	American Standard Code for Information Interchange
CLI	Command-line Interface
CSV	Command-separated Values
DASH	Digital Cash
DNS	Domain Name System
FTK	Forensic Toolkit
GDPR	General Data Protection Regulation
HD	Hierarchical Deterministic
IP	Internet Protocol
JSON	JavaScript Object Notation
MFT	Master File Table
NIST	National Institute of Standards and Technology
OS	Operative System
PII	Personally Identifiable Information
RAM	Random Access Memory
PID	Process Identifier
RPC	Remote Procedure Calls
SHA	Secure Hash Algorithm
TOR	The Onion Router
VDI	Virtual Drive Image
VM	Virtual Machine
XPRV	Extended Private Key
XPUB	Extended Public Key
zk-SNARK	Zero-Knowledge Succinct Non-Interactive Argument of Knowledge

Table of Contents

List of Tables

List of Figures

1 Introduction

Nowadays, talking about cryptocurrency has become a mainstream topic. In fact, the first word that comes into mind when talking about cryptocurrencies is Bitcoin. Bitcoin is, without doubt, the most traded and well-known cryptocurrency [1].

Bitcoin has huge popularity, including amongst criminals, as a means of payment for illegal activities such as drug dealing, weapons trade, child pornography, money laundering, and cyberattacks. Related to cryptocurrency in crime, “over 97% of illicit activity on the darknet has been conducted through Bitcoin over the years.” [2].

However, from a criminal perspective, Bitcoin has a “*weakness*”, which is its lack of privacy and anonymity, making it difficult to hide felonies behind this cryptocurrency. This lack of privacy and anonymity is due to how Bitcoin works, registering all transactions in a public ledger called Blockchain, making it possible for private companies and law enforcement agencies to trace the source of the illegal transactions.

Proof of that is a study conducted in “2018 by blockchain analysis startup, Elliptic and the Center on Sanctions and Illicit Finance that found a fivefold increase in the number of large-scale illegal operations working on the Bitcoin blockchain between 2013 and 2016. By analysing the history of more than 500,000 bitcoins, the organisations identified 102 criminal entities, which included dark-web marketplaces, Ponzi schemes and ransomware/malware attackers” [3].

For that same reason, several studies related to the forensic analysis of Bitcoin have been carried out, and considerable information associated with that subject can now be found on the internet.

However, this is changing, and a group of cryptocurrencies called privacy-oriented has gained popularity with criminals because they have built-in anonymity and privacy features that make them harder to trace than Bitcoin. For instance, some of these features allow obscuring both the transaction recipient and transaction amount. Another example is that users can make transactions without revealing their addresses to others.

Privacy-enhancing coins such as “Monero is gradually becoming the most established privacy coin for Darkweb [1] transactions, followed by Zcash and Dash” [5].

Zcash is a cryptocurrency that makes use of a cryptography technique called zero-knowledge proof. This technique allows Zcash to encrypt the transaction details, including the sending and recipient address, on the blockchain.

On the other hand, Dash makes use of the technique called Coin Mix. This technique consists of mixing the coins from different users and sending these mixed coins to the desired recipient address in one transaction. This technique hides the transaction’s real source and destination when observed in the blockchain explorer.

With these features that enhance privacy and anonymity, the work of investigators will become harder to achieve due to the information of the transaction is not public in the blockchain. For instance, only by analysing the blockchain it was possible to detect a significant trade of 28 bitcoins (approx. $522,000) that had as destination people involved in the riots that occurred last January 6 in the U.S. Capitol [6] [7]. Therefore, if the transaction would have been done with some privacy-oriented cryptocurrency, perhaps this conclusion would not have been possible.

Consequently, the forensic analysis of a suspect’s wallet is crucial since it can reveal details about the transactions that only by checking the blockchain it would be difficult or impossible to determine the source and destination of such illegal activity.

For the previously exposed and considering that there is little study on privacy-oriented cryptocurrencies, the forensic analysis of these systems is crucial to tackle criminals who are taking advantage of the privacy features and cover their felonies behind them.

[1] “The dark web is the hidden collective of internet sites only accessible by a specialized web browser. It is used for keeping internet activity anonymous and private, which can be helpful in both legal and illegal applications” [4].

1.1 Scope and Goal

The scope of this study is to focus on the forensic investigation of privacy-oriented cryptocurrency wallet software to identify what forensic artefacts can be collected as a result of the user's interaction with the application.

For this study, the selected wallets have been Zcash and Dash, based on the increase of their acceptance on the Darkweb markets [5] and the possibility to undermine the work of law enforcement agencies to detect activities that finance terrorism or money laundering [8].

The main outcome of this research is the creation of a technical guide to be used mainly by law enforcement authorities or any other person who wishes to perform a forensic analysis of this type of cryptocurrency wallets. The fundamental concept of this technical guide is to advise about what kind of forensic artefacts can be collected, where and how to acquire them.

1.2 Research Problem

The utilisation of new cryptocurrencies by criminals has increased, and so has the usage of new software wallets. This study will analyse these software wallets to identify their forensic value and provide support through the development of a technical guide.

Consequently, this research attempts to answer the following questions:

- What forensic artefacts can be obtained from the analysis of the Zcash and DASH wallets in their full node and light version?
- How different are Zcash and DASH regarding their light and full node version in terms of forensic artefacts obtained after the analysis?
- What artefacts obtained after the forensic analysis can be used in the blockchain for further steps on the investigation.

1.3 Key Assumptions

The study assumes that:

The forensic framework followed during this study helps preserve the integrity of the evidence collected and analysed, making it legally acceptable.

The software wallet analysed does not harm or infect with malware the guest OS even when this last one raised an alert classifying the wallets as malware.

Results obtained from the data collected from the virtual environment do not differ from the results that can be obtained if the data would have been collected from a non-virtual environment.

1.4 Ethical Issues

The cryptocurrency addresses and the guest operating system where the wallets were installed do not contain Personally Identifiable Information[2] (PII) that can compromise someone's identity at the moment of data collection and analysis.

The cryptocurrency addresses and their corresponding private keys used during the transactions between the different cryptocurrency wallets were created and handled only for this study and did not compromise someone else's funds.

1.5 Novelty

The illicit activity as a percent of total transactions of one of the most popular privacy-oriented cryptocurrencies, such as Monero, is by far more significant than it is for Bitcoin [6]. Furthermore, a study conducted in 2020 shows that there has been a shift from Bitcoin to privacy-oriented cryptocurrencies in the dark web markets [10].

The increasing use of privacy-oriented cryptocurrencies has raised the alarm for law enforcement agencies [5], [8]. Moreover, an ex-CIA agent expert has reported that terrorist groups have started to use different cryptocurrencies that employ anonymising

---

[2] Information gathered from different sources that can be related to an individual's identity, rendering it identifiable [9] .

techniques in the flow of funds that support their activities, becoming a key part to monitor [6].

Due to the features that allow Zcash and Dash to offer more privacy and anonymity to their users, these represent serious competition for Bitcoin in the Darkweb markets when doing illicit activities [10]. Furthermore, cybercriminals offer discounts to their victims when the ransom payment is made with privacy-oriented cryptocurrencies [11].

Most of the forensic analysis of cryptocurrency wallets is related to Bitcoin [12]–[18], but as it can be seen above, Bitcoin is being left behind due to the existence of alternatives that offer anonymity and privacy features, allowing people to hide their illicit activities behind these “new” privacy-oriented cryptocurrencies.

1.6 Outline of the Thesis

Chapter 2 introduces the central concepts used for the thesis and reports a general review of existing related literature. Chapter 3 lays the methods used during the investigation and the theoretical part of the research. Chapter 4 shows the results of the data collected. Chapter 5 discusses the results of the experiments performed. Chapter 6 presents the conclusions and suggests future work.

2 Background Information

This chapter aims to provide a basic understanding of the most relevant concepts employed during the development of this thesis. The chapter will begin by giving the concept of privacy and anonymity, next presents the concept of digital forensics, later provides a brief description of Zcash, DASH and cryptocurrency wallet to finally show some related work.

2.1 Privacy

According to the definition given by the Cambridge Dictionary, privacy is “someone’s right to keep their personal matter and relationships secret” [19].

Taking that definition into a digital world, and according to the General Data Protection Regulation (GDPR), “Data privacy means empowering your users to make their own decisions about who can process their data and for what purpose” [20]. In other words, a user keeps his/her activity entirely private for himself/herself or limited to a defined group of people.

Therefore, in the case of cryptocurrency, it is the user’s ability to make transactions without revealing partial or complete information about the transaction. This information can be the amount, the sender’s address, or the recipient’s address.

2.2 Anonymity

According to Kathleen A. Wallace, it is defined as “the *non-coordinatability of traits in a given respect*. In other words, one has anonymity or is anonymous when others are unable to relate a given feature of the person to other characteristics.” [21].

To put it in other words, a user can let someone else see what he/she is doing without being identifiable. Hence, in the case of cryptocurrency, anonymity is the user’s ability to make transactions without being recognised by someone else, even if the transaction’s information is revealed or not [22].

2.3 Digital Forensics

The science that concentrates its efforts in the recovery and analysis of information obtained from digital devices involved in cybercrimes is called Digital Forensics. This term was recognised in the 1990s, but it was not until the beginning of the 21st century when the police forces started to create units specialised in this field [23].

“Digital forensics is the process of identifying, preserving, analysing, and documenting digital evidence. This is done to present evidence in a court of law when required” [23].

According to the National Institute of Standards and Technology (NIST) in the Special Publication (SP) 800-86 (NIST SP 800-86), “Forensic tools and techniques are most often thought of in the context of criminal investigations and computer security incident handling-used to respond to an event by investigating suspect systems, gathering, and preserving evidence, reconstructing events, and assessing the current state of an event”

[24]. Likewise, the NIST SP 800-86 mentions that regardless of the situation in which those techniques and tools are required, the forensic process comprises the following basic phases [24]:

- **Collection:** “identifying, labelling, recording, and acquiring data from the possible sources of relevant data, while following procedures that preserve the integrity of the data” [24].

- **Examination:** with the use of manual and automatic methods, the collected data is processed, and relevant information is extracted, preserving the integrity [24].

- **Analysis:** “analysing the results of the examination, using legally justifiable methods and techniques, to derive useful information that addresses the questions that were the impetus for performing the collection and examination” [24].

- **Reporting:** the results of the analysed data is presented, explaining what methods were used to obtain the results and what tools were employed during the analysis. [24]. Moreover, the report can include recommendations about the tools, procedures or policies [24].

2.4 Zcash

Zcash is a cryptocurrency focused on privacy and anonymity. It was launched in 2016 as a fork of Bitcoin Core, and it makes use of Zero-Knowledge Proofs, which “are an elegant technique to limit the amount of information transferred from a prover A to a verifier B in a cryptographic protocol” [25].

In the case of Zcash, the sender can prove to the recipient that the transaction has been done without revealing any information about the transaction itself [26]. To achieve this, Zcash uses a type of proof called zk-SNARKs [26].

Zcash has two types of addresses: one called Transparent address (t-addr or T), which starts with “t”, and Shielded address (z-addr or Z), which begins with “z”. Each address has its corresponding private key. For instance, Table 1 shows the addresses and their corresponding private keys obtained using the “export private keys” option from the wallet application installed on the VM Fullnode and VM Lite. These two Virtual Machines (VM) will be explained in the data collection and laboratory section.

Table 1. Transparent and shielded addresses.

VM Fullnode		
Type	**Address**	**Private key**
Transparent	**t**1gxPPoGQuy6PT5QJFd C8wEjP7hUETG3Yrw	L2tKDay3FH3NUro……rXkj1HpJrhYkn1p
Shielded	**z**s1e4jvjsaft625y28jtcm9v yehak7u0jzlyqsr0y43y308 y8ntdvvev37g7maq37seylj kxtsflfu	secret-extended-key- main1qw2hpuseqqqqpqx29e720k770mervrdpnrggh 8hu8g6t4k9yxn…….yvl0srglpzv6p8hcgw340q
VM Lite		
Transparent	**t**1QbX4ec2KBjAhyN1Q M1gqqHGtF7P66iz6h	L46vxEYZLpoK3bP64e…..CF3J4m7Tg3ihKB
Shielded	**z**s1zr0v2y48jqazu3rhjdnv 4msrx6wrfsk8xumnzyqxpt 5fhu9d4n3r8y5wdwsnu9f w5784g2n4jrt	secret-extended-key- main1q0j4frjlqqqqpqqyjtfv0f73my9u02lxxmyfp9s yzd56szktsqf2xue44y56gw6jtsec92jkrt6fnksmj… ……s6anq8uwksp8k25jwguwegpslf6zgl5mx26

The transparent address is 35 characters long, while its key is 52 characters long. Moreover, a transparent address works identically as a Bitcoin address, meaning that transactions between t-addr to t-addr, information such as sender address, recipient address, and amount, are public on the blockchain.

By contrast, a shielded address is 78 characters long while its private key is 302 characters long, starting with “secret-extended-key-”. Furthermore, shielded addresses are used in the type of transactions that use zero-knowledge proofs to allow transaction data to be encrypted but remain verifiable by network nodes. Meaning that information in the blockchain is not visible.

Likewise, when a transaction is sent from a shielded address to another shielded address, there is no trace about who was the sender, and to overcome this issue, an additional field called “memo field” was included in this type of transactions allowing the sender of the transaction to add a note that will be only visible on the recipient’s wallet [27].

Zcash has four types of transactions illustrated in Figure 1:

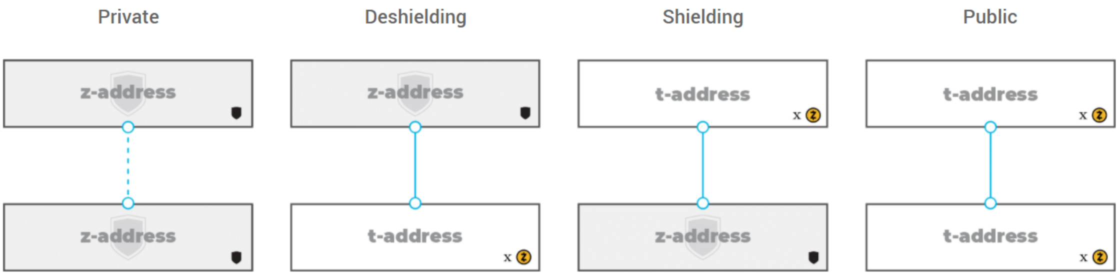


Figure 1. Zcash types of transactions [22]

In a Z-to-Z transaction, also called Private, the transaction is recorded in the blockchain, registering that it happened, but information such as sending address, recipient address, memo field and the amount is encrypted [22].

In a Z-to-T transaction or Deshielding, the sender is not revealed in the blockchain [22].

In a T-to-Z transaction or Shielding, the recipient is not revealed on the blockchain [22].

A T-to-T transaction, also called Public, as mentioned before, works like Bitcoin and is entirely public [22].

Figure 2 summarises the difference between these four transactions, and what is visible in the blockchain explorer is:

	Block explorer: Zcash					
		Sender	Recipient	TX	Memo	Timestamp
Private (Transaction information is not revealed on the blockchain)	z to z	zX5dlylkjadsY...	zgrkd5jlalkdsf...	0.45 ZEC	This is a message!	1/9/2021 10:10
Deshielding (sender is not revealed in the blockchain)	z to t	zDfd3g79mdf...	tyiOjfdmnusd...	1.09473 ZEC	Tere!	1/9/2021 10:14
Shielding (recipient is not revealed on the blockchain)	t to z	tF7u9Emnusd...	zhlkj8mn6nks...	2 ZEC	Hola!	1/9/2021 10:31
Public (Transaction information is revealed on the blockchain)	t to t	tAlskdfn7saw...	tk9ss36Hdfkz...	.0005 ZEC	-----	1/9/2021 10:50

Figure 2. The information is shown in the blockchain according to the transaction type [22]

As can be seen in Figure 2, the private transaction only shows the timestamp as information of the transaction.

Additionally, the Shielded address has a corresponding viewing key, which allows the owners to disclose details regarding incoming transactions but not details about the sender address unless the Memo Field contains something that makes it identifiable.

2.5 Dash

Dash was initially launched in 2014 with the name of xcoin, then darkcoin and finally became Dash in 2015 [28]. “Dash focuses primarily on privacy and transaction speeds. Consequently, Dash transactions are near-instantaneous and close to impossible to trace” [29].

Dash operates with two principal components in the network. The first one is the miners that have the same tasks as the ones in Bitcoins, and the second component is the “master nodes”, which have advanced functions such as the governance in the blockchain, and they are responsible for executing the special transactions called Instant Send [28].

An Instant Send (IS) transaction uses the protocol with the same name, and it bypasses the miners to eliminate the waiting time of the normal transaction and instead uses the master nodes to validate the transaction [29]. On the other hand, a Private Send (PS) transaction is the kind of transaction that offers anonymity and privacy to the users by applying the technique called coin mixing [28].

Coin mixing is a technique that “consists of taking a certain amount of coins and mixing them with others. Thus, it seeks to completely hide what funds come from which direction and to which direction they go. The process is also reinforced by the number of mixtures that are made since the greater the number of mixtures the safer and anonymous the process is” [30].

Table 2 shows the addresses used during the analysis. Dash addresses are 34 characters long and begin with an uppercase X.

Table 2. Dash addresses used in the cases.

Address	Private Key
VMfullnode	
XtaXbvRWspeVDE1YPA4z93Fa2JvubBdS4J	XDGm6zn3P7…..iqMendz74Exo2tDz8q
VMlite	
Xy33PKeqtootPQ591v5VDSGwNQzdm9MZxQ	XCHiBZMCRK…Yj9HCk7ZwBEb2bPr
XgWKMkASgroRmi5UrbfMb2Pb2ZV6KouKyi	XGuZDowfH4…y6hbNs1M3LLADWar

2.6 Cryptocurrency wallet

A cryptocurrency wallet is a software program that allows the users to interact with the blockchain to control the balance of their cryptocurrencies and to send or receive cryptocurrencies [31]. The process of sending and receiving cryptocurrencies is called transaction, and this is possible thanks to the capacity of the wallets to store private and public keys [31]. An address (public key) is what can be shared with anyone, and it is used to receive cryptocurrency, while its corresponding private key is used to send cryptocurrency and must not be shared with anyone [32].

"Cryptocurrency wallets can be divided into two major categories, and they are cold wallets and hot wallets. The difference between the two of them is that for hot wallets is necessary an internet connection and for cold wallets not" [33]. To this study, the focus will be on the hot wallet category, especially in the wallet for the desktop version.

When a software wallet is executed for the first time, this will randomly generate a list of 12-24 words. Those words are called the seed phrase or mnemonic phrase, which is unique in each wallet [32]. This seed is used to restore the wallet in case necessary. So, when the disk of the computer fails, or some other thing happens to the computer where the wallet is installed, the user can restore it using this mnemonic phrase typing it in the same order it was generated.

"A hierarchical deterministic wallet (HD wallet) is a wallet that generates all its keys and addresses from a single source. Deterministic means the keys and addresses are always generated in the same way every time, and Hierarchical means the keys and addresses can be organised into a tree" [34]. The source of the HD wallet is the mnemonic phrase which, after a cryptographic process it will create a master key pair: an extended private key (xprv) and an extended public key (xpub). From the xprv it can derive child private keys with their respective public keys (address), and from the xpub it is possible to derive the child public keys. In any case, it is not recommended to share the xprv and xpub because it can allow someone else to take control over the wallet funds.

2.7 Literature Review

The increasing use of privacy-oriented cryptocurrencies has been in part thanks to the acceptance on the darknet marketplaces due to the benefits these privacy-oriented cryptocurrencies offer in terms of privacy and anonymity [8]. Moreover, these cryptocurrencies use a non-public or private blockchain that may undermine law enforcement agencies tasks such as the anti-money laundering checks to comply with the Banks Secret Act requirements [8]. However, Bitcoin still remains the most traded and popular payment method in darknet marketplaces thanks to its wide adoption and ease of use [5].

Given the fact that Bitcoin is the most widely used and popular cryptocurrency among users, many studies on forensic analysis about Bitcoin and its blockchain have been carried out [12]–[18]. However, few studies related to the analysis of these privacy-oriented cryptocurrencies were found. One of these studies is oriented in evaluating the security of the wallet application for mobile devices such as Mycelium, Coinomiand and BRD [35], while the other one is focused on the forensic analysis of Monero and Verge [36].

Despite being one of the oldest documents that makes a forensic analysis on Bitcoin wallet and mining software, the study conducted by Michael Dorian states that "Building a case involving the forensic artefacts of Bitcoin is more difficult than the average case due to the technology that Nakamoto implemented to keep the transactions pseudonymous" [12]. Likewise, the author concludes the study by mentioning that the memory analysis has returned lots of information regarding the transaction history, addresses and Bitcoin application installed on the system where the tests were carried out [12].

On the other hand, taking further steps into the forensic investigation to determine if bitcoin transaction can be de-anonymised by analysing the Blockchain in combination with machine learning techniques and social media technology to identify illicit transaction [13] was carried out, concluding that it is possible to create a profile behaviour of Bitcoin addresses and illegal transactions.

A framework called Forensic Analysis of Bitcoin Transaction (FATB) was introduced by Yan Wu, Anthony Luo and Dianxiang Xu. "FATB formalises the clues of a given case as transaction patterns defined over a comprehensive set of features regarding transactions, addresses, and transaction flows. To facilitate pattern matching, FABT converts the bitcoin transaction data into a formal model, called Bitcoin Transaction Net (BTN), which is an extended form of safe Petri nets" [14], [15].

Different approaches and methodologies have been developed to overcome the analysis of Bitcoins transactions. Regardless of the cryptocurrency which is being analysed, as most of the altcoins are a fork of Bitcoin, the methods can be the same but will slightly change in terms of tools and how the investigator is employing them during the case. In like manner, the three main methods are Network analysis, which is highly technical and experimental; Transactional analysis, which can be very straightforward but easily defeated by services that use techniques such as Coin Join; and finally the Wallet analysis that is supported by the expert witness testimony [16].

Evaluation of cryptocurrency wallets has been done from the point of view of how secure these are and dividing the wallets in those that need an internet connection from those that do not [33]. Some of these studies performed over the cryptocurrency wallets are a sort of hacking proof since they are performing brute force attacks to guess the seed phrase of the wallet [37]. However, others have a broader scope making a manual inspection about what permissions the applications require, static analysis of the code and how transactions are propagated from the application to the blockchain [35].

In terms of the forensic analysis of wallet applications, the results obtained from the Bitcoin Electrum and Bitcoin Core based on the methodology followed by the authors that focus on digital evidence present in memory [18], reveals significant findings that can be identifiable thanks to the fingerprints recollected during the analysis. On the other hand, with the focus, this time not only on memory processes but also on the disk and network analysis, Monero and Verge wallet analysis [36] shows similarities in the findings obtained in the Bitcoin Electrum and Bitcoin Core, despite being privacy-oriented cryptocurrencies, meaning that the software can have the same behaviour but not the protocol.

Diverse frameworks for digital forensic investigation have been developed with different approaches [38] [39]. However, from the collected documents in the literature review, the forensic studies have utilised the Investigation Process for Digital Forensic Science proposed by the Digital Forensic Research Workshop (DFRWS) [12], the methodology proposed by Cassey, E. [36] and the methodology proposed by McKemmish, R [40] [41].

Since this study is intended to be a guide for law enforcement agencies, the forensic framework that goes along with this purpose is the one proposed by McKemmish, R. [42]. McKemmish not only addresses the technical side of the forensic investigation but also mentions how the evidence must be treated and presented in order to be valid in a court of law.

Also, some studies not related to cryptocurrencies but inside the scope of the digital forensic analysis have been considered in the literature review with the sole purpose to serve as a guide. One of the considerations to select these documents has been the McKemmish framework applied for the Forensics Analysis of an On-line Game over Steam Platform [40] and Windows Instant Messaging App Forensics: Facebook and Skype as Case Studies [41]. These two forensic cases are an example of how the evidence is collected, processed, and presented under the selected forensic framework.

Finally, with cybercriminals embracing more privacy-oriented cryptocurrencies due to the built-in anonymity and privacy features that make them more challenging to track than Bitcoin, it is essential to know what forensic artefacts can be obtained from the analysed devices that in combination with information available in the blockchain, can help the investigators to link who is trying to hide behind these illicit transactions.

3 Methodology and Research Design

The following section will describe the used methodology, how the data was collected, and the case studies proposed. Likewise, after the mentioned steps, it is expected to obtain as much information as possible such as details of the transactions, contact list, backups, private keys, etc. The obtained information attempts to assist the investigator in complementing the gathered information from the blockchain to finally create a bigger picture of the case.

3.1 Method and Forensic Framework

Experimental research methods were used during the development of this study. Moreover, it was conducted in a controlled environment composed of virtual machines. For achieving this purpose and following the example from previous works related to the forensic analysis of cryptocurrencies, the McKemmish forensic framework was used to support the investigation. The mentioned framework consists of the following steps:

3.1.1 Identification of Digital Evidence

The investigator has to know what evidence is present, how it is stored and where it is stored to determine what processes need to be employed in order to proceed with its recovery [42]. Moreover, the investigator must be capable of identifying the type of information stored in the device to be analysed with the purpose to select the adequate technology to extract the evidence [42].

3.1.2 Preservation of Digital Evidence

Digital evidence must be handled carefully with the purpose to preserve its integrity since there exists the possibility that it can be presented in a court of law [42]. However, the alteration of the digital evidence may be inevitable; in such a case, the investigator should be able to explain the reasons for the alteration [42].

3.1.3 Analysis of Digital Evidence

In this step, the investigator extracts, processes, and interprets the data to make it understandable and readable by people that have no previous knowledge or background in digital forensics [42].

3.1.4 Presentation of Digital Evidence

"Involves the actual presentation in a court of law. This includes the manner of presentation, the expertise and qualifications of the presenter and the credibility of the processes employed to produce the evidence being tendered" [42].

3.2 Data Collection and Laboratory

The software used in the creation of the laboratory setup and the workflow followed during the forensic data acquisition and interaction between the wallets is described as follows.

3.2.1 Laboratory Setup

The analysis started by running two VMs hosted in VirtualBox. Both VMs had Microsoft Windows 10 operative system, where the wallet application was installed. Also, an iPhone was included in the setup that supported the creation of cases for both cryptocurrency wallets, but this device is out of the scope of the forensic analysis.

The use of a virtual environments for the experiment responds to the need of making acquisitions without the limitation that non-virtual machines have, for instance, taking snapshots of a clean installation of windows that can be reused as much as is needed.

This setup works in the same way for Zcash and Dash, but the interaction is only between the wallets of the same cryptocurrency, meaning that Zcash and Dash do not interact with each other.

The first VM was called VMfullnode, and during the first experiment, it installed the Zecwallet FullNode v0.0.24.0. Later a clean snapshot was restored to start with the second experiment that installed the Dash Core v0.16.1.1.

On the other hand, the second VM was called VMlite, and it had the Zecwallet Lite v1.4.2. installed during the first experiment. Later a clean snapshot was restored to start with the second experiment and installed the Dash Electrum v4.0.9.3.

3.2.2 Workflow Acquisition

A series of steps were followed to obtain the network, live and post-mortem forensic acquisition. Open-source tools such as FTK Imager v5.5.3, Volatility v2.6.1, Foremost v1.5.7, Bulk Extractor v1.6.0, TShark v3.4.3 and Wireshark v3.4.3 were used to support the forensic acquisition.

The network acquisition was made by running TShark on the host machine applying filters such as network interface and the IP address of the VMs. Those filters captured the inbound and outbound network traffic related to the VMs. While the network traffic was captured, transactions from and to the wallet installed on the VMs were made. Once the transactions were finalised, previous confirmation that the funds were added or debited from the wallet, the network acquisition was stopped, and the evidence was saved with .pcapng format. Later these files were analysed using Wireshark.

Before the live acquisition, the VM was restarted and turned off to start from a fresh RAM state, and the network interface was disconnected. The process started inserting a 2GB USB drive with FTK Imager and another USB external storage of 2 TB capacity, where the memory RAM dump files were saved. The analysis of the files was divided into two parts. The first part, called Structured analysis that was done using tools such as Volatility and Foremost, and the second part, called Unstructured analysis that was performed using Linux commands such as strings, grep supported by keywords [18].

To proceed with the post-mortem acquisition, the VM was turned off, and the VirtualBox command *clonemedium* was executed on the VDI file, which represents the disk of the VM. The execution of this command allows duplicating a virtual disk in raw format that was analysed later using FTK Imager and Bulk Extractor.

3.3 Case Studies

The user’s interaction with the wallet applications by sending and receiving money and exploring additional options with the purpose to generate the necessary data to proceed with the forensic acquisition and subsequent analysis are described in the following case studies.

3.3.1 Zcash Cases

The case studies have been divided into two parts. The first part starts with the installation of the Zecwallet Fullnode on the VMfullnode and the interaction with the iPhone. As a result of these actions, 8 cases were produced and described in Table 3.

Table 3. Zecwallet Fullnode cases.

Case Studies	**Description**
Case 1	The user downloads and installs the wallet application. Then executes it and waits until the Blockchain is downloaded and synchronised.
Case 2	The user receives ZEC from iPhone to VMfullnode through the **Private** transaction. Transaction data: Recipient address.: zs1e4j…sflfu Amount: 0.00000001 Memo Field: From Z i to Z vm. JM
Case 3	The user receives ZEC from iPhone to VMfullnode through the **Deshielding** transaction. Transaction data: Recipient address.: t1gxP...G3Yrw Amount: 0.001
Case 4	The user sends ZEC from VMfullnode to iPhone using the **Private** transaction. Transaction data: Recipient address.: zs13t...670mu Amount: 0.0006 Memo Field: from Z vm to Z iphone.
Case 5	The user sends ZEC from VMfullnode to iPhone using the **Deshielding** transaction. Transaction data: Recipient address.: t1dv9...ospqa

	Amount: 0.00007
Case 6	The user sends ZEC from VMfullnode to iPhone using the **Shielding** transaction. Transaction data: Recipient address.: zs13t...670mu Amount: 0.0002 Memo Field: from T vm to Z iphone.
Case 7	The user sends ZEC from VMfullnode to iPhone using the **Public** transaction. Transaction data: Recipient address.: t1dv9...ospqa Amount: 0.00069
Case 8	This case aims to identify if the user has executed additional options such as the available from the CLI and documented in the official repository of Zcash [43]. Even though this action could be considered for an advanced user since previous modifications to the default wallet configuration have to be done, the information that they provide is valuable, and if the investigator can obtain it, it would be helpful as a part of the investigation case. The command to be executed as part of this case is the "*z_exportwallet*". This command exports into a file the list of all transparent and shielded private keys with their associated public addresses; moreover, the HD seed is exported to this file.

The second part encompasses the installation of the Zecwallet Lite on the VMlite and the interaction with the VMfullnode and the iPhone. As a result of these actions, 5 cases were produced and described in Table 4. An important aspect to point out is that Zecwallet Lite sends transactions in shielded mode by default. Meaning that the user cannot select a transparent address as the sending address when doing the transaction. Figure 5 explains this limitation graphically for a better understanding.

Table 4. Zecwallet Lite cases.

Case Studies	Description
Case 1	The user downloads and installs the wallet application. Then opens the application, and this one shows the 24-word mnemonic phrase that is automatically generated.
Case 2	The user receives ZEC from the VMfullnode to the VMlite through **Public** and **Private** transactions. Later, the user encrypts the wallet using the password "*arribaperu*". Transaction data from the first and second transaction: Recipient address.: t1QbX...6iz6h Amount: 0.2499 Recipient address.: zs1zr...n4jrt Amount: 0.24991
Case 3	The user receives ZEC from the VMfullnode to the VMlite through **Shielding** and **Deshielding** transactions. Additionally, the user adds a transparent and a shielded address to the Address Book of the wallet application. Transaction data from the first and second transaction: Recipient address.: zs1zr...n4jrt Amount: 0.0999 Recipient address.: t1QbX...6iz6h Amount: 0.09

Case 4	The user inputs the wallet's password to send ZEC from the VMlite to the iPhone using the **Private** transaction. Transaction data: Recipient address.: zs13t...670mu Amount: 0.344755 Likewise, the user exports the history transactions that are saved in CSV format and saved on the Desktop.
Case 5	The user inputs the wallet's password to send ZEC from VMlite to iPhone using the **Deshielding** transaction. Transaction data: Recipient address.: t1dv9...ospqa Amount: 0.344755 It also executes the option "Export All Private Keys" that shows all the addresses with their corresponding private keys, but these are not saved in the disk.

Unlike the Zecwallet Fullnode case studies, the Zecwallet Lite has in total 5 case studies but with the same number of transactions. As it can be seen in Table 4, case 2 and 3 are composed of two transactions, each one including the transactions in case 4 and 5 give a total of 6 transactions. On the other hand, case 8 from Table 3 could not be reproduced in the Lite version since this one does not have the CLI option available.

As it was explained before, the case studies were divided into two parts. The first part started when the user installed the wallet application in the VMfullnode, and once installed, this one downloaded and synchronised with the blockchain. This action took around 7 hours, and that is why no network acquisition was performed for Case 1. Later the user initiated the wallet application, created the first shielded address to finally close the application and the analyst proceeded with the memory acquisition and disk acquisition. Case 2 and 3 initiated when the analyst started capturing the network traffic, and then the user opened the wallet applications to receive the ZEC from the first and second transactions. When the transactions received the confirmations from the blockchain, the application was closed, and the analyst stopped the network acquisition and started making the memory and disk acquisition, respectively, to finalise Case 2 and 3. From Case 4 to 7, the analyst began the acquisition of network traffic, and the user ran the application to send ZEC to the iPhone wallet; and after these four transactions had been confirmed in the blockchain, the analyst stopped the network acquisition and started the memory and disk acquisition respectively.

On the other hand, the second part initiated when the analyst started the network acquisition, and the user installed the wallet application in the VMlite and waited until this synchronised in around 5 minutes. Once the synchronisation finisheed, the analyst stopped the network acquisition and proceeded with the memory and disk acquisition, respectively, to finalise with Case 1. From Case 2 to 3, the analyst started capturing the network traffic, and the user executed the wallet application to receive the ZEC from the first four transactions coming from the VMfullnode. Once received and confirmed these transactions, the analyst stopped the network acquisitions and proceeded with the memory and disk acquisition, respectively, ending Cases 2 and 3. Case 4 and 5 started when the analyst captured the network traffic, and the user ran the applications to send ZEC to the iPhone, and when the transactions have been confirmed, the analyst stops the network acquisition and proceeds the acquisitions of memory and disk, respectively, ending Case 4 and 5.

Figure 3 shows the workflow where the three wallet applications interacted between each other and where the forensic evidence was captured.

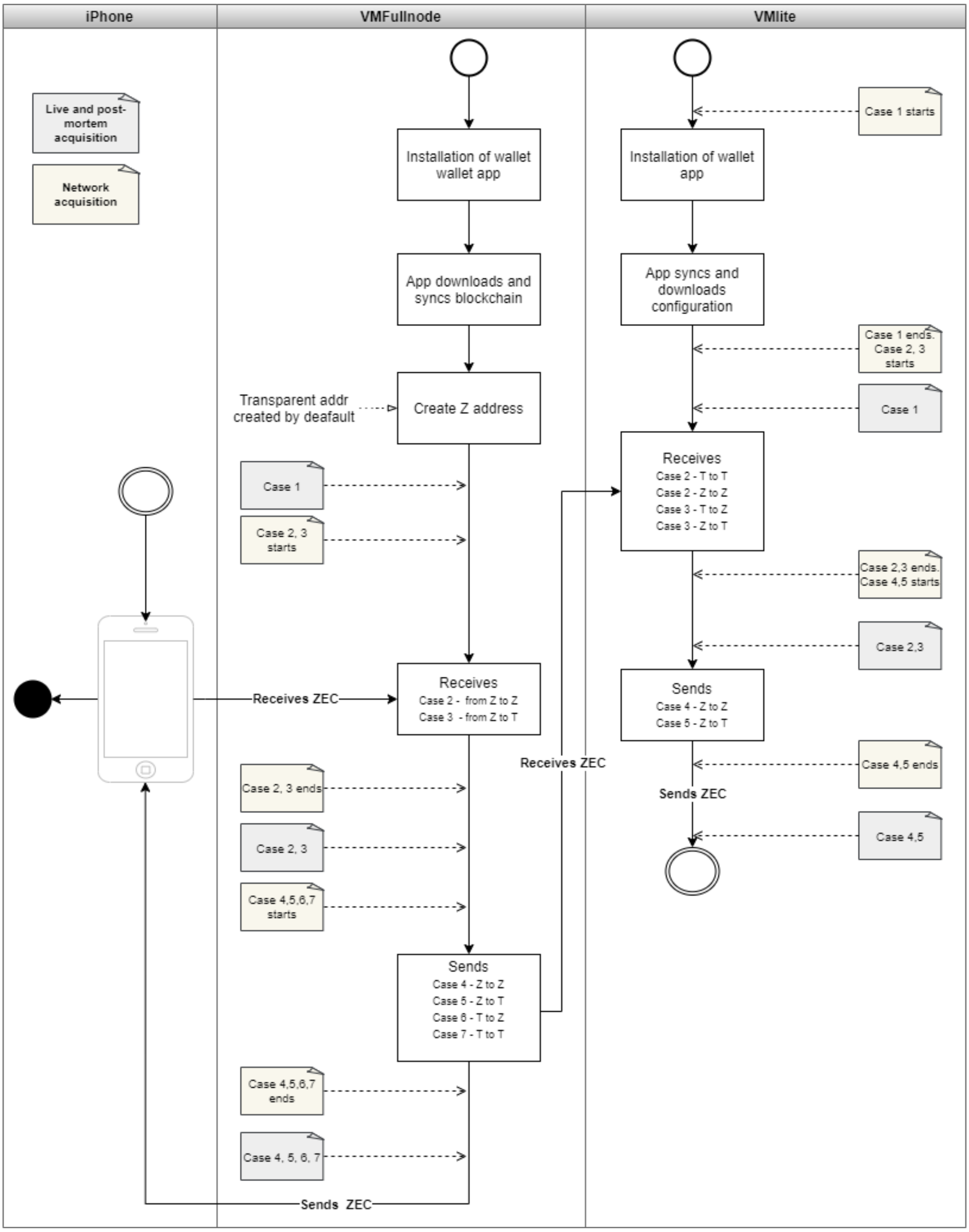


Figure 3. Interaction between Zcash wallets and data acquisition cases.

Finalised these stages, all files were hashed using the SHA256 algorithm, and the results are shown in Appendix B.

3.3.2 Dash Cases

The case studies were composed of two parts. The first part started with the installation of the Dash Core on the VMfullnode and the interaction with the iPhone. As a result of these actions, 6 cases were produced and explained in Table 5.

Table 5. Dash Core cases.

Case Studies	Description
Case 1	The user downloads and installs the wallet application until this one downloads and synchronises with the blockchain.
Case 2	The user receives DASH from the iPhone to the VMfullnode through an **Instant Send** transaction. Transaction data: Recipient address: XtaXb...BdS4J Amount: 0.646905
Case 3	The user sends DASH from the VMfullnode to the iPhone using **Instant Send** transaction. Transaction data: Recipient address: Xr2D3...r9Wtn Amount: 0.32 Label: iPhone addr Likewise, the user encrypts the wallet with the password *"4rr1b4p3ru"* and makes a backup saving the file *BKwallet.dat* on the desktop.
Case 4	The user sends DASH from the VMfullnode to the iPhone using a **Private Send** transaction. To proceed with the transaction, first, the user has to "Start Mixing" the available funds he/she has in the wallet to obtain private send available coins. Finally, the user inputs the password used in Case 3 to spend the desired amount of DASH. Transaction data:

	Recipient address: Xogci...hZp25 Amount: 0.20700207 Label: iPhone addr. PS
Case 5	By default, Dash Core does not have the mnemonic phrase enabled, which means that the user has to make backups of the wallet.dat file to restore it in case it is necessary. To enable the mnemonic phrase to restore the wallet through this method later, the user needs to activate it manually, executing a couple of commands that are documented in the official repository of DASH [44]. The commands to execute are "*dashhd.exe --usehd=1*" and "*dumphdinfo*" from the command prompt and the Dash console, respectively. Once the commands are executed, DASH mentions that the 24-word mnemonic phrase is stored in plaintext in the wallet.dat file [44]. So, this case aims to verify if the 24-word mnemonic phrase is recoverable as part of the study.
Case 6	The user encrypts the wallet created in Case 5 to verify if the mnemonic phrase is still present or not in plaintext when doing the backup of the wallet.dat file.

The second part encompasses the installation of the Dash Electrum on the VMlite and the interaction with the VMfullnode and the iPhone. As a result of these actions, 6 cases were produced and described in Table 6. It is good to mention that Dash Electrum, which is the wallet version for mobile devices, does not have the feature to make a private send transaction. Meaning that the user can send DASH only by instant send transactions.

Table 6. Dash Electrum cases.

Study Case	Description
Case 1	The user installs the application and selects the Tor [3] Proxy to be installed as an additional component. Next, the user creates the wallet file and names it *testttu_wallet*, then the wallet shows the 12-word mnemonic phrase and finally, the user encrypts the wallet with the password *“4rr1b4p3ru3”*.
Case 2	The user receives DASH from the VMfullnode to the VMlite through the **Instant Send** transaction. To see the transferred funds reflected on the wallet, the user needs to open it and input the wallet password that was entered in Case 1. Transaction data: Recipient address: Xy33P...9MZxQ Amount: 0.14999774 Label: From VMfull to VMlite. Cas1
Case 3	The user receives DASH from the VMfullnode to the VMlite through a **Private Send** transaction. To see the transferred funds reflected on the wallet, the user needs to open it and input the wallet password that was entered in Case 1. Transaction data: Recipient address: XgWKM...ouKyi Amount: 0.04999266 Label: Case3. from VMfull to VM lite

[3] ¨The Tor project is a non-profit organisation that conducts research and development into online privacy and anonymity. It is designed to stop people – including government agencies and corporations – learning your location or tracking your browsing habits¨ [45].

Case 4	The user sends DASH from the VMlite to the iPhone using the **Instant Send** transaction. To complete the transaction, the user needs to enter the password two times. The first one when he/she opens the wallet application, and the second one when he/she sends the funds. Transaction data: Recipient address: XgWKM...ouKyi Amount: 0.10000339 Label: Case 4. from vmlite to iPhone
Case 5	The user sends DASH from the VMlite to the iPhone using the **Private Send** transaction. To complete the transaction, the user needs to enter the password two times. The first one when he/she opens the wallet application, and the second one when he/she sends the funds. Likewise, the “Start Mixing” option needs to be activated to create available private send balance. Transaction data: Recipient address: XosGs...p6f4K Amount: 0.11100111 Label: Case 5. from vmlite to iphone. Private Send
Case 6	The user explores the different options that the wallet has, such as backup the wallet, the screen shows the mnemonic phrase, export the private keys, and execute commands from the embedded console of the wallet. These actions require the user to enter the password to be completed.

As previously explained, the DASH cases were also divided into two parts. The first part covered the interaction between the iPhone and the VMfullnodes. This interaction started with Case 1 when the user installed the wallet application, downloaded, and synchronised with the blockchain, which took around 5 hours to finalise; therefore, no network acquisition was performed in this step; later, the analyst started the memory and disk acquisition ending the Case 1. Case 2 began when the analyst initiated the network acquisition, and the user executed the applications to receive the first transaction from the iPhone. Once the transaction had received the confirmations from the blockchain, the analyst stopped the network acquisition and initiated the memory and disk acquisition, finishing Case 2. Case 3 and 4 started when the analyst captured the network traffic, then the user opened the application and proceeded to send DASH to the iPhone and finalised when the transactions had been confirmed, and then the analyst made the memory and disk acquisitions.

On the other side, the second part involved the interaction between the VMfullnode, VMlite and iPhone; and started with Case 1 when the analyst captured the network traffic, and the user installed the wallet application on the VMlite. Once the wallet was installed and synchronised, the analyst stopped the network acquisition to proceed with the memory and disk acquisition, respectively. Case 2 and 3 started when the analyst made the network acquisition, then the user opened the wallet application and received DASH from the first and second transaction coming from the VMfullnode. Once the transactions were confirmed, the analyst stopped the network acquisition and initiated the memory and disk acquisition, respectively, ending Case 2 and 3. Case 4 and 5 began when the analyst started making the network acquisition, then the user ran the wallet application and sent DASH to the iPhone in the third and fourth transaction. After the confirmation of the third and fourth transaction, the analyst stopped the network acquisition to proceed with the memory and disk acquisition ending Case 3 and 4.

The interaction between the three wallets and where the evidence was taken is illustrated in the flowchart in Figure 4.

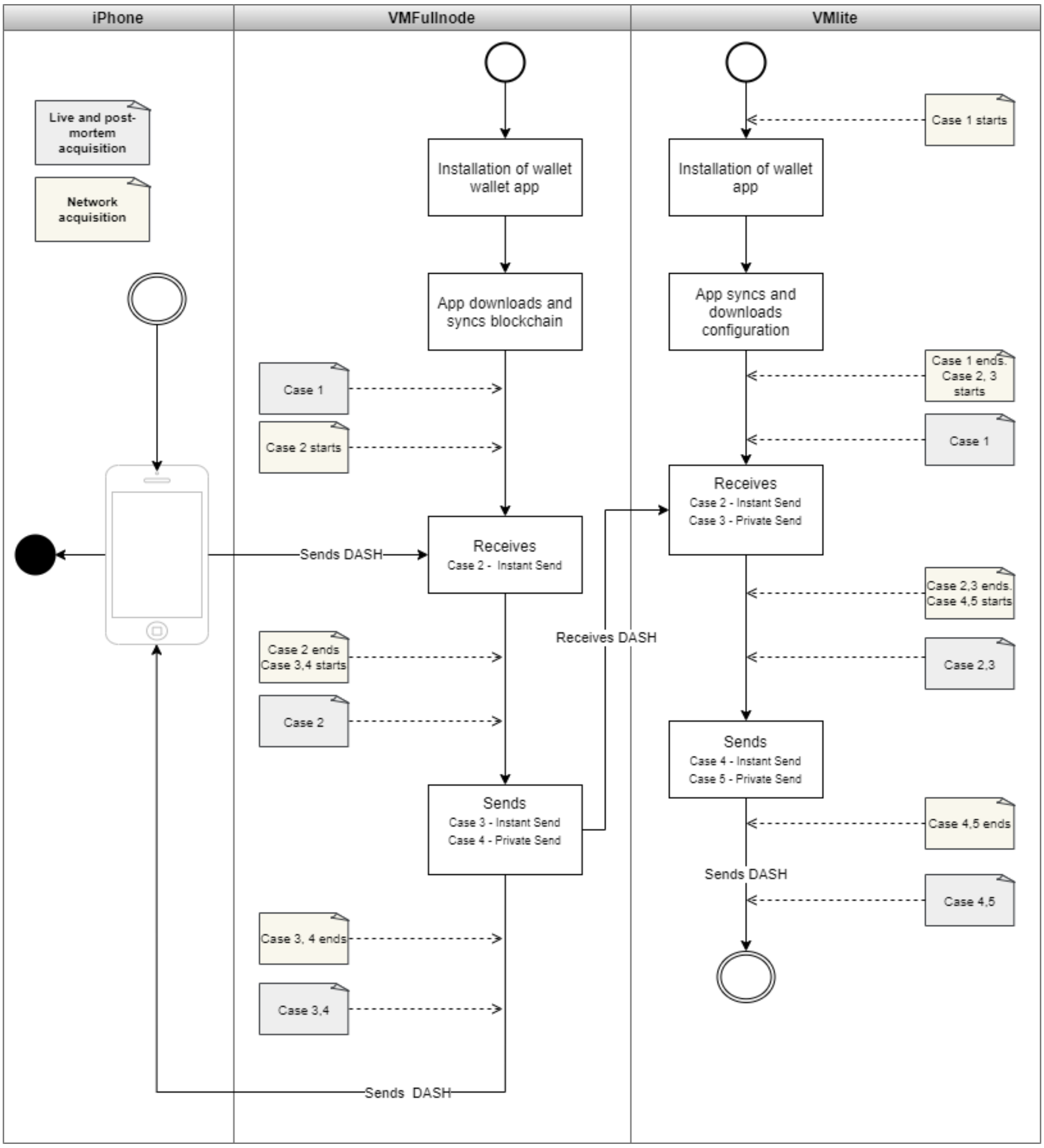


Figure 4. Interaction between Dash wallets and data acquisition cases.

Finalised these stages, all files were hashed using the SHA256 algorithm, and the results are shown in Appendix B.

Although the case studies presented in Table 3 and Table 4 are related to Zcash while Table 5 and Table 6 with Dash, the results could differ from one to another since the wallet applications are from different versions, as previously explained.

4 Analysis and Results

In this section of the document, the analysis and results performed over the network, live-acquisition and post-mortem acquisition files obtained during the case studies will be presented and explained.

4.1 Zecwallet Fullnode

In this section of the study, the full node version of the Zecwallet software will be analysed with the purpose to identify what forensic artefacts can be obtained. To have a better understanding of the direction of the transactions, Figure 5 depicts who was the sender and recipient from cases 2 to 7.

As it can be seen in the diagram, there exist two transactions marked in red; this is because the iPhone version does not support sending ZEC from a transparent address and only from a private address.

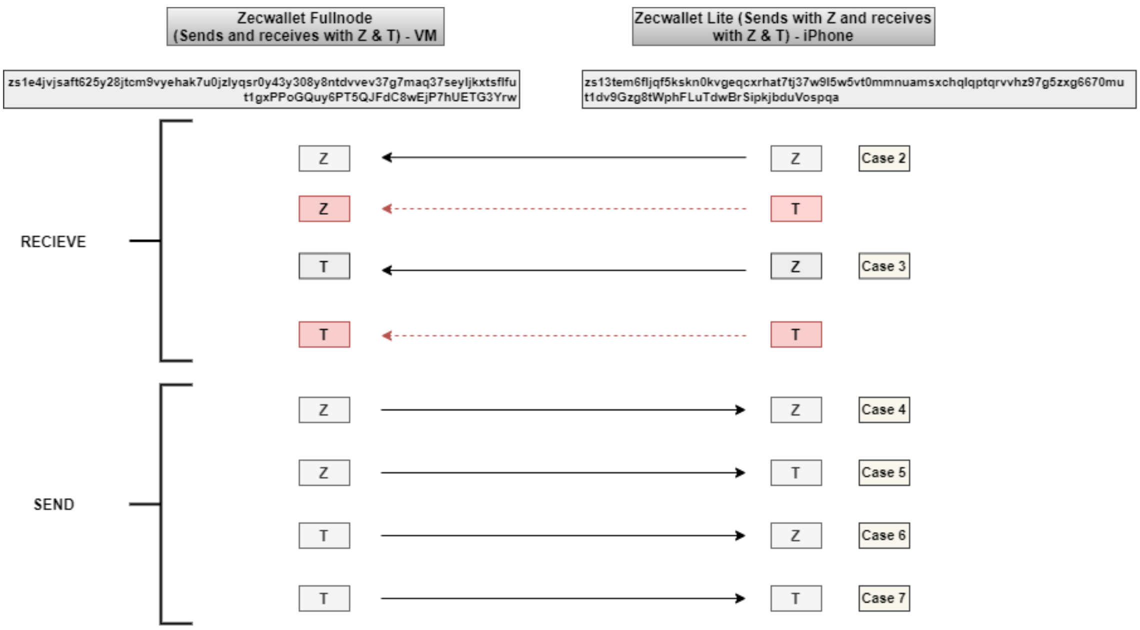


Figure 5. The direction of transactions between Zecwallet Fullnode and Zecwallet Lite.

Likewise, as was explained in the case studies section, case 1 and case 8 do not encompass transactions; that is why they are not present in the diagram.

4.1.1 Memory Images

4.1.1.1 Case 1

By default, the wallet application creates a transparent address while the shielded address was created manually. The structured analysis shows information about the Master File Table (MFT) record, illustrated in Figure 6, where file *zecwallet_transactions .csv* was created.

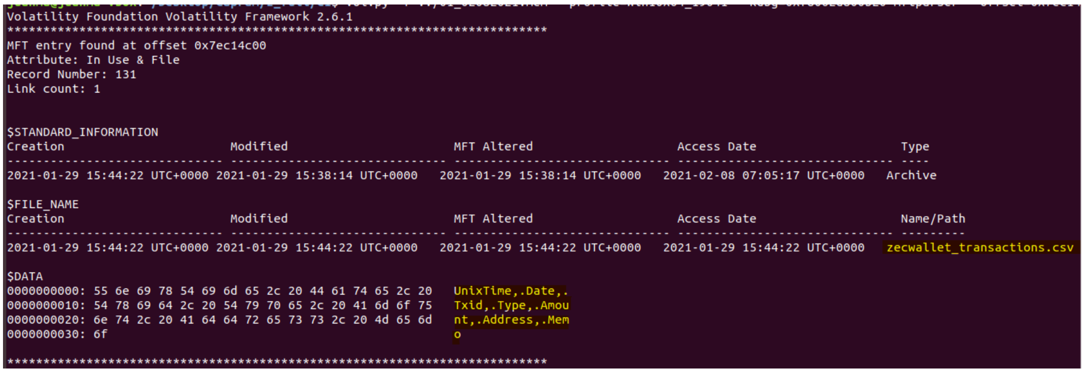


Figure 6. Z. Fullnode – Mem. analysis. Case 1. MFT record of the creation of CSV transactions file.

The file has no additional information than the headers. Since the user did not execute any action to create the file, it can be said that it is an automatic action performed by the application wallet.

The unstructured analysis shows information about the transparent and private addresses, illustrated in Figure 7, that was created by the application and the user respectively once the wallet was executed for the first time.

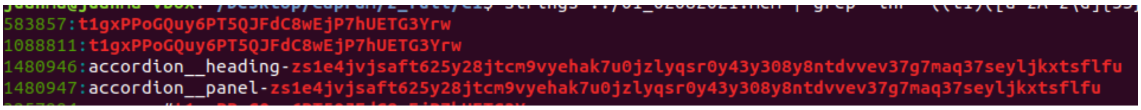


Figure 7. Z. Fullnode – Mem. analysis Case 1. T and Z addresses.

No private keys from these addresses were found.

4.1.1.2 Case 2

Structured analysis shows files downloaded by the wallet application as part of the blockchain synchronisation. No valuable artefacts were found during the analysis.

The unstructured analysis shows the incoming transaction in JSON format. Figure 8 illustrates the transaction ID

25ee0e307e63efb06f07c0574de8dabddb245fcbd6e252eaa4709746da31de32, the receiving shielded address *zs1e4jvjsaft625y28jtcm9vyehak7u0jzlyqsr0y43y308y8ntdvvev37g7maq37seyljkxtsflfu*, the transferred amount for the value of *0.00001*, and the memo field *46726f6d205a206920746f205a20766d2e204a4d* in hexadecimal value that converted to ASCII reveals the message sent from the iPhone that was "*From Z i to Z vm. JM*".

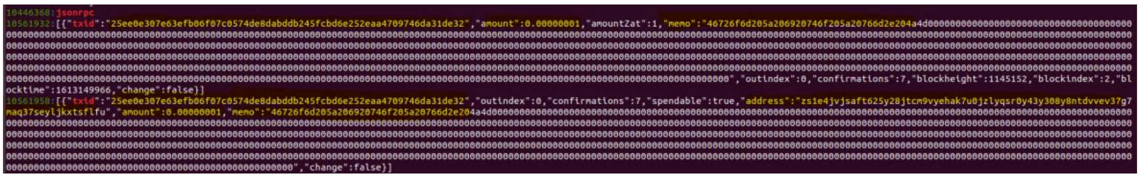


Figure 8. Z. Fullnode – Mem. analysis Case 2. First incoming transaction in JSON format.

The iPhone's screen illustrated in Figure 9 shows the original message that is shown in hexadecimal format in Figure 8.

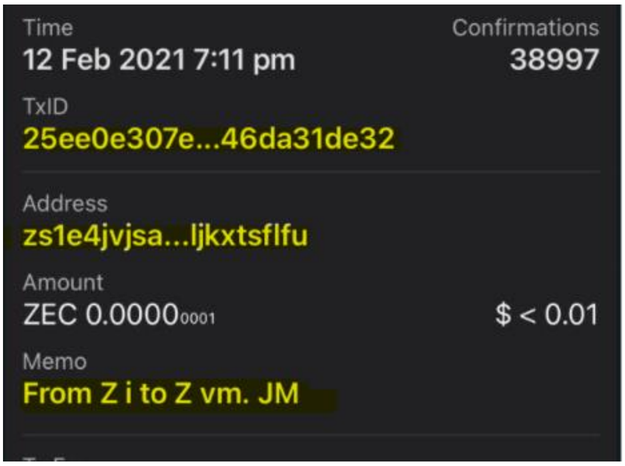


Figure 9. Z. Fullnode. Case 2. iPhone's screen from the first incoming transaction.

No private keys were found nor the sending address.

4.1.1.3 Case 3

The structured analysis performed over this memory dump file does not show valuable information. However, the unstructured analysis shows the incoming transaction in JSON format. This includes the transaction ID *25bc98a33f1c33d81ed3bed427aeecb211605cb95ef36288f64a6bf538efeb35*, destination transparent address *t1gxPPoGQuy6PT5QJFdC8wEjP7hUETG3Yrw*, the amount for the value of 0.001, and the timestamp 1613161378 in UNIX format.

Figure 10. Z. Fullnode – Mem. Analysis Case 3. Second incoming transaction in JSON format.

No private keys were found during the analysis.

4.1.1.4 Case 4

The structured analysis shows the MFT record of the creation of the file AddressBook.json, whose content has the label “*ZiPhone*” and “*TiPhone*” given by the user with its corresponding shielded and transparent addresses, respectively. This result is illustrated in Figure 11.

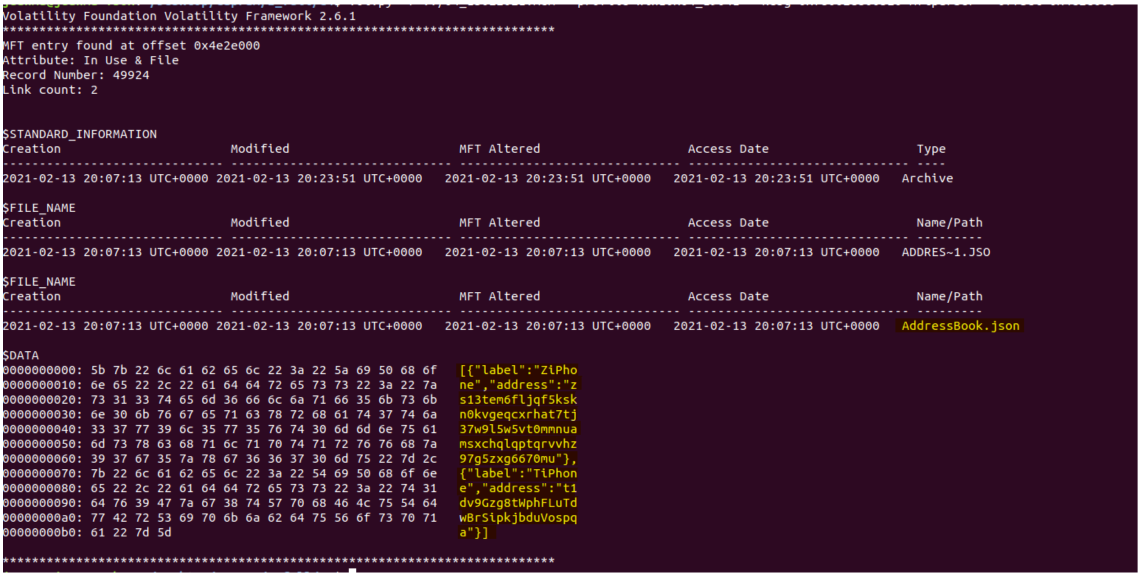


Figure 11. Z. Fullnode – Mem. analysis Case 4. MFT record of AddressBook.json file.

The unstructured analysis shows information about the incoming transaction in JSON format. Figure 12 depicts the transaction ID *43baa44e9f1335a15e5c5412584b2e001def74d94a76ddcc30b22fee15f79289*, the sending shielded address *zs1e4jvjsaft625y28jtcm9vyehak7u0jzlyqsr0y43y308y8ntdvvev37g7maq37seyljkxtsflfu*, the amount 0.00007 and the memo field in hexadecimal format.

Figure 12. Z. Fullnode – Mem. analysis Case 4. Third outgoing transaction in JSON format.

When the content of the memo field was decoded, it did show random characters and not the original message “*from Z vm to Z iphone*” that was included when the transaction was made. This is normal behaviour, and the message will be shown to the owner of the recipient address. For instance, Figure 13 shows what the owner of the recipient address sees:

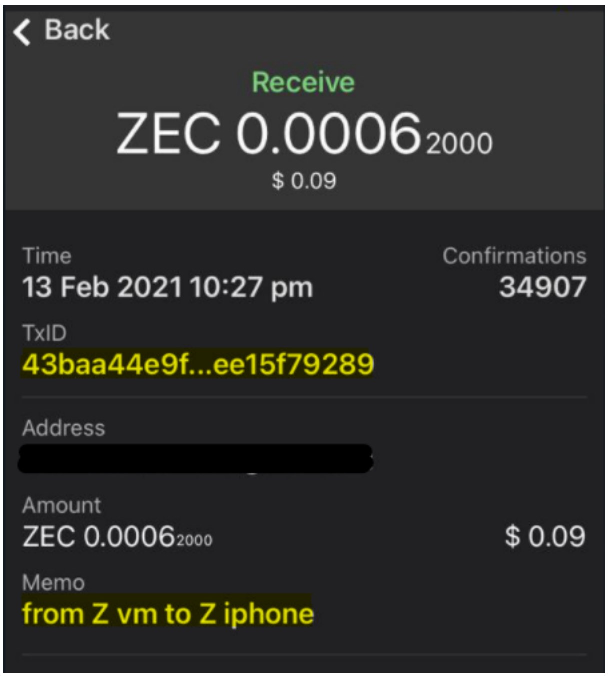


Figure 13. Z. Fullnode. Case 4. iPhone's screen after receiving ZEC from the third outgoing transaction showing the message included in the memo field.

The results of the unstructured analysis also show the content of the AddressBook.json file that was presented as part of the structured analysis and is illustrated in Figure 14.

Figure 14. Z. Fullnode – Mem. analysis Case 4. Content of AddressBook.json file.

No private keys were present during the analysis.

4.1.1.5 Case 5

The structured analysis does not show relevant information. Unlike the previous cases, the unstructured analysis does not show the transaction in JSON format. Only the transaction ID *b48591f1cabd46509a66b937fe0b7905085da5a882cb343f863604d8464c28bf* as can be seen in Figure 15.

```
7863581:Feb 14 16:26:15.212  INFO main: AddToWallet b48591f1cabd46509a66b937fe0b7905085da5a882cb343f863604d8464c28bf  update
13959753:b48591f1cabd46509a66b937fe0b7905085da5a882cb343f863604d8464c28bf
18977161:CTransaction(hash=b48591f1ca, ver=4, fOverwintered=1, nVersionGroupId=892f2085, vin.size=0, vout.size=1, nLockTime=0, nExpiryHeight=1147351, valueBalance=8001, vShieldedSpend.size=3, vShieldedOutput.size=0)
18977163:Feb 14 16:24:25.565  INFO main: AddToWallet b48591f1cabd46509a66b937fe0b7905085da5a882cb343f863604d8464c28bf  new
18977164:Feb 14 16:24:25.596  INFO main: Relaying wtx b48591f1cabd46509a66b937fe0b7905085da5a882cb343f863604d8464c28bf
18977165:Feb 14 16:24:25.596  INFO main: opid-4f9c9621-c82f-43f9-b10e-e4e7a6220e6a: z_sendmany finished (status=success, txid=b48591f1cabd46509a66b937fe0b7905085da5a882cb343f863604d8464c28bf)
18977168:Feb 14 16:24:26.213  INFO main: AddToWallet b48591f1cabd46509a66b937fe0b7905085da5a882cb343f863604d8464c28bf
18977176:Feb 14 16:26:15.212  INFO main: AddToWallet b48591f1cabd46509a66b937fe0b7905085da5a882cb343f863604d8464c28bf  update
32096466:b48591f1cabd46509a66b937fe0b7905085da5a882cb343f863604d8464c28bf
36519839:Feb 14 16:24:25.596  INFO main: opid-4f9c9621-c82f-43f9-b10e-e4e7a6220e6a: z_sendmany finished (status=success, txid=b48591f1cabd46509a66b937fe0b7905085da5a882cb343f863604d8464c28bf)
```

Figure 15. Z. Fullnode – Mem. analysis Case 5. Transaction ID of the fourth outgoing transaction.

No private keys were found during the analysis.

4.1.1.6 Case 6

The structured analysis does not show relevant information. The unstructured analysis shows evidence of the transaction in JSON format. The Figure 16 illustrates the transaction ID *f011ca4db4810b61c4e5beee53bf4d2938f486a7cc84639a94525f6c7edef107*, the amount for 0.0002, the fee for 0.00001, and the timestamp 1613322189 in UNIX format.


```
walletconflicts":[],"time":1613161378,"timereceived":1613161378,"vjoinsplit":[],"size":2407},{"account":"","category":"send","amount":-
0.00020000,"amountZat":-20000,"vout":1,"fee":-0.00001000,"confirmations":5,"blockhash":"0000000001a3126a8b47685dadec4bcbe807c191b5e6b87
e2b177cf9fcd678c3","blockindex":3,"blocktime":1613322252,"expiryheight":1147472,"status":"mined","txid":"f011ca4db4810b61c4e5beee53bf4d
2938f486a7cc84639a94525f6c7edef107","walletconflicts":[],"time":1613322189,"timereceived":1613322189,"vjoinsplit":[],"size":1222}],"err
or":null,"id":"curltest"}
```


Figure 16. Z. Fullnode– Mem. analysis Case 6. Fifth outgoing transaction in JSON format.

No private keys were found during the analysis.

4.1.1.7 Case 7

The structured analysis does not show relevant information. The unstructured analysis shows evidence of the transaction in JSON format. Figure 17 shows the fee 0.00001, the amount 0.00006, the recipient transparent address *t1dv9Gzg8tWphFLuTdwBrSipkjbduVospqa*, and the transaction ID *441479f39c59ec4e171bd6f952d238fc60d341670a46ad607f3438d27400c4a7*.

Figure 17. Z. Fullnode – Mem. analysis Case 7. Sixth outgoing transaction in JSON format.

No private keys were found during the analysis.

4.1.1.8 Case 8

The structured analysis shows on the MFT record the content of the *zcash.conf* file. The details are illustrated in Figure 18 and display the additional parameter *exportdir* required to execute the command *z_exportwallet*.

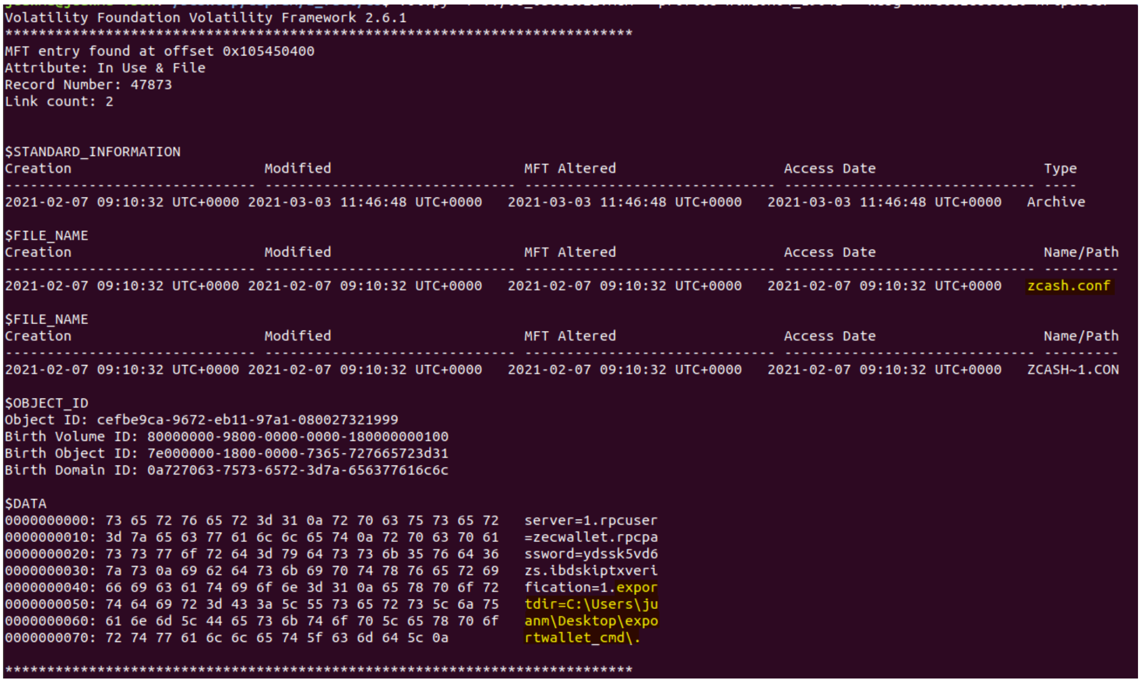


Figure 18. Z. Fullnode– Mem. analysis Case 8. MFT record displaying the content of zcash.conf file.

Besides, Figure 19 shows evidence of the access to the file *zcash-cli.exe* that allows the execution of the CLI commands. This finding was not present in the previous cases, which is helpful during an investigation giving clues to the investigator that the wallet owner has executed commands using the CLI option.

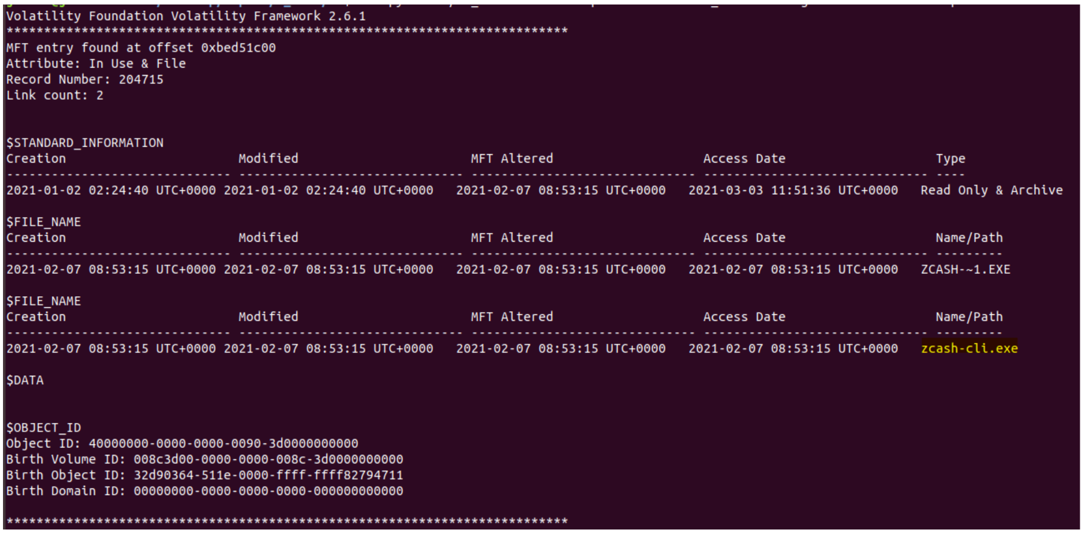


Figure 19. Z. Fullnode – Mem. analysis Case 8. MFT record displaying the modification/access to the zcash-cli.exe file.

As part of the unstructured analysis, it is possible to observe the private keys for both transparent and shielded addresses. This is the result of the execution of the *z_exportwallet* command. For instance, Figure 20 shows the transparent addresses highlighted in red and their corresponding private keys in yellow.

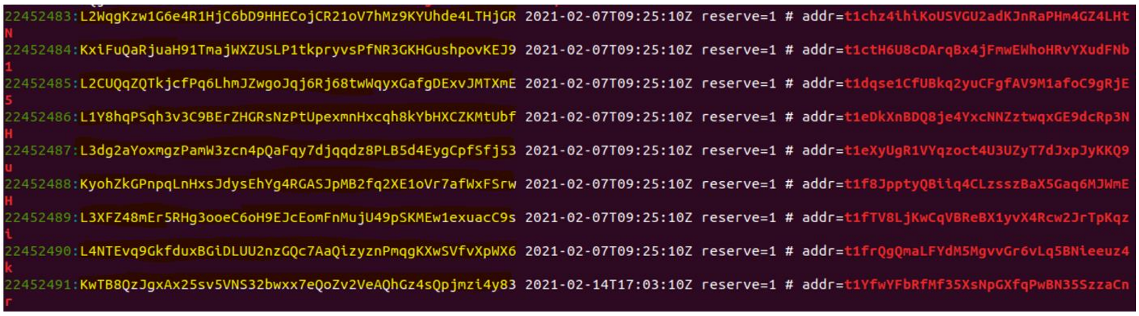


Figure 20. Z. Fullnode – Mem. analysis Case 8. Transparent addresses and their private keys.

Similar results but for shielded addresses are depicted in Figure 21. In this case, the format for the shielded private keys starts with *secret-extended-key-main1q*, and there are two records for the two addresses created from the wallet application.

Figure 21. Z. Fullnode – Mem. analysis Case 8. Shielded addresses and their private keys.

Even though it was possible to find the private keys, the HD seed was not found.

4.1.2 Disk Files

During the analysis of the raw images, the most relevant files containing interesting information were:

- The *debug.log* file.
- The *AddressBook.json* file.
- The *wallet.dat* file.
- The *zcash.conf* file

The above-mentioned files were created by the wallet application, and no user intervention was required.

The *debug.log* file contains general debug information about the application but also contains the transaction IDs of incoming and outgoing transactions. The IDs are illustrated in Figure 22.

```
Line 1188337: Feb 12 19:12:46.381  INFO main: AddToWallet 25ee0e307e63efb06f07c0574de8dabddb245fcbd6e252eaa4709746da31de32  new
Line 1189270: Feb 12 19:13:35.346  INFO main: AddToWallet 25ee0e307e63efb06f07c0574de8dabddb245fcbd6e252eaa4709746da31de32  update
Line 1189390: Feb 12 22:21:42.896  INFO Init: main: AddToWallet 25ee0e307e63efb06f07c0574de8dabddb245fcbd6e252eaa4709746da31de32  update
Line 1189579: Feb 12 22:22:58.285  INFO main: AddToWallet 25bc98a33f1c33d81ed3bed427aeecb211605cb95ef36288f64a6bf538efeb35  new
Line 1189581: Feb 12 22:23:25.278  INFO main: AddToWallet 25bc98a33f1c33d81ed3bed427aeecb211605cb95ef36288f64a6bf538efeb35  update
Line 1191107: Feb 13 22:13:44.272  INFO main: AddToWallet b9a4cce1c1b1ae84261c0687ab152a83f3adb0c47390349cc6f841143627d4de  new
Line 1191111: Feb 13 22:14:27.268  INFO main: AddToWallet b9a4cce1c1b1ae84261c0687ab152a83f3adb0c47390349cc6f841143627d4de  update
Line 1191196: Feb 13 22:18:11.096  INFO Init: main: AddToWallet b9a4cce1c1b1ae84261c0687ab152a83f3adb0c47390349cc6f841143627d4de  update
Line 1191313: Feb 13 22:23:15.348  INFO Init: main: AddToWallet b9a4cce1c1b1ae84261c0687ab152a83f3adb0c47390349cc6f841143627d4de  update
Line 1191377: Feb 13 22:27:06.165  INFO main: AddToWallet 43baa44e9f1335a15e5c5412584b2e001def74d94a76ddcc30b22fee15f79289  new
Line 1191380: Feb 13 22:27:06.264  INFO main: AddToWallet 43baa44e9f1335a15e5c5412584b2e001def74d94a76ddcc30b22fee15f79289  update
Line 1191386: Feb 13 22:28:01.247  INFO main: AddToWallet 43baa44e9f1335a15e5c5412584b2e001def74d94a76ddcc30b22fee15f79289  update
Line 1192733: Feb 14 16:20:01.246  INFO main: AddToWallet f6d3e4a05d2c5b43caf3cdc1a36ba93877f0d786b87a107ef348e16d2d6b908c  new
Line 1192776: Feb 14 16:24:25.565  INFO main: AddToWallet b48591f1cabd46509a66b937fe0b7905085da5a882cb343f863604d8464c28bf  new
Line 1192781: Feb 14 16:24:26.213  INFO main: AddToWallet b48591f1cabd46509a66b937fe0b7905085da5a882cb343f863604d8464c28bf
Line 1192789: Feb 14 16:26:15.212  INFO main: AddToWallet b48591f1cabd46509a66b937fe0b7905085da5a882cb343f863604d8464c28bf  update
Line 1193289: Feb 14 19:03:09.459  INFO main: AddToWallet f011ca4db4810b61c4e5beee53bf4d2938f486a7cc84639a94525f6c7edef107  new
Line 1193292: Feb 14 19:03:10.359  INFO main: AddToWallet f011ca4db4810b61c4e5beee53bf4d2938f486a7cc84639a94525f6c7edef107
Line 1193304: Feb 14 19:04:29.356  INFO main: AddToWallet f011ca4db4810b61c4e5beee53bf4d2938f486a7cc84639a94525f6c7edef107  update
Line 1193431: Feb 14 20:28:42.806  INFO Init: main: AddToWallet f011ca4db4810b61c4e5beee53bf4d2938f486a7cc84639a94525f6c7edef107
Line 1193552: Feb 14 20:31:56.267  INFO main: AddToWallet 441479f39c59ec4e171bd6f952d238fc60d341670a46ad607f3438d27400c4a7  new
Line 1193555: Feb 14 20:31:56.277  INFO main: AddToWallet 441479f39c59ec4e171bd6f952d238fc60d341670a46ad607f3438d27400c4a7
```

Figure 22. Z. Fullnode - Disk analysis. Incoming and outgoing transaction IDs in debug.log file.

One way to identify what are the outgoing transactions is with the keyword *z_sendmany*. In Figure 23, it is possible to observe four transaction IDs that belong to transactions 4, 5, 6 and 7, and each record has the keyword previously mentioned. However, from this file, it was not possible to differentiate what transaction was private or public, for that it would be necessary to use the transaction ID on the blockchain.

```
Line 1191379: Feb 13 22:27:06.198  INFO main: opid-3d594061-b2b3-4848-a480-dfbcfed70ad7: z_sendmany finished
    (status=success, txid=43baa44e9f1335a15e5c5412584b2e001def74d94a76ddcc30b22fee15f79289)
Line 1192617: Feb 14 16:06:09.352  INFO main: opid-3c90c382-5ef0-4f94-99ba-e3868d1521d1: z_sendmany finished
    (status=failed, error=Insufficient shielded funds, have 0.00007001, need 0.00008001)
Line 1192758: Feb 14 16:21:08.610  INFO main: opid-e27917d5-bab5-469f-b4b9-dd7a69844793: z_sendmany finished
    (status=failed, error=Insufficient shielded funds, have 0.00008001, need 0.00009001)
Line 1192765: Feb 14 16:21:47.996  INFO main: opid-ff5da6d4-dd4b-4636-bef1-6c8ea683a402: z_sendmany finished
    (status=failed, error=Insufficient shielded funds, have 0.00008001, need 0.00009001)
Line 1192778: Feb 14 16:24:25.596  INFO main: opid-4f9c9621-c82f-43f9-b10e-e4e7a6220e6a: z_sendmany finished
    (status=success, txid=b48591f1cabd46509a66b937fe0b7905085da5a882cb343f863604d8464c28bf)
Line 1193291: Feb 14 19:03:09.477  INFO main: opid-4de11b15-6f30-4d04-bd1d-e8b1842aca7f: z_sendmany finished
    (status=success, txid=f011ca4db4810b61c4e5beee53bf4d2938f486a7cc84639a94525f6c7edef107)
Line 1193539: Feb 14 20:30:09.972  INFO main: opid-76b43a97-5586-490f-9d0a-7e989d23dd3c: z_sendmany finished
    (status=failed, error=Insufficient transparent funds, have 0.00079, need 0.0008)
Line 1193554: Feb 14 20:31:56.277  INFO main: opid-7bb2a0d0-4eb1-4a03-ad47-6b121b9d690a: z_sendmany finished
    (status=success, txid=441479f39c59ec4e171bd6f952d238fc60d341670a46ad607f3438d27400c4a7)
```

Figure 23. Z. Fullnode - Disk analysis. Outgoing transaction IDs along with the z_sendmany keyword.

Another interesting piece of information that can be found in this file is the public IP address used while the wallet application was connected to the internet. The IP address 193.40.148.245 is depicted in Figure 24.

```
Line 1182510: Feb 08 08:02:43.236  INFO main: AdvertizeLocal: advertizing address 193.40.148.245:8233
Line 1182511: Feb 08 08:02:43.236  INFO main: AdvertizeLocal: advertizing address 193.40.148.245:8233
Line 1182512: Feb 08 08:02:43.236  INFO main: AdvertizeLocal: advertizing address 193.40.148.245:8233
Line 1182513: Feb 08 08:02:43.236  INFO main: AdvertizeLocal: advertizing address 193.40.148.245:8233
Line 1182514: Feb 08 08:02:43.236  INFO main: AdvertizeLocal: advertizing address 193.40.148.245:8233
Line 1182515: Feb 08 08:02:43.236  INFO main: AdvertizeLocal: advertizing address 193.40.148.245:8233
Line 1182516: Feb 08 08:02:43.236  INFO main: AdvertizeLocal: advertizing address 193.40.148.245:8233
```

Figure 24. Z. Fullnode - Disk analysis. External IP address used by the wallet application.

This finding is valuable for the investigator since it can be used as an input for further steps of the investigations, such as IP geolocation in coordination with the internet service provider. Moreover, taking into account that a PC can use dynamic IP addresses, this finding can provide the investigator with the exact IP address used by the PC while the wallet application was active.

The *AddressBook.json* file was created during case 4 after the user's action of adding contacts to the address book. The information contained in this file can reveal what other transparent and shielded addresses could the user have been sending or receiving money. Likewise, it is possible to see that each address has its corresponding label that helps the owner of the wallet to recognise easily and differentiate one from another. For instance, Figure 25 shows the content of the file and the addresses that belong to the iPhone device used to support the creation of cases, but it is outside the scope of the forensic analysis.


```
[{"label":"ZiPhone","address":
"zs13tem6fljqf5kskn0kvgeqcxrhat7tj37w9l5w5vt0mmnuamsxchqlqptqrvvhz97g5zxg6670mu"
},{"label":"TiPhone","address":"t1dv9Gzg8tWphFLuTdwBrSipkjbduVospqa"}]
```


Figure 25. Z. Fullnode - Disk analysis. Content of AddressBook.json file.

The *wallet.dat* file contains information only from the local wallet, such as transaction history, transparent and shielded addresses with their corresponding private keys. This file is one of the most important since it has all the information related to the wallet. However, when it was analysed without being restored in the wallet application, this one showed little information. For instance, no traces of shielded addresses were found, and only transparent addresses were present. Figure 26 illustrates the VMfullnode's transparent address *t1gxPPoGQuy6PT5QJFdC8wEjP7hUETG3Yrw* used during the creation of cases.


```
wallet.dat
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
000104C0  08 00 01 07 72 65 63 65 69 76 65 84 2C 00 01 07  ....receive„,...
000104D0  70 75 72 70 6F 73 65 23 74 31 67 78 50 50 6F 47  purpose#t1gxPPoG
000104E0  51 75 79 36 50 54 35 51 4A 46 64 43 38 77 45 6A  Quy6PT5QJFdC8wEj
000104F0  50 37 68 55 45 54 47 33 59 72 77 00 22 00 01 21  P7hUETG3Yrw."..!
00010500  02 AC 7D 99 05 50 10 87 34 B6 27 4D 63 29 47 59  .¬}™.P.‡4¶'Mc)GY
00010510  C0 B3 29 0F 89 3E 02 23 62 27 79 4F 56 36 A9 7B  À³).‰>.#b'yOV6©{
00010520  D2 3A D7 5B 0B 00 01 0A 64 65 66 61 75 6C 74 6B  Ò:×[....defaultk
00010530  65 79 04 20 2D 00 01 0A 00 00 00 F6 B1 1F 60 00  ey. -......ö±.`.
```


Figure 26. Z. Fullnode - Disk analysis. Transparent address of VMfullnode.

In like manner, the change transparent address *t1TtbEmyGdrWGkg6Cqpia4wjAz7uYDe5ouJ*, created automatically to receive the unspent amount of a transaction, was found in the file, and it is illustrated in Figure 27.

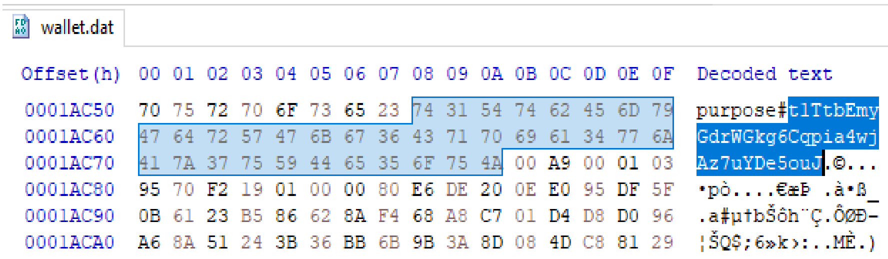


Figure 27. Z. Fullnode - Disk analysis. Transparent change address created automatically by the wallet application.

Moreover, this file needs to be backed up constantly and right after a new address is created. Otherwise, a previous backup will not contain the new address. Besides, if someone else has access to this file, they can gain access to the entire wallet and funds.

The last file named *zcash.conf* contains the configuration settings to interact with the Zcash. This file by default contains the following parameters that are documented in detail in the Zcash documentation [46]:

- *server=1*. Tells zcashd to accept JSON-RPC commands.
- *rpcuser=zecwallet*. Default user to interact with the zcashd.
- *rpcpassword=ydssk5vd6zs*. The default password for rpcuser.
- *ibdskiptxverification=1* Allows faster synchronisation during initial block sync [47].

Besides those four parameters above mentioned, and as it was explained in the memory analysis, case eight, the parameter *exportdir=C:\Users\juanm\Desktop\exportwallet_cmd* was added as part of the steps required to execute the *z_exportwallet* command. This parameter contains the path where the bundle file will be saved. For instance, Figure 28 depicts the file with the five parameters mentioned above:

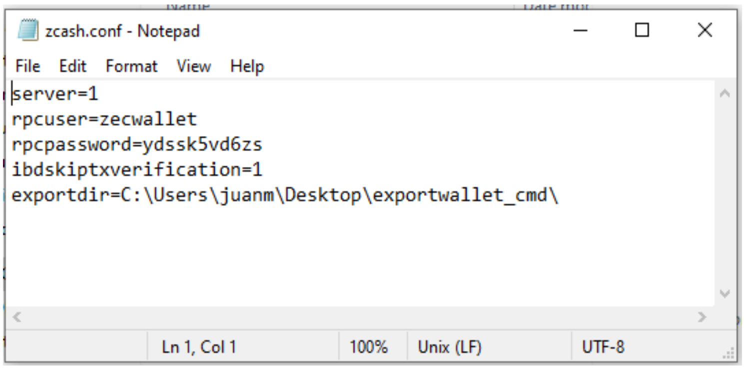


Figure 28. Z. Fullnode - Disk analysis. zcash.conf file obtained from the disk acquisition.

This last parameter could be an indicator for the investigator that the user has interacted with Zcash using the CLI command to make a backup of the HD Seed and private keys.

4.1.3 Network Traffic

These files do not contain much valuable information since all network traffic is encrypted by the wallet application. Only DNS traffic is observable, and this traffic goes to the Zcash DNS seeders. “DNS seeds are well-known stable domain names that, when resolved, return the addresses of peers that are currently participating in the network” [48].

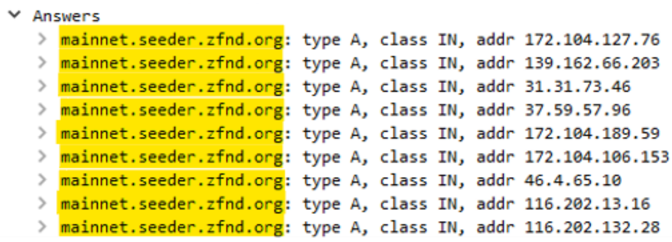


Figure 29. Z. Fullnode - Network analysis. DNS queries to Zcash seeders.

4.2 Zecwallet Lite

The following section will analyse the light version of the Zecwallet software to identify what forensic artefacts can be obtained. To have a better understanding of the direction of the transactions, the diagram depicted in Figure 30 shows who was the sender and recipient from cases 2 to 5.

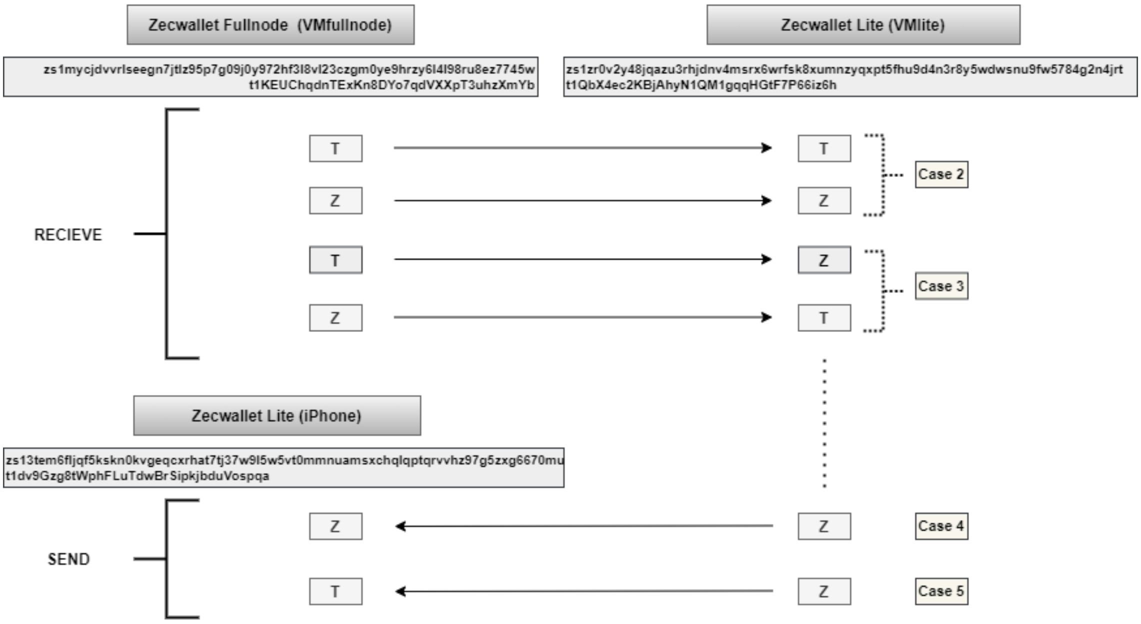


Figure 30. The direction of transactions between Zecwallet Fullnode, Zecwallet Lite and iPhone device.

Likewise, as was previously explained in the workflow section, case 1 does not encompass transactions; that is why it is not present in the diagram.

4.2.1 Memory Images

4.2.1.1 Case 1

Once the application is executed for the first time, this automatically generates the 24-word mnemonic phrase or seed phrase that is shown in Figure 31.

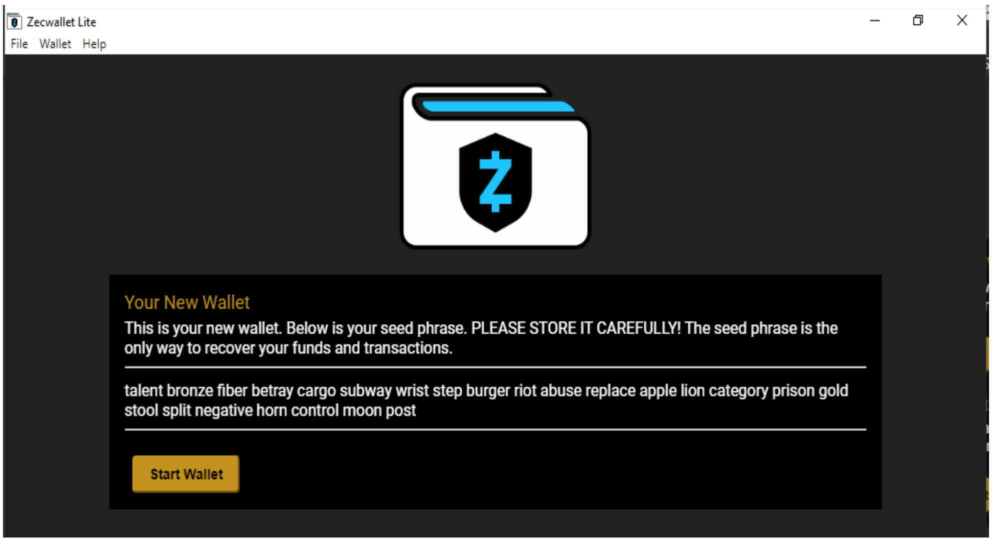


Figure 31. Z. Lite - Mem. analysis. Case 1. The mnemonic phrase generated automatically by the wallet application.

Transparent and shielded addresses were created by default once the application was started.

As part of the structured analysis, the 24-words mnemonic phrase was found in the dumped PID 6004. Figure 32 illustrates the finding. For an investigator, this would be difficult to find since there is not a keyword that makes it easy to locate.

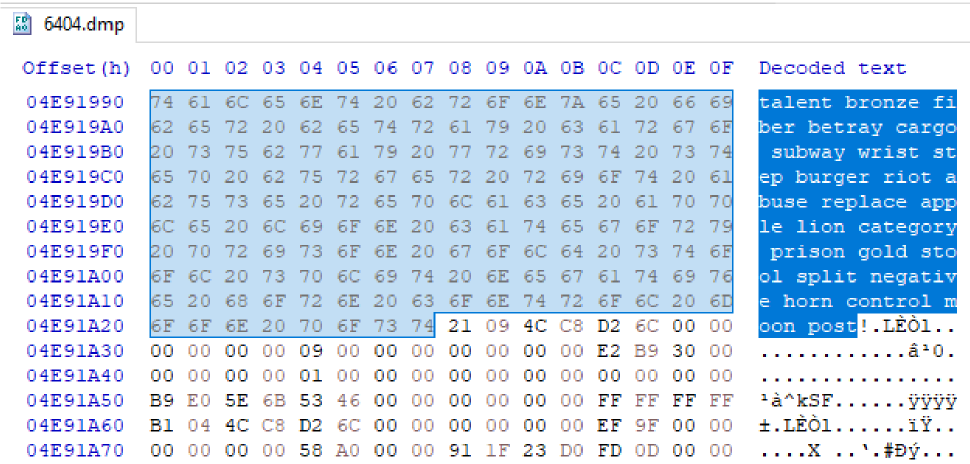

```
6404.dmp
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
04E91990  74 61 6C 65 6E 74 20 62 72 6F 6E 7A 65 20 66 69  talent bronze fi
04E919A0  62 65 72 20 62 65 74 72 61 79 20 63 61 72 67 6F  ber betray cargo
04E919B0  20 73 75 62 77 61 79 20 77 72 69 73 74 20 73 74   subway wrist st
04E919C0  65 70 20 62 75 72 67 65 72 20 72 69 6F 74 20 61  ep burger riot a
04E919D0  62 75 73 65 20 72 65 70 6C 61 63 65 20 61 70 70  buse replace app
04E919E0  6C 65 20 6C 69 6F 6E 20 63 61 74 65 67 6F 72 79  le lion category
04E919F0  20 70 72 69 73 6F 6E 20 67 6F 6C 64 20 73 74 6F   prison gold sto
04E91A00  6F 6C 20 73 70 6C 69 74 20 6E 65 67 61 74 69 76  ol split negativ
04E91A10  65 20 68 6F 72 6E 20 63 6F 6E 74 72 6F 6C 20 6D  e horn control m
04E91A20  6F 6F 6E 20 70 6F 73 74 21 09 4C C8 D2 6C 00 00  oon post!.LÈÒl..
04E91A30  00 00 00 00 09 00 00 00 00 00 00 00 E2 B9 30 00  ............â¹0.
04E91A40  00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00  ................
04E91A50  B9 E0 5E 6B 53 46 00 00 00 00 00 00 FF FF FF FF  ¹à^kSF......ÿÿÿÿ
04E91A60  B1 04 4C C8 D2 6C 00 00 00 00 00 00 EF 9F 00 00  ±.LÈÒl......ïŸ..
04E91A70  00 00 00 00 58 A0 00 00 91 1F 23 D0 FD 0D 00 00  ....X ..'.#Ðý...
```

Figure 32. Z. Lite – Mem. analysis. Case 1. Mnemonic phrase present in dumped PID 6004.

The unstructured analysis shows information about the transparent address *t1QbX4ec2KBjAhyN1QM1gqqHGtF7P66iz6h* and shielded addresses *zs1zr0v2y48jqazu3rhjdnv4msrx6wrfsk8xumnzyqxpt5fhu9d4n3r8y5wdwsnu9fw5784g2n4jrt*, created automatically by the wallet application.

Figure 33. Z. Lite – Mem. analysis. Case 1. Transparent and shielded address created by the wallet application.

No private keys related to the transparent and shielded address were found.

4.2.1.2 Case 2

The wallet was encrypted with the password "*arribaperu*" as part of the test. Figure 34 illustrates the pop-up message confirming the encryption.

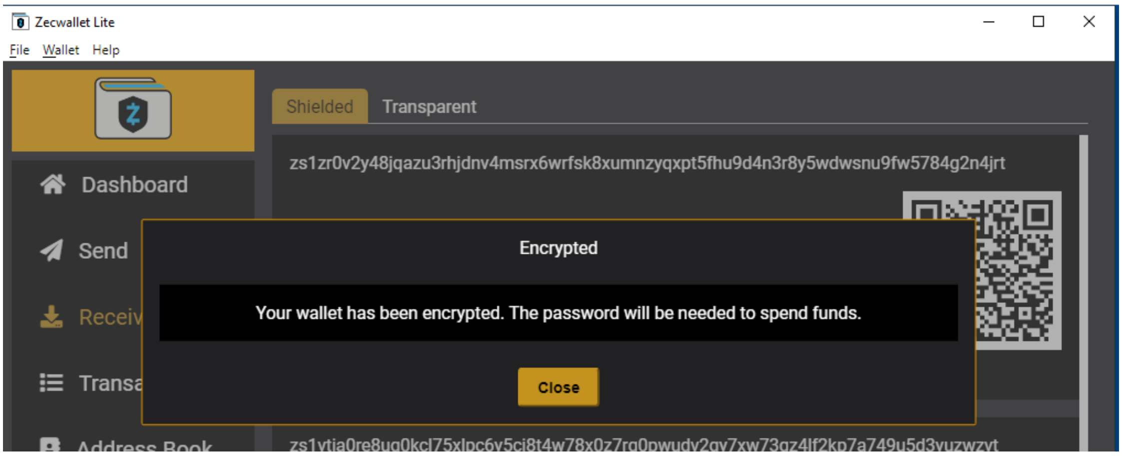


Figure 34. Z. Lite. Case 2. A message confirming that the wallet was encrypted.

The structured analysis did not show any process related to the wallet application as the previous case did. This behaviour may be because the wallet was first encrypted, and then the memory acquisition was performed. However, it is still possible to see information about what files and directories were accessed or modified by the wallet application on the MFT records.

As for the unstructured analysis, Figure 35 shows the transaction ID of the fist transaction *2850f2152523bdff6f48d7ab475718785e56c947cd2967b1b5f8d3cb7ec072aa*.

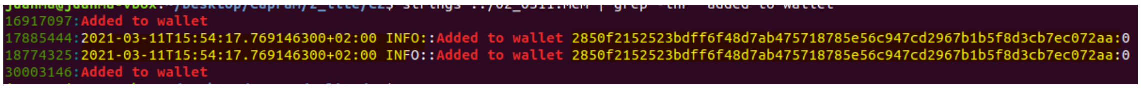


Figure 35. Z. Lite – Mem. analysis. Case 2. First transaction ID.

Regarding the second transaction, Figure 36 also reveals its corresponding transaction ID *066e1bd24b796e76be202ab99ffa87688bdf032e281c97d58a2fa2a1fb71e584*.

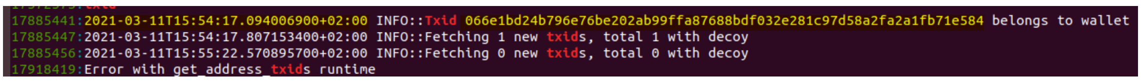


Figure 36. Z. Lite – Mem. analysis. Case 2. Second transaction ID.

Also, the memory file shows evidence that an incoming transaction was done since the phrase *Receiving sapling output to* is followed by the receiving shielded local address *zs1zr0v2y48jqazu3rhjdnv4msrx6wrfsk8xumnzyqxpt5fhu9d4n3r8y5wdwsnu9fw5784g2n4jrt*. Figure 37 depicts the finding.

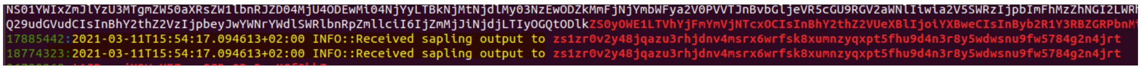


Figure 37. Z. Lite – Mem. analysis. Case 2. Evidence to an incoming transaction to a shielded address

Finally, the password used to encrypt the wallet was not found; neither were the private keys from transparent nor the shielded addresses.

4.2.1.3 Case 3

The structured analysis shows in Figure 38 the message sent on the memo field from the third transaction that was found in the dumped PID 5860.

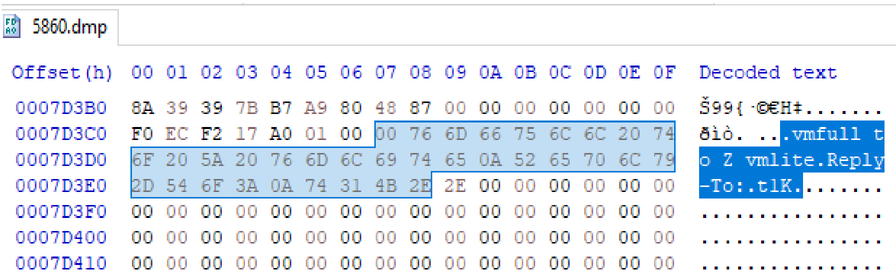


Figure 38. Z. Lite – Mem. analysis. Case 3. Content of the memo field sent on the first incoming transaction.

Information about the creation of the file AddressBook.json was found in the MFT record of the memory file and is illustrated in Figure 39.

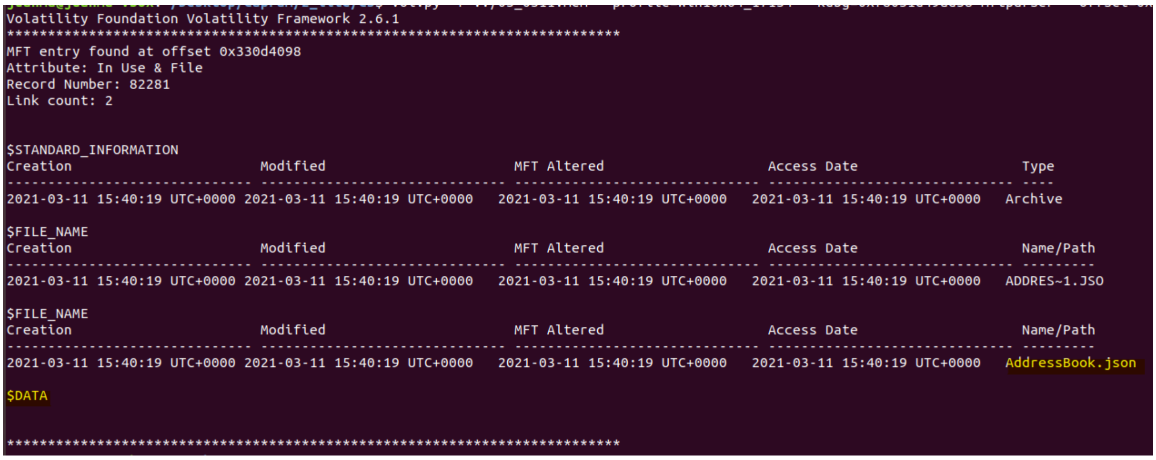


Figure 39. Z. Lite – Mem. analysis. Case 3. MFT record of AddressBook.json file.

The unstructured analysis in Figure 40 shows the transaction IDs of the four transactions completed up to that point.

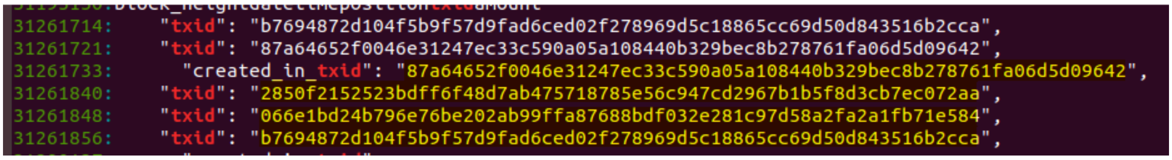


Figure 40. Z. Lite – Mem. analysis. Case 3. Transaction IDs of the fourth first transactions.

Since the wallet only received money from the third and fourth transaction, the password to decrypt it was not utilized, that is why there is no trace of the password used in case 2, nor any private keys were found.

4.2.1.4 Case 4

In the structured analysis, Figure 41 shows information of the MFT related to the AddressBook.json file that was created in the previous case. In the figure is illustrated the content of the file and metadata about the access date.

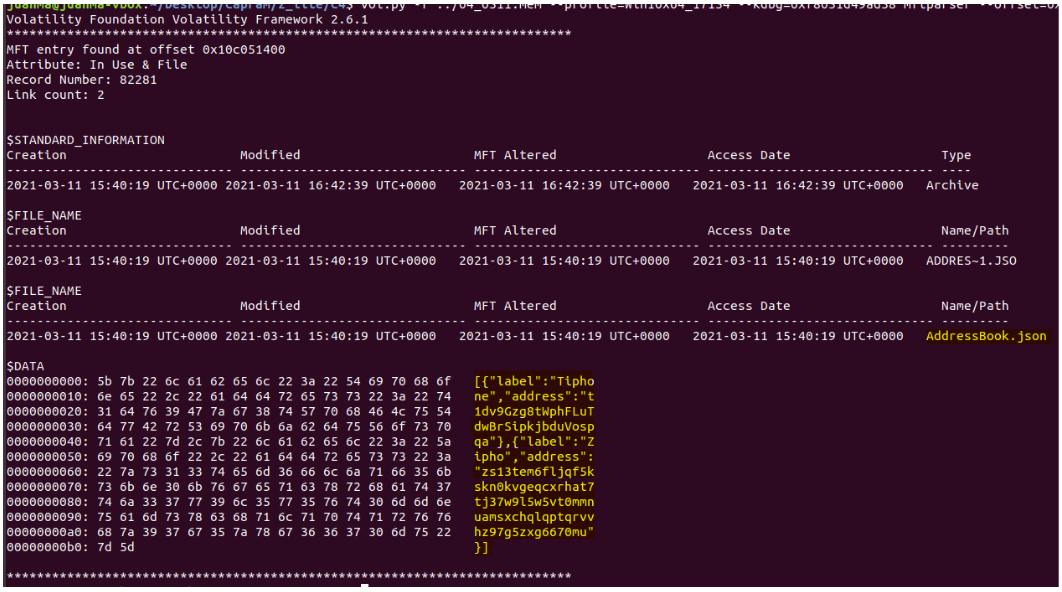


Figure 41. Z. Lite – Mem. analysis. Case 4. MFT record showing information about the AddressBook.json.

Also, it can be seen metadata information about the *zecwallet_transactions.csv* file on the MFT records as illustrated in Figure 42.

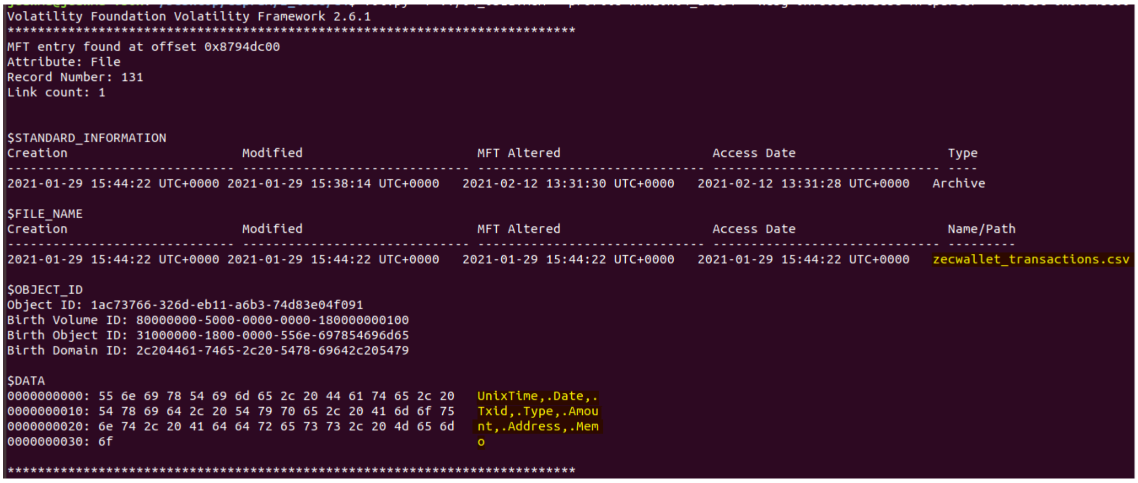


Figure 42. Z. Lite – Mem. analysis. Case 4. MFT record showing metadata of the file zecwallet_transactioins.csv.

Another interesting finding was the password used to encrypt the wallet. This one was found on the dumped PID 6852, and it is illustrated in Figure 43. However, since there was not a keyword to make it easy to identify for the investigator, it would be challenging to locate it.

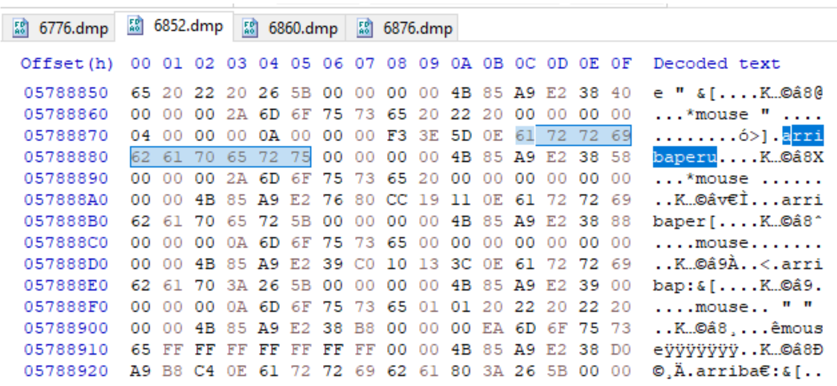


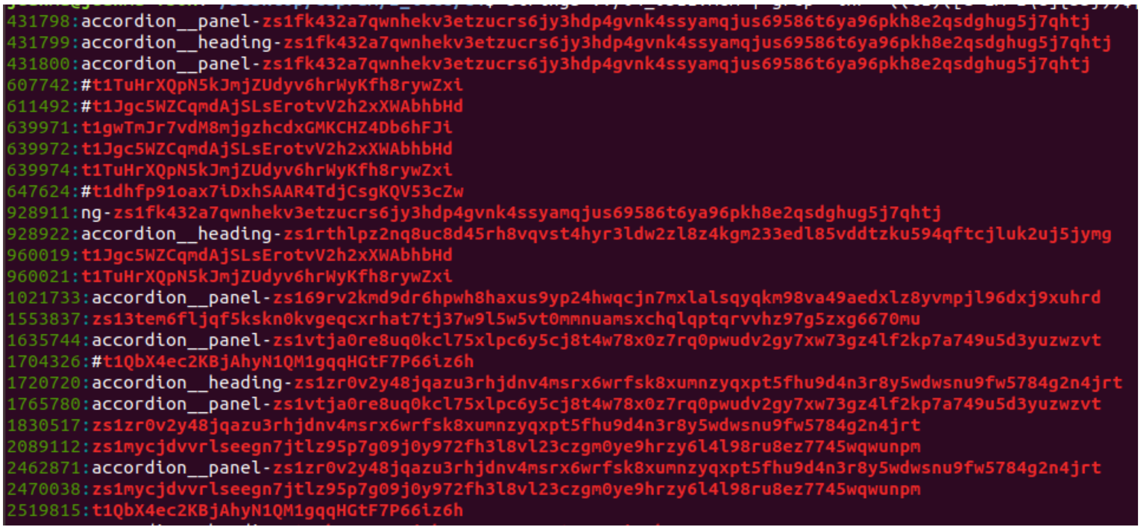
Figure 43. Z. Lite – Mem. analysis. Case 4. Password used to encrypt/decrypt wallet in plaintext.

The results of the unstructured analysis show information about the transaction IDs of the transactions made until now. Figure 44 illustrates the results.

```
18538228:     "txid": "2850f2152523bdff6f48d7ab475718785e56c947cd2967b1b5f8d3cb7ec072aa",
18538236:     "txid": "066e1bd24b796e76be202ab99ffa87688bdf032e281c97d58a2fa2a1fb71e584",
18538244:     "txid": "b7694872d104f5b9f57d9fad6ced02f278969d5c18865cc69d50d843516b2cca",
18538251:     "txid": "87a64652f0046e31247ec33c590a05a108440b329bec8b278761fa06d5d09642",
18538259:     "txid": "fb9d975ad5a2a09ebff448a47318c4b8a04e59be761a551a1d8ac904a27232aa",
18641753:     "txid": "2850f2152523bdff6f48d7ab475718785e56c947cd2967b1b5f8d3cb7ec072aa",
18641761:     "txid": "066e1bd24b796e76be202ab99ffa87688bdf032e281c97d58a2fa2a1fb71e584",
18641769:     "txid": "b7694872d104f5b9f57d9fad6ced02f278969d5c18865cc69d50d843516b2cca",
18641776:     "txid": "87a64652f0046e31247ec33c590a05a108440b329bec8b278761fa06d5d09642",
18641784:     "txid": "fb9d975ad5a2a09ebff448a47318c4b8a04e59be761a551a1d8ac904a27232aa",
```

Figure 44. Z. Lite – Mem. analysis. Case 4. Transaction ID of the first five transactions.

Also, it is possible to see depicted in Figure 45 the change addresses created automatically by the wallet application to receive the remainder of ZEC when the amount of the transaction is not exact.



```
431798:accordion__panel-zs1fk432a7qwnhekv3etzucrs6jy3hdp4gvnk4ssyamqjus69586t6ya96pkh8e2qsdghug5j7qhtj
431799:accordion__heading-zs1fk432a7qwnhekv3etzucrs6jy3hdp4gvnk4ssyamqjus69586t6ya96pkh8e2qsdghug5j7qhtj
431800:accordion__panel-zs1fk432a7qwnhekv3etzucrs6jy3hdp4gvnk4ssyamqjus69586t6ya96pkh8e2qsdghug5j7qhtj
607742:#t1TuHrXQpN5kJmjZUdyv6hrWyKfh8rywZxi
611492:#t1Jgc5WZCqmdAjSLsErotvV2h2xXWAbhbHd
639971:t1gwTmJr7vdM8mjgzhcdxGMKCHZ4Db6hFJi
639972:t1Jgc5WZCqmdAjSLsErotvV2h2xXWAbhbHd
639974:t1TuHrXQpN5kJmjZUdyv6hrWyKfh8rywZxi
647624:#t1dhfp91oax7iDxhSAAR4TdjCsgKQV53cZw
928911:ng-zs1fk432a7qwnhekv3etzucrs6jy3hdp4gvnk4ssyamqjus69586t6ya96pkh8e2qsdghug5j7qhtj
928922:accordion__heading-zs1rthlpz2nq8uc8d45rh8vqvst4hyr3ldw2zl8z4kgm233edl85vddtzku594qftcjluk2uj5jymg
960019:t1Jgc5WZCqmdAjSLsErotvV2h2xXWAbhbHd
960021:t1TuHrXQpN5kJmjZUdyv6hrWyKfh8rywZxi
1021733:accordion__panel-zs169rv2kmd9dr6hpwh8haxus9yp24hwqcjn7mxlalsqyqkm98va49aedxlz8yvmpjl96dxj9xuhrd
1553837:zs13tem6fljqf5kskn0kvgeqcxrhat7tj37w9l5w5vt0mmnuamsxchqlqptqrvvhz97g5zxg6670mu
1635744:accordion__panel-zs1vtja0re8uq0kcl75xlpc6y5cj8t4w78x0z7rq0pwudv2gy7xw73gz4lf2kp7a749u5d3yuzwzvt
1704326:#t1QbX4ec2KBjAhyN1QM1gqqHGtF7P66iz6h
1720720:accordion__heading-zs1zr0v2y48jqazu3rhjdnv4msrx6wrfsk8xumnzyqxpt5fhu9d4n3r8y5wdwsnu9fw5784g2n4jrt
1765780:accordion__panel-zs1vtja0re8uq0kcl75xlpc6y5cj8t4w78x0z7rq0pwudv2gy7xw73gz4lf2kp7a749u5d3yuzwzvt
1830517:zs1zr0v2y48jqazu3rhjdnv4msrx6wrfsk8xumnzyqxpt5fhu9d4n3r8y5wdwsnu9fw5784g2n4jrt
2089112:zs1mycjdvvrlseegn7jtlz95p7g09j0y972fh3l8vl23czgm0ye9hrzy6l4l98ru8ez7745wqwunpm
2462871:accordion__panel-zs1zr0v2y48jqazu3rhjdnv4msrx6wrfsk8xumnzyqxpt5fhu9d4n3r8y5wdwsnu9fw5784g2n4jrt
2470038:zs1mycjdvvrlseegn7jtlz95p7g09j0y972fh3l8vl23czgm0ye9hrzy6l4l98ru8ez7745wqwunpm
2519815:t1QbX4ec2KBjAhyN1QM1gqqHGtF7P66iz6h
```

Figure 45. Z. Lite – Mem. analysis. Case 4. Transparent and shielded change addresses.

No private keys related to the addresses used during the transactions were found.

4.2.1.5 Case 5

The option “Export All Privates Keys” was executed, showing the private keys in a pop-up window, but these were not saved on disk. Figure 46 illustrates the private keys.

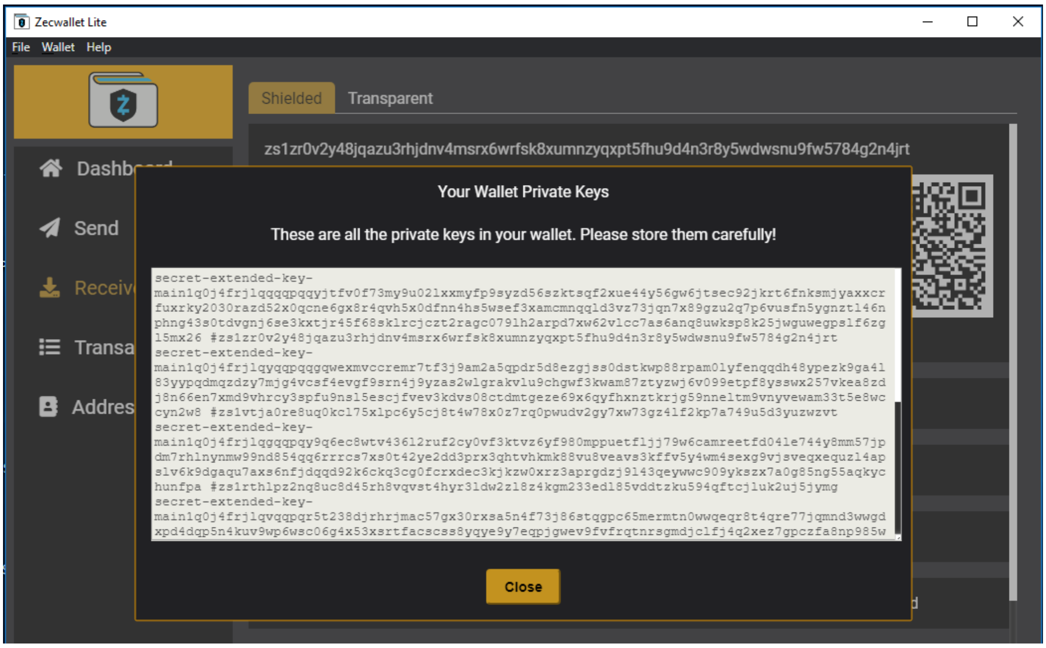


Figure 46. Z. Lite. Case 5. Private keys after the execution of export all private keys.

The structured analysis shows information about the password used to encrypt the wallet that was found in the dumped PID 4568. As was mentioned before, it would be challenging to find for the investigator. Figure 47 illustrates the finding.

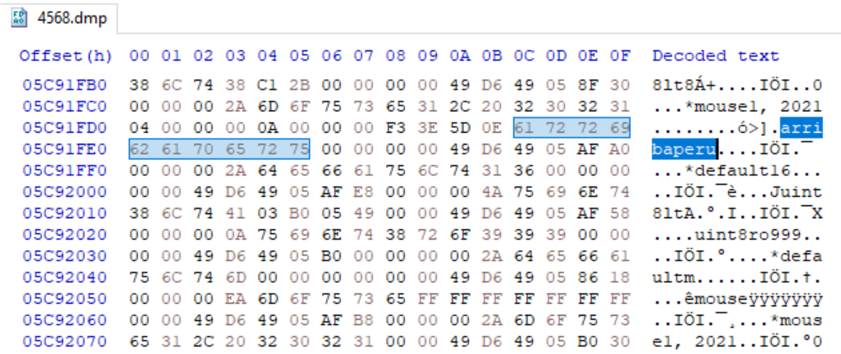


Figure 47. Z. Lite – Mem. analysis. Case 5. Password to decrypt the wallet found in plaintext.

Unstructured analysis shows in Figure 48 all transparent and shielded addresses with their corresponding private keys.

Figure 48. Z. Lite – Mem. analysis. Case 5. Transparent and shielded address with their corresponding private keys on memory.

This is interesting because the result obtained after the execution of the option *Export all private keys* was only shown on the screen and not saved on disk, which means that all information is kept in memory.

In like manner, as it was shown in previous cases, Figure 49 shows the transaction IDs from all transactions made up to this point.

Figure 49. Z. Lite – Mem. analysis. Case 5. Transaction IDs of case 1, 2, 3, 4 and 5.

4.2.2 Disk Files

Relevant information was found in the following files:

- The *zecwallet-light-wallet.dat* stores the local transactions made by the user of the wallet application.

- The *zecwallet-light-wallet.debug.log* contains general debugging information about the application and the transaction ID.

- The *AddressBook.json* file contains information about the contacts added from the wallet application by the user.

The *zecwallet-light-wallet.dat* file shows valuable information such as the transparent addresses owned by the local wallet application, the content of the memo field of the incoming and outgoing transactions, and the external transparent address involved in a transaction. Figure 50 depicts the transparent addresses of the local wallet application, but no private keys were found:

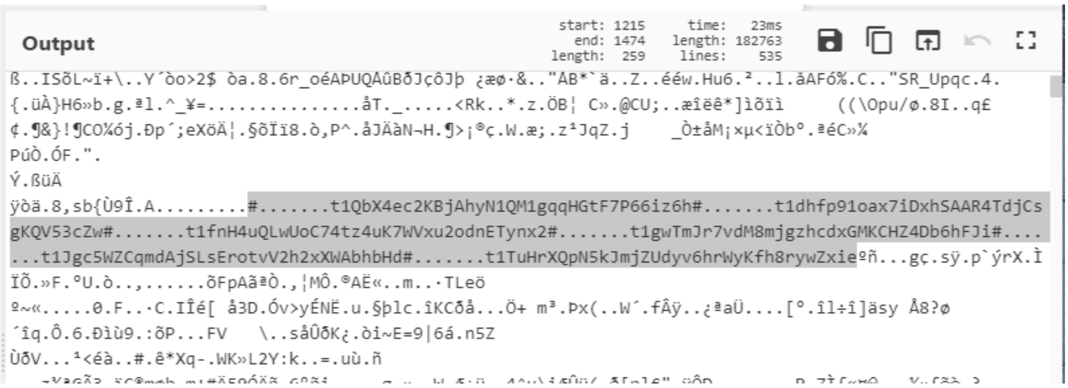


Figure 50. Z. Lite – Disk analysis. Local transparent addresses in the zecwallet-light-wallet.dat file.

Figure 51 shows the transparent iPhone address *t1dv9Gzg8tWphFLuTdwBrSipkjbduVospqa* used to create the study cases, which means that not only local addresses are displayed.

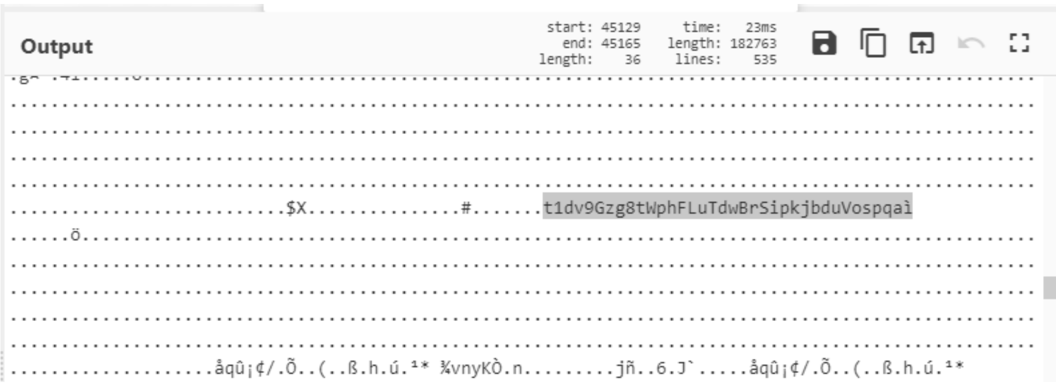


Figure 51. Z. Lite – Disk analysis. iPhone's transparent address in the zecwallet-light-wallet.dat file.

Likewise, for those transactions that allow one to include a memo field, it is possible to see the message. For instance, Figure 52 illustrates the message that was used in the first incoming transaction between the VMFullnode and the VMlite.

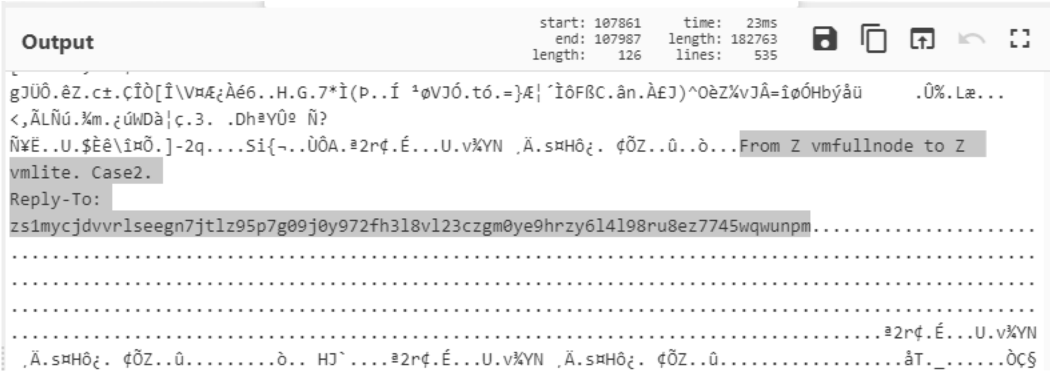


Figure 52. Z. Lite – Disk analysis. Content of memo field of first incoming transaction in the zecwallet-light-wallet.dat file.

The following message was used in case 3 when the third incoming transaction was done between the VMfullnode and the VMlite. The finding is depicted in Figure 53.

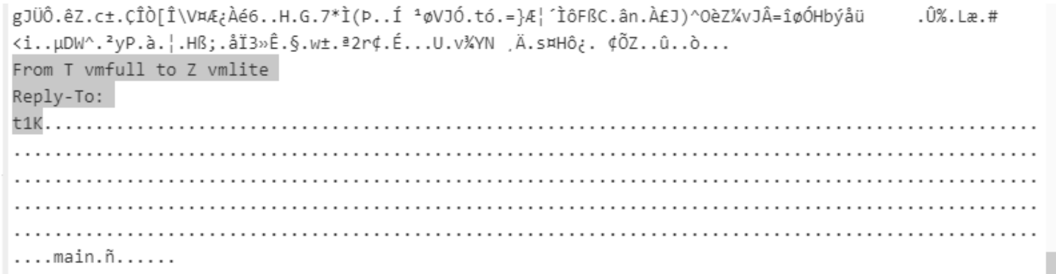


Figure 53. Z. Lite – Disk analysis. Content of memo field used in case 3 located in the zecwallet-light-wallet.dat file.

Finally, Figure 54 shows the message used on the fifth outgoing transaction of case 4. The most important finding here is the destination shielded address *zs13tem…6670mui*.

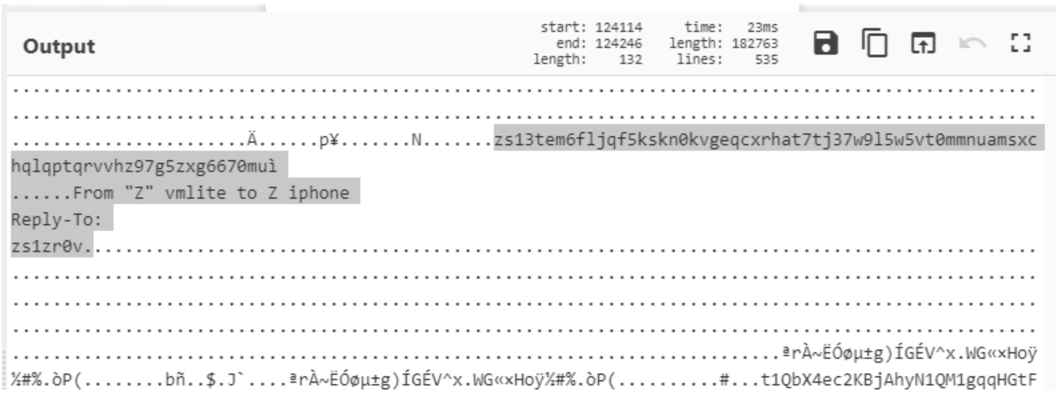


Figure 54. Z. Lite – Disk analysis. Content of memo field used in case 4 located in the zecwallet-light-wallet.dat file.

The content of these memo fields is valuable for the investigator, but this increases when the user of the local wallet application selects the *Include reply-To address* option before confirming the transaction. Because this option works like the email *reply to*, meaning that automatically the sending address is added to the message, at the same time revealing the source of the transaction. For instance, Figures 52, 53 and 54 have the *Reply-To:* text followed by the sending address, but only Figure 52 has the complete address; the rest was purposely removed as part of the tests.

For instance, Figure 55 illustrates on the left the screen of the VMfullnode wallet application, and on the right the screen of the VMlite application. As it can be seen, the VMfullnode has the option *Include Reply-To address* marked; this automatically adds the sending address *zs1my…wunpm* to the memo field. On the other hand, when the user of the VMlite receives the transaction, one can see the sending address *zs1my…wunpm* as part of the message on the memo field.

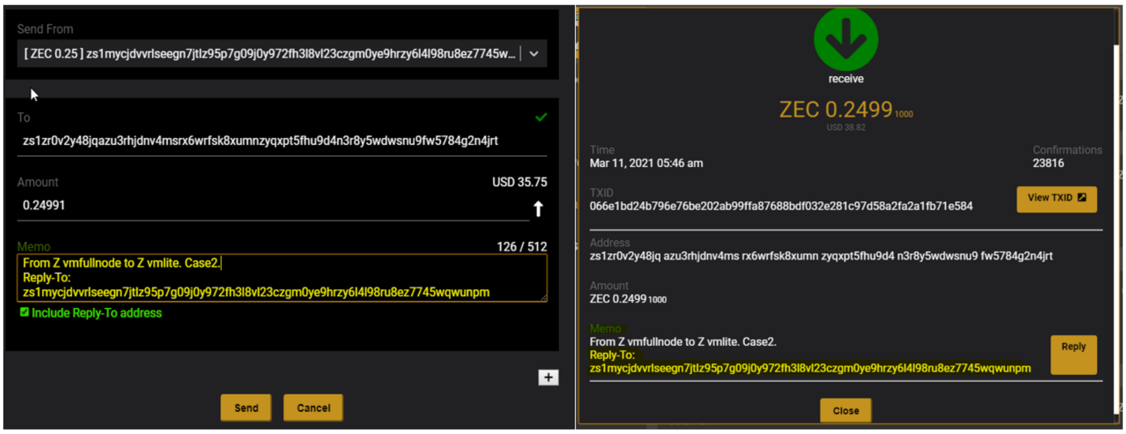


Figure 55. Z. Lite – Disk analysis. On the left is the sender’s screen, including the reply-to option. On the right the recipient’s screen with the sending address on the memo field.

The *zecwallet-light-wallet.debug.log* file contains information about the incoming and outgoing transaction IDs. This information can be input for the investigator to start looking for clues on the blockchain.

The *AddressBook.json* file has the addresses and corresponding labels that were added by the user from the wallet application.

No private keys nor the password used to encrypt the wallet were found.

4.2.3 Network Files

The network traffic was encrypted, and only the DNS queries were identified. The DNS queries were against the domain *lightwalletd.zecwallet.co*, which is the node or zcash network it can be observed in Figure 56. Additionally, the IP address answering the DNS queries was always 52.52.174.26 and belonged to Amazon.

```
0000  08 00 27 e6 e5 59 e8 1c  ba f0 11 27 08 00 45 00   ··'··Y·· ···'··E·
0010  00 57 90 2e 00 00 79 11  9d 2d 08 08 04 04 0a 1e   ·W·.··y· ·-······
0020  fe 10 00 35 d1 21 00 43  f5 86 05 76 81 80 00 01   ···5·!·C ···v····
0030  00 01 00 00 00 00 0c 6c  69 67 68 74 77 61 6c 6c   ·······l ightwall
0040  65 74 64 09 7a 65 63 77  61 6c 6c 65 74 02 63 6f   etd·zecw allet·co
0050  00 00 01 00 01 c0 0c 00  01 00 01 00 00 01 2b 00   ········ ······+·
0060  04 34 34 ae 1a                                     ·44··
```

Figure 56. Z. Lite – Network analysis. DNS queries to zcash network.

4.3 Dash Core

The present section will analyse the full node version of the software wallet for Dash cryptocurrency with the aim to identify the forensic artefacts.

The following diagram depicted in Figure 57 shows the direction of the transactions between the iPhone and the VMfullnode. Also, it is shown the addresses involved in each case and the type of transaction that was used. Moreover, Dash Core utilises only one address during the three transactions.

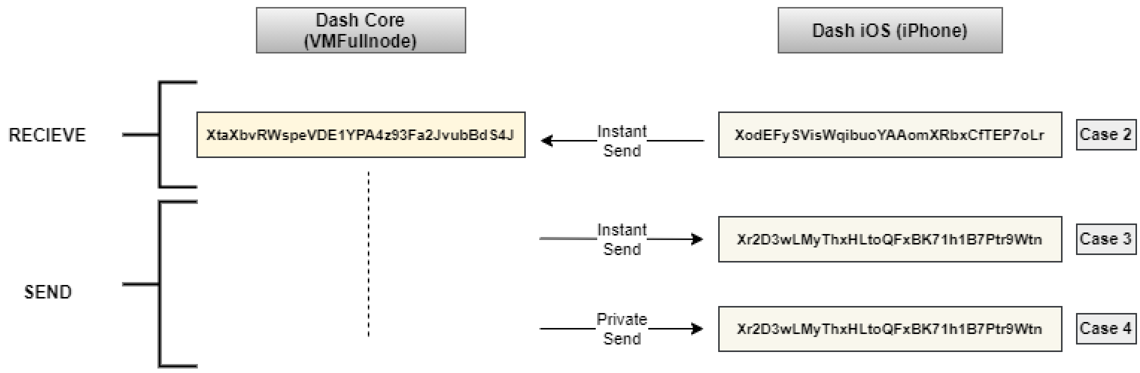


Figure 57. The direction of transactions between Dash Core and iPhone device.

Also, it can be noticed that case 1 and case 5 are not present in the diagram. This is due to those cases not encompassing a transaction, and their aim is another, as was explained in the case studies section.

4.3.1 Memory Images

4.3.1.1 Case 1

The structured and unstructured analysis shows information related to the installation of the wallet application and the download of the blockchain files. However, not relevant information for forensic investigation was found. Likewise, no private keys or any other interesting information was found.

4.3.1.2 Case 2

The structured analysis does not show interesting information that can be used during a forensic investigation. On the other hand, the unstructured analysis shows the date and transaction ID *d1b97eff84da15e1b10d95f2bdbf23feffb0e2af18e2465959c1b90dc58b25d7* illustrated in Figure 58.

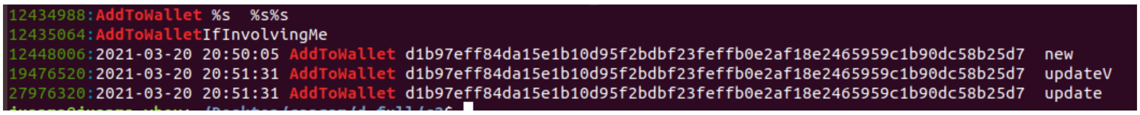


Figure 58. D. Core – Mem. analysis. Case 2. Date and ID from the first transaction.

And Figure 59 shows the recipient local address *XtaXbvRWspeVDE1YPA4z93Fa2JvubBdS4J* of the VMfullnode.

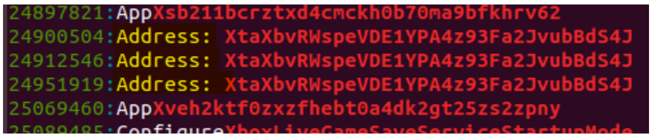


Figure 59. D. Core – Mem. analysis. Case 2. Sending address of the first transaction.

No trace of the sending address or private keys were found.

4.3.1.3 Case 3

The structured analysis shows no processes related to the wallet application. This behaviour might be due to the additional step of encrypting the wallet. However, the MFT records show the files that are part of the installation of the wallet application.

The unstructured analysis shows the path *C:\Users\juanm\Desktop\BKwallet.dat* where the backup wallet was saved. This finding is illustrated in Figure 60.

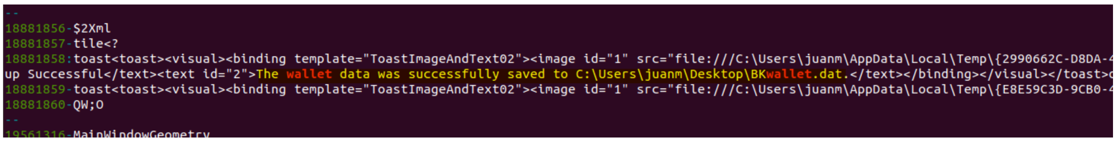


Figure 60. D. Core – Mem. analysis. Case 3. The path where the backup wallet was saved.

Also, Figure 61 shows information of the transaction such as the date *3/21/2021*, the amount *0.32 DASH*, the type *sent to the*, and the label *iPhone addr* that represents the recipient address. Nevertheless, the sending address is not present during the analysis.

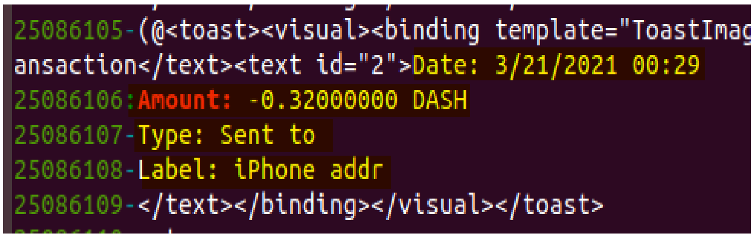


Figure 61. D. Core – Mem. analysis. Case 3. Information of second IS transaction.

To have a better understanding of how the real transaction was done, Figure 62 depicts the screen of the sending wallet application with the same information shown in Figure 61. In this picture, it is possible to see the recipient address *Xr2D3wLMyThxHLtoQFxBK71h1B7Ptr9Wtn* followed by the label field *iPhone addr*.

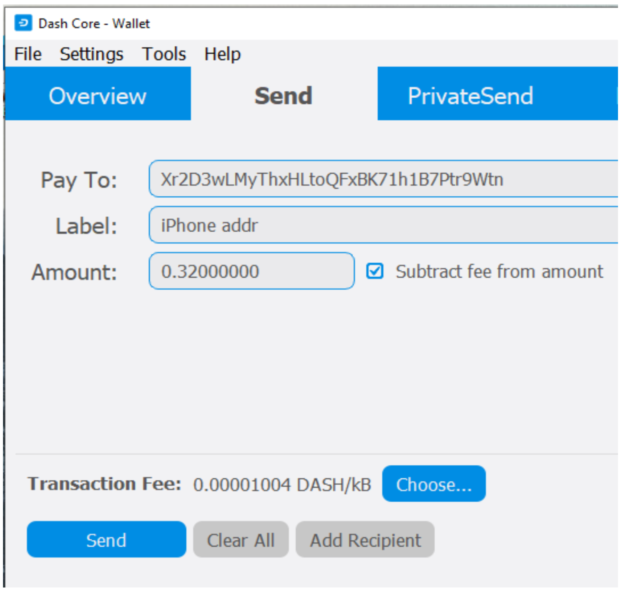
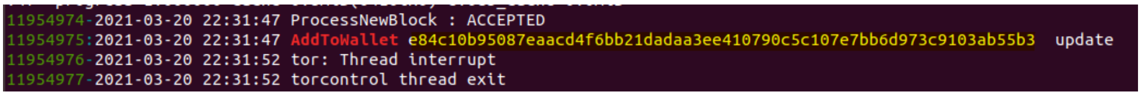


Figure 62. D. Core. Case 3. The screen of sending wallet showing the details of the transaction.

The transaction ID *e84c10b95087eaacd4f6bb21dadaa3ee410790c5c107e7bb6d973c9103ab55b3* was also found and illustrated in Figure 63.



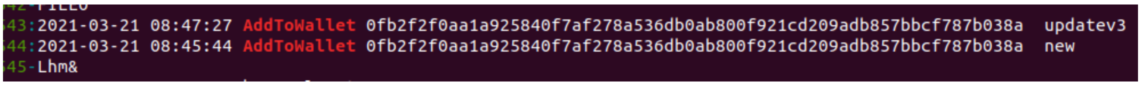
```
11954974-2021-03-20 22:31:47 ProcessNewBlock : ACCEPTED
11954975:2021-03-20 22:31:47 AddToWallet e84c10b95087eaacd4f6bb21dadaa3ee410790c5c107e7bb6d973c9103ab55b3  update
11954976-2021-03-20 22:31:52 tor: Thread interrupt
11954977-2021-03-20 22:31:52 torcontrol thread exit
```

Figure 63. D. Core – Mem. analysis. Case 3. The ID of the second transaction.

No private keys were found nor the password to encrypt the wallet.

4.3.1.4 Case 4

The structured analysis did not show interesting information to be used in a forensics investigation, contrasting with the unstructured analysis that shows the date and transaction ID *0fb2f2f0aa1a925840f7af278a536db0ab800f921cd209adb857bbcf787b038a* depicted in Figure 64.



```
43:2021-03-21 08:47:27 AddToWallet 0fb2f2f0aa1a925840f7af278a536db0ab800f921cd209adb857bbcf787b038a  updatev3
44:2021-03-21 08:45:44 AddToWallet 0fb2f2f0aa1a925840f7af278a536db0ab800f921cd209adb857bbcf787b038a  new
45-Lhm&
```

Figure 64. D. Core – Mem. analysis. Case 4. Date and ID from the second transaction.

No private keys were found nor the password to encrypt the wallet.

4.3.1.5 Case 5

The results obtained from the executed command *dumphdinfo* are depicted in Figure 65. Highlighted in green is the HD seed, while in yellow, the mnemonic phrase composed of 24-words.

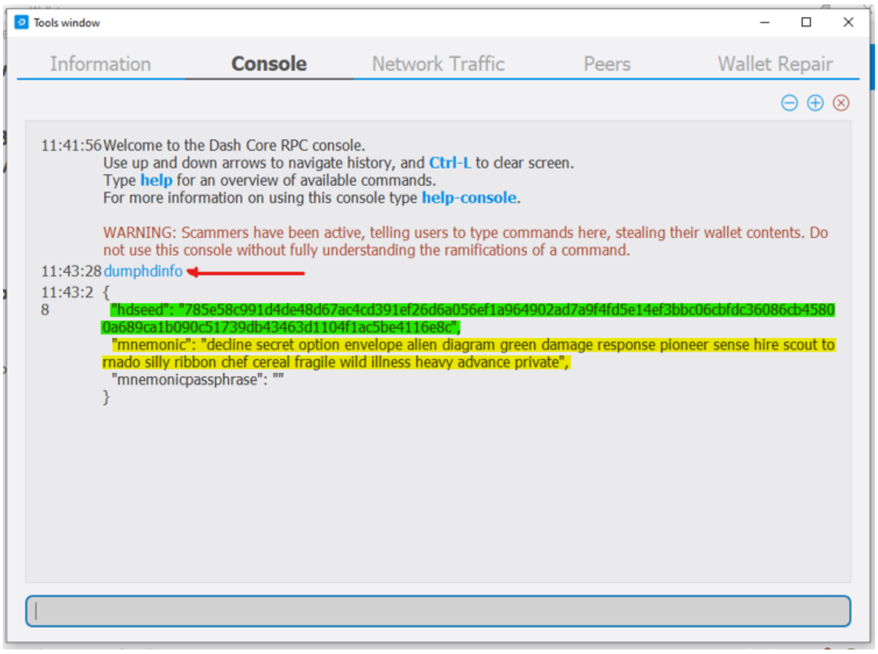


Figure 65. D. Core. Case 5. RPC console shows the executed dumphdinfo and the results.

The structured analysis shows in Figure 66 the presence of file *dashd.exe* in the MFT records. This file was not present in the previously analysed cases. Moreover, this file needs to be executed to enable the HD wallet, as explained in case 5 of Table 5.

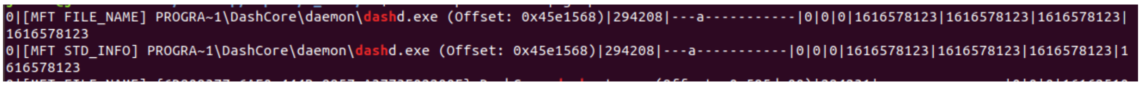


Figure 66. D. Core - Mem. analysis. Case 5. MFT is showing the file dashd.exe.

This is an interesting finding due to the fact that it gives the investigator clues about some additional features that have been executed on the wallet, such as the *dumphdinfo* command.

Likewise, the mnemonic phrase shown in Figure 65 appears in the dumped PID 2244, and it was present three times. In two out of three, the phrase has the *hdchain* word that can be used as a keyword to find it during analysis. This is illustrated in Figure 67.

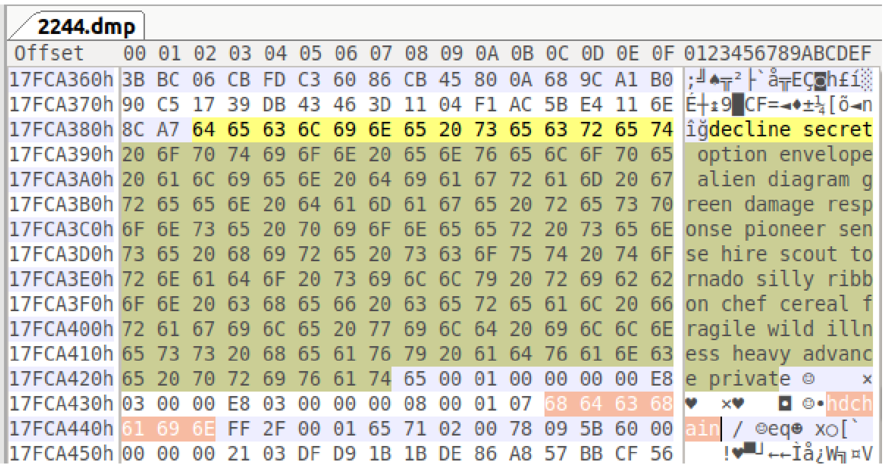


Figure 67. D. Core - Mem. analysis. Case 5. The mnemonic phrase found in memory.

4.3.1.6 Case 6

The analysis of the file does not show the password used for the encryption. Neither the passphrase was found.

4.3.2 Disk Files

Relevant information was found in the following files:

- The *wallet.dat* file contains information about the transactions, addresses and mnemonic phrase from the local wallet.
- The *debug.log* contains the transaction IDs and general information about the synchronization of the local wallet.

In the first file, it is possible to see traces of the transactions. Figure 68 depicts the same information used and shown in case 3 of memory analysis. The sending address *XtaXbvRWspeVDE1YPA4z93Fa2JvubBdS4J* and the recipient address *Xr2D3wLMyThxHLtoQFxBK71h1B7Ptr9Wtn* with the label *iPhone addr*.

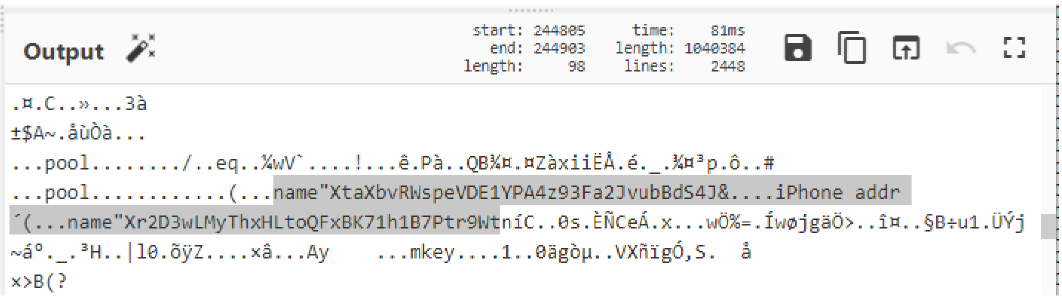


Figure 68. D. Core – Disk analysis. Information used in the transaction of case 3.

One interesting finding in this file is the correlation that can be done of who the sending and receiving address is. For instance, Figure 69 illustrates one *receive* highlighted in yellow, and two *send* highlighted in green. The yellow part represents the first incoming transaction, while the green part represents the second and third outgoing transactions. With this information, it is relatively easy to determine that *XtaXbvRWspeVDE1YPA4z93Fa2JvubBdS4J* is the local address of the wallet while *Xr2D3wLMyThxHLtoQFxBK71h1B7Ptr9Wtn* and *XogciEjYTBsczMER4dub1wqVf745GhZp25* are the external addresses that have received the money.

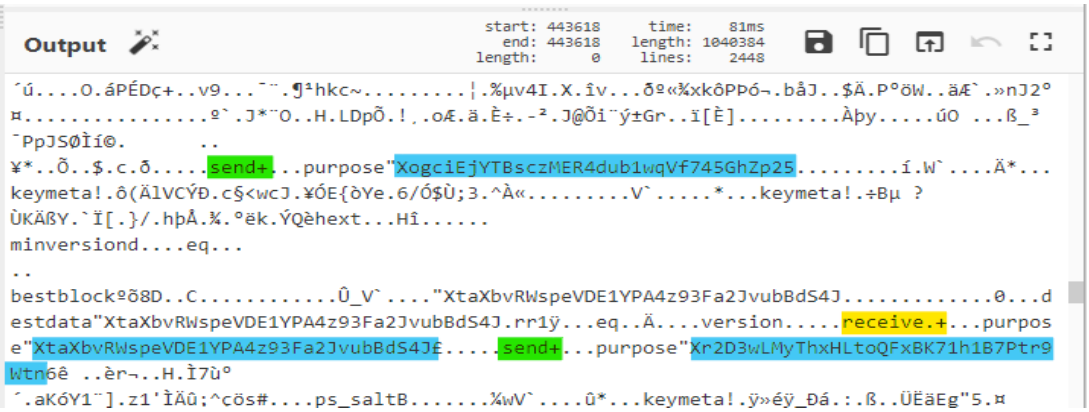


Figure 69. D. Core – Disk analysis. Correlation between sending and receiving addresses.

As the Dash official documentation mentions, if the user does not make use of the *encrypt wallet* option from the wallet application, the seed passphrase will be stored in plain text in the *wallet.dat* file. Figure 70 depicts the passphrase obtained after the execution of the command *dumphdinfo* in case 5 of the memory analysis.

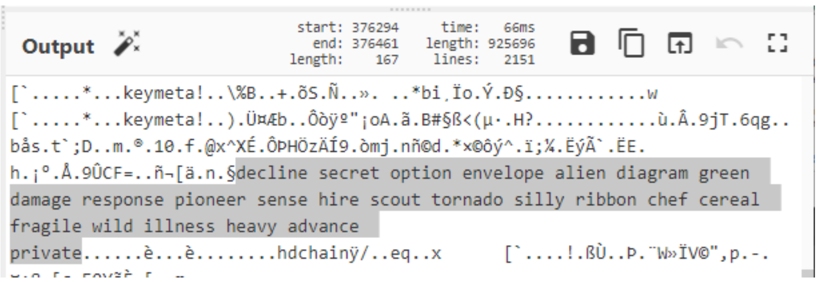


Figure 70. D. Core – Disk analysis. Mnemonic phrase located in plain text on wallet.dat file.

Later, in case 6, the wallet was encrypted, and the passphrase was not present or at least it was not in plain text.

The second file contains general debug information about the application but also contains the IDs of the transactions part of the "Coin Join" process required to make a private send transaction. Figure 71 shows the details.

```
Line 67580: 2021-03-21 08:27:14 AddToWallet 87b7cff94ea3cc4df355e058a40d5c0ea077237f097dd46e5679effb9ffee822  update
Line 67581: 2021-03-21 08:27:14 AddToWallet 6f94d5924a7303c084e6aed4af4db018570eabc801fb9f5bd4b58e83cc89dd4f  update
Line 67614: 2021-03-21 08:27:51 AddToWallet d1309d50747902bd18285f79ee93015f736363c7a74ec5eee75a2f897a147317  new
Line 67616: 2021-03-21 08:27:51 AddToWallet f48efedbe18bb0b1fa15fe1d1d86c9bbdb7eb7d8c41949066d1bee5bebd86d19  new
Line 67627: 2021-03-21 08:28:17 AddToWallet 5b48197e818e5a15a662093b8edbb0855ced7778f83f6d1f0643f9c688461080  new
Line 67637: 2021-03-21 08:29:12 AddToWallet d1309d50747902bd18285f79ee93015f736363c7a74ec5eee75a2f897a147317  update
Line 67638: 2021-03-21 08:29:12 AddToWallet f48efedbe18bb0b1fa15fe1d1d86c9bbdb7eb7d8c41949066d1bee5bebd86d19  update
Line 67639: 2021-03-21 08:29:12 AddToWallet 5b48197e818e5a15a662093b8edbb0855ced7778f83f6d1f0643f9c688461080  update
Line 67654: 2021-03-21 08:29:29 AddToWallet 6701cfb31f32d2a5320ade6afd611ab3610f14b1112aa9473302b7bcfce4eabd  new
Line 67668: 2021-03-21 08:29:47 AddToWallet 334fd34363677d6cb3a83b4dd13c2fa79427b12bb616b308c9fb50a36527de33  new
Line 67687: 2021-03-21 08:31:24 AddToWallet 334fd34363677d6cb3a83b4dd13c2fa79427b12bb616b308c9fb50a36527de33  update
Line 67688: 2021-03-21 08:31:24 AddToWallet 6701cfb31f32d2a5320ade6afd611ab3610f14b1112aa9473302b7bcfce4eabd  update
Line 67715: 2021-03-21 08:31:58 AddToWallet fe45871ccabc189a915d2d4a090e6a6a2cdc9daf9ef65c35e66f204ce7685022  new
Line 67724: 2021-03-21 08:32:18 AddToWallet 67fce4bf308918d48bc9b218fafad39ad4dc4451c555bf95e19576c87ea32aa8  new
Line 67748: 2021-03-21 08:40:01 AddToWallet fe45871ccabc189a915d2d4a090e6a6a2cdc9daf9ef65c35e66f204ce7685022  update
Line 67749: 2021-03-21 08:40:01 AddToWallet 67fce4bf308918d48bc9b218fafad39ad4dc4451c555bf95e19576c87ea32aa8  update
Line 67763: 2021-03-21 08:40:15 AddToWallet 2703a81f642dbc46054b7e4424332b6ccdce005e2614e49dbb272265e982ed7f  new
Line 67772: 2021-03-21 08:40:36 AddToWallet 68153ee636a98dff43585f932cc9851bdfab310c574452dfe45e4a4d00af70a5  new
Line 67779: 2021-03-21 08:40:45 AddToWallet 2703a81f642dbc46054b7e4424332b6ccdce005e2614e49dbb272265e982ed7f  update
Line 67797: 2021-03-21 08:43:08 AddToWallet 68153ee636a98dff43585f932cc9851bdfab310c574452dfe45e4a4d00af70a5  update
Line 67835: 2021-03-21 08:45:44 AddToWallet 0fb2f2f0aa1a925840f7af278a536db0ab800f921cd209adb857bbcf787b038a  new
Line 67837: 2021-03-21 08:45:44 AddToWallet 0fb2f2f0aa1a925840f7af278a536db0ab800f921cd209adb857bbcf787b038a
```

Figure 71. D. Core – Disk analysis. Debug.log file showing the transaction IDs.

Finally, no private keys or seed phrase was found.

4.3.3 Network Files

These files do not contain much valuable information since all network traffic is encrypted by the wallet application.

Only DNS traffic is observable, and this traffic goes to the Dash seeders or nodes that are connected in the Dash network and is depicted in Figure 72.

```
Queries
  x1.dnsseed.dash.org: type A, class IN
Answers
  x1.dnsseed.dash.org: type A, class IN, addr 136.243.29.222
  x1.dnsseed.dash.org: type A, class IN, addr 194.135.83.60
  x1.dnsseed.dash.org: type A, class IN, addr 3.133.151.167
  x1.dnsseed.dash.org: type A, class IN, addr 104.248.212.101
  x1.dnsseed.dash.org: type A, class IN, addr 45.32.157.229
  x1.dnsseed.dash.org: type A, class IN, addr 176.223.139.123
  x1.dnsseed.dash.org: type A, class IN, addr 45.32.243.157
  x1.dnsseed.dash.org: type A, class IN, addr 212.24.104.235
  x1.dnsseed.dash.org: type A, class IN, addr 198.27.69.190
  x1.dnsseed.dash.org: type A, class IN, addr 85.209.241.220
  x1.dnsseed.dash.org: type A, class IN, addr 85.209.242.9
  x1.dnsseed.dash.org: type A, class IN, addr 107.170.196.35
```

Figure 72. D. Core – Disk analysis. DNS traffic to the Dash seeders.

4.4 Dash Electrum

In this section the light version of the Dash wallet software will be studied to identify forensic artefacts can be obtained. Figure 73 depicts the direction and order followed in each transaction between the VMfullnode, VMlite and the iPhone to obtain the forensic images.

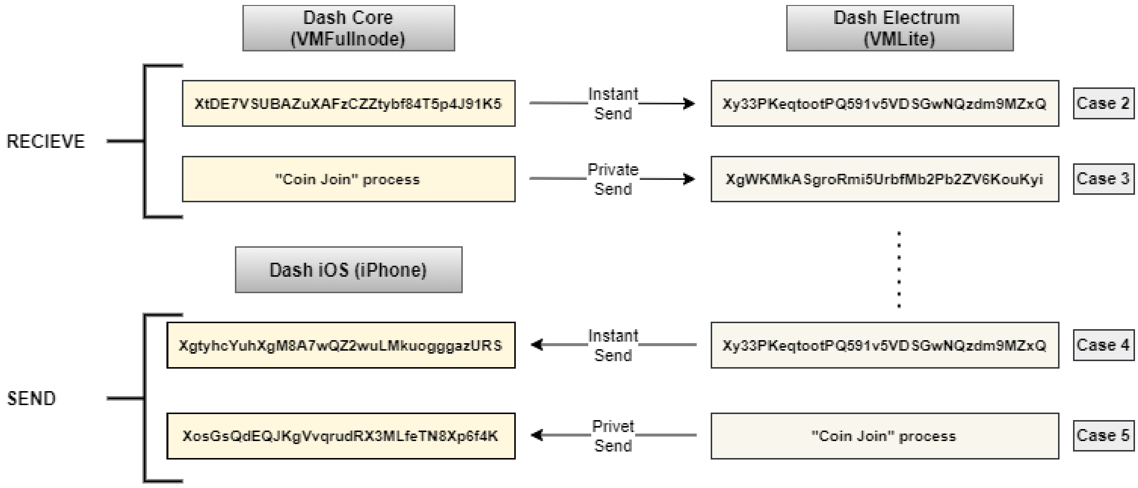


Figure 73. The direction of transactions between Dash Core, Dash Electrum, and iPhone device.

As it was explained in the case studies section, case 1 and 6 do not encompass transactions; that is why they are not present in Figure 73.

4.4.1 Memory Images

4.4.1.1 Case 1

Unlike the previous wallet applications, Dash Electrum allows the user to choose the name and path of the wallet file. Figure 74 illustrates the wallet was named *testttu_wallet.*

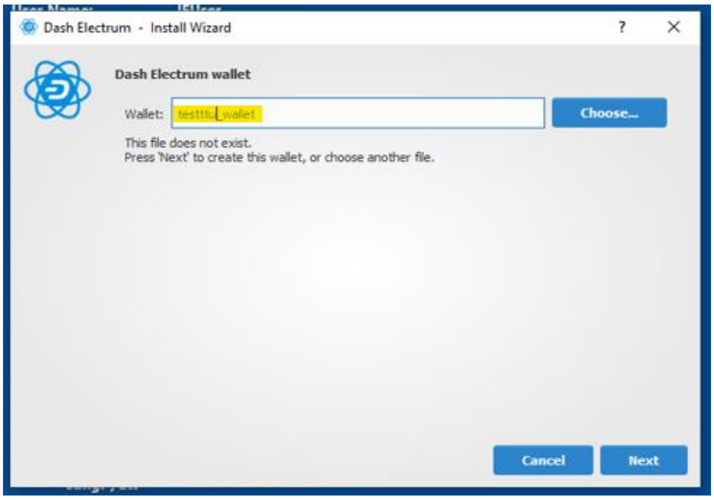


Figure 74. D. Electrum. Case 1. The name of the wallet file selected by the user is testttu_wallet.

Likewise, Figure 75 shows the mnemonic phrase automatically generated by the wallet application and composed of 12-words.

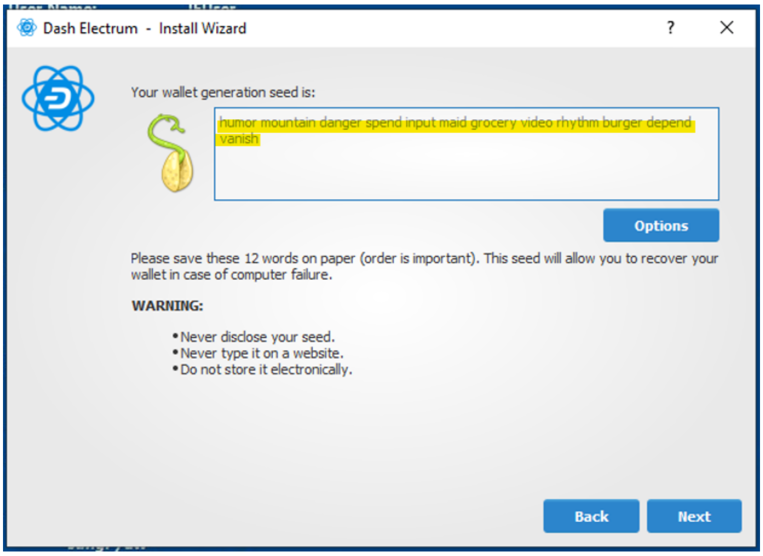


Figure 75. D. Electrum. Case 1. The mnemonic phrase generated automatically for the wallet application.

The structured analysis showed the installation path of the application but is not relevant for a forensic investigation. On the other hand, the unstructured analysis shows a list of addresses created automatically by the wallet. Figure 76 shows the list of addresses.

Figure 76. D. Electrum – Mem. analysis. Case 1. List of addresses reserved by the wallet application.

There was no evidence of the password used to encrypt the wallet or the mnemonic phrase that was given automatically by the application.

4.4.1.2 Case 2

The structured analysis shows information in the dumped PID 420. First, information about the transaction in the following order: recipient address, amount, message, and time. For instance Figure 77 depicts the transactions as follows *dash:**Xy33PKeqtootPQ591v5VDSGwNQzdm9MZxQ**?amount=0.15&message=From%20VMfull%20to%20VMlite.%20Cas1&time=1616618267&exp=86400?3*.

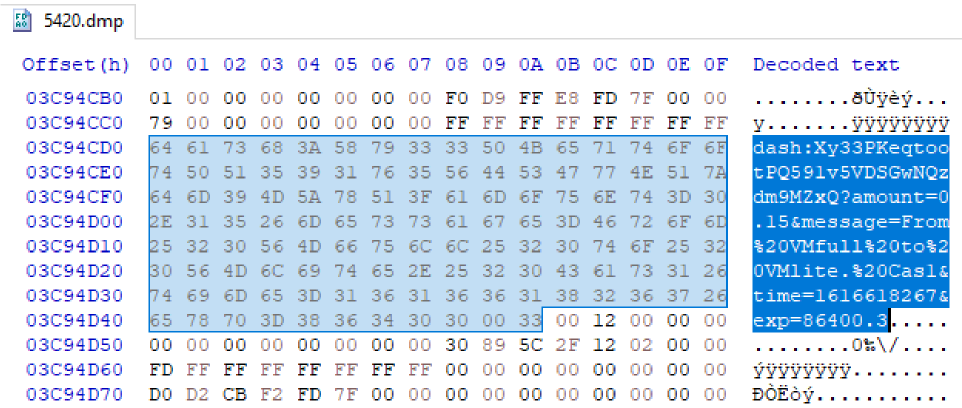


Figure 77. D. Electrum – Mem. analysis. Case 2. Information about the first incoming Instant Send transaction.

Second, it is possible to differentiate between the *change* addresses depicted in Figure 78 while the receiving addresses are illustrated in Figure 79.

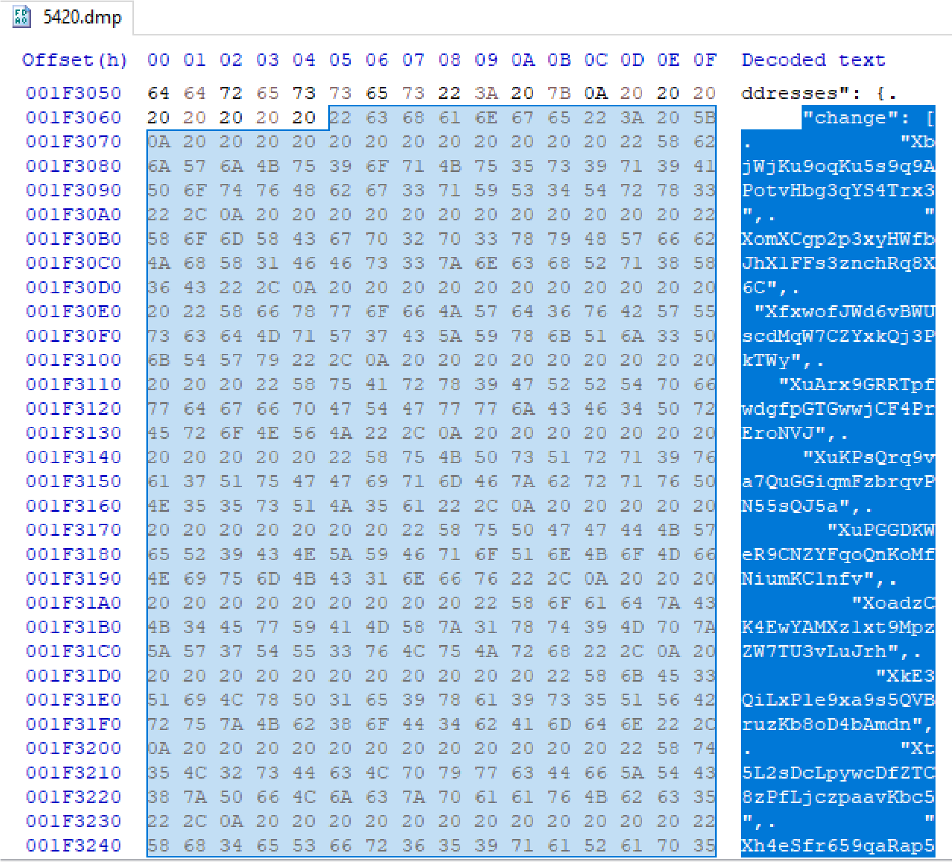


Figure 78. D. Electrum – Mem. analysis. Case 2. List of change addresses.

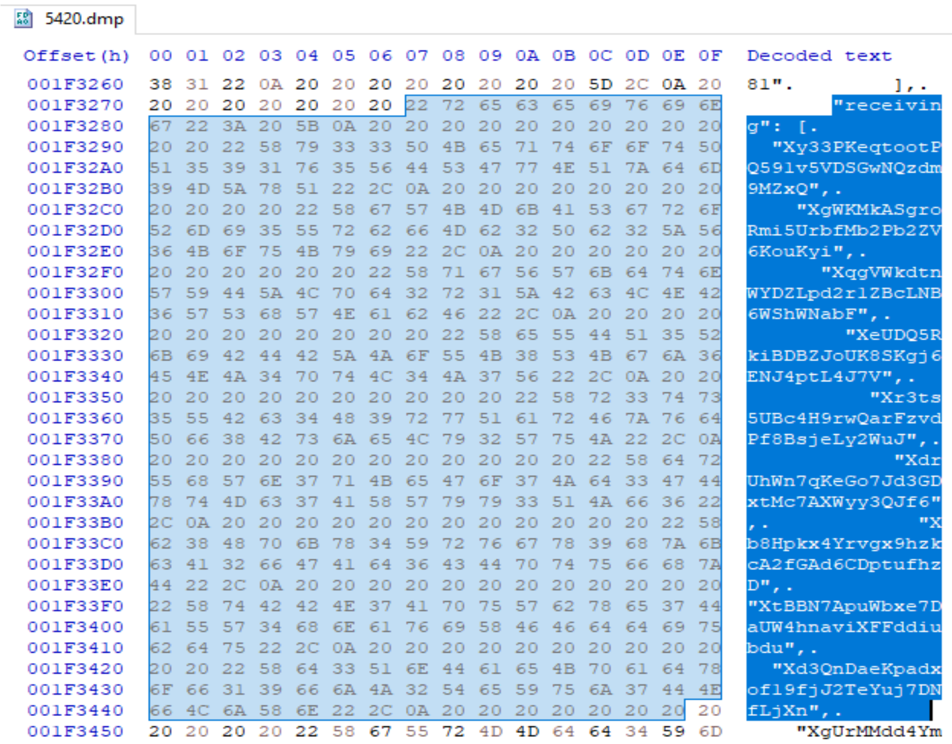


Figure 79. D. Electrum – Mem. analysis. Case 2. List of receiving addresses.

Next, another interesting finding illustrated in Figure 80 was the seed wallet in Base64 format, but when it was decoded, this seems to be encrypted.

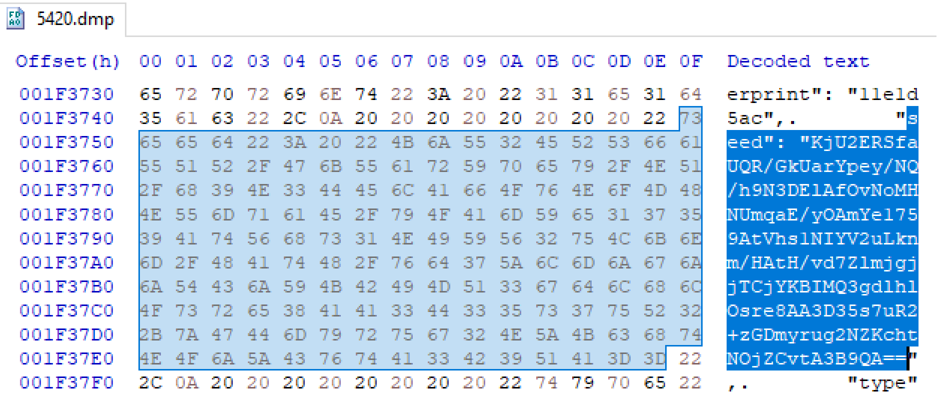


Figure 80. D. Electrum – Mem. analysis. Case 2. Seed wallet in Base64 format.

Finally, the xprv and the xpub were also found. The first one is in Base64 format, and when it was decoded, it did not show the real value since it is encrypted. However, the second one is cleartext, and it can be used to generate more addresses. Figure 81 illustrates xprv and xpub.

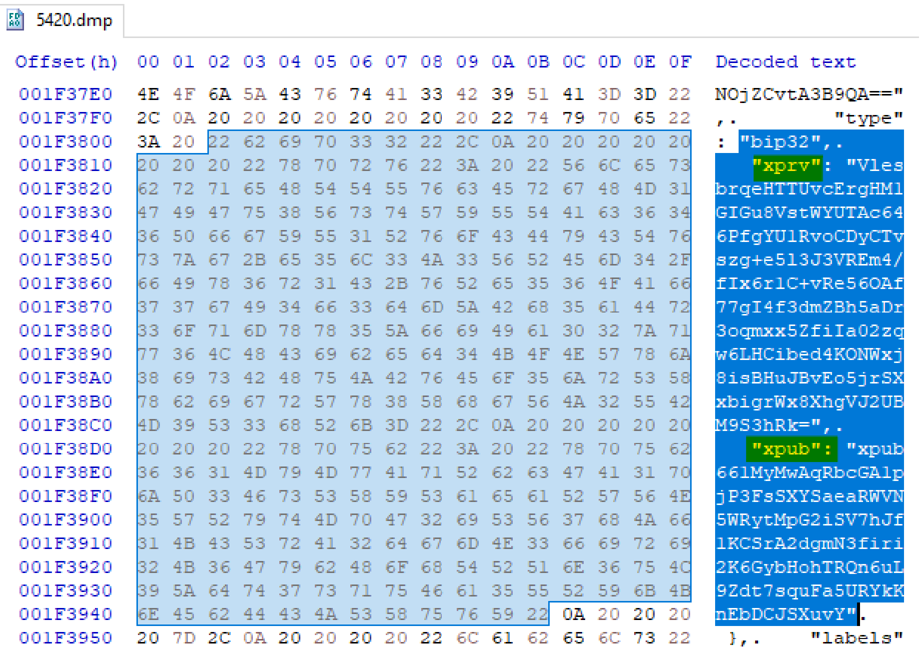


Figure 81. D. Electrum – Mem. analysis. Case 2. ºXprv and xpub found in PID 5420.

The unstructured analysis showed the addresses that are also depicted in Figure 75 from case 1, but there was no evidence of the sending address. However, the transaction ID *b13ba4f5e4be8093f052dc679c86027d737706f0c5bd5798e504d7ba1f813cb5* was also found but because this was known beforehand. However, up until this point, a keyword to easily identify a transaction ID has not been found yet. Figure 82 depicts the finding.

Figure 82. D. Electrum – Mem. analysis. Case 2. First incoming transaction ID.

No private keys nor the password to decrypt the wallet or seed phrase were found.

4.4.1.3 Case 3

The structured analysis shows information about the transaction in the dumped PID 5690. Figure 83 illustrates the message *Case3. From VMfull to VM lite* included in the transaction, and the recipient address *XgWKMkASgroRmi5UrbfMb2Pb2ZV6KouKyi* separately in Figure 84.

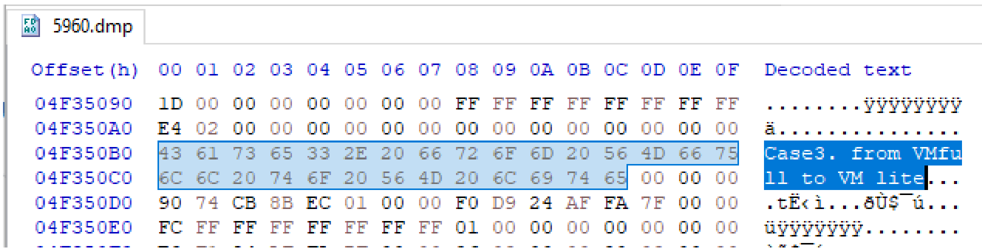


Figure 83. D. Electrum – Mem. analysis. Case 3. The message included in the second incoming PS transaction.

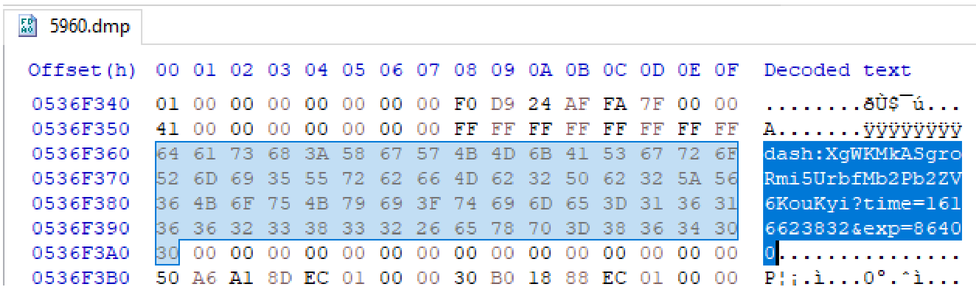


Figure 84. D. Electrum – Mem. analysis. Case 3. Recipient address of second incoming PS transaction.

However, the information shown in Figure 84 was dispersed in the memory file, making it difficult to determine that they belong to the same transaction.

The unstructured analysis shows the addresses belonging to the local wallet, as it was shown in Figure 75, but no sending address of the transactions made until this point were present. Also, it is possible to see the transaction ID, but it was again separated from the rest of the transaction information, making it difficult to associate with the transaction itself. Figure 85 shows the details.

Figure 85. D. Electrum – Mem. analysis. Case 3. Transaction ID from second incoming PS transaction.

The analysis did not show any evidence of private keys nor the password used to open the wallet application.

4.4.1.4 Case 4

The structured analysis shows information related to the transaction, such as the message *Case4. from vmlite to iPhone* and the recipient address *XgtyhcYuhXgM8A7wQZ2wuLMkuogggazURS* depicted in Figure 86. However, there was no trace of the amount and sending address.

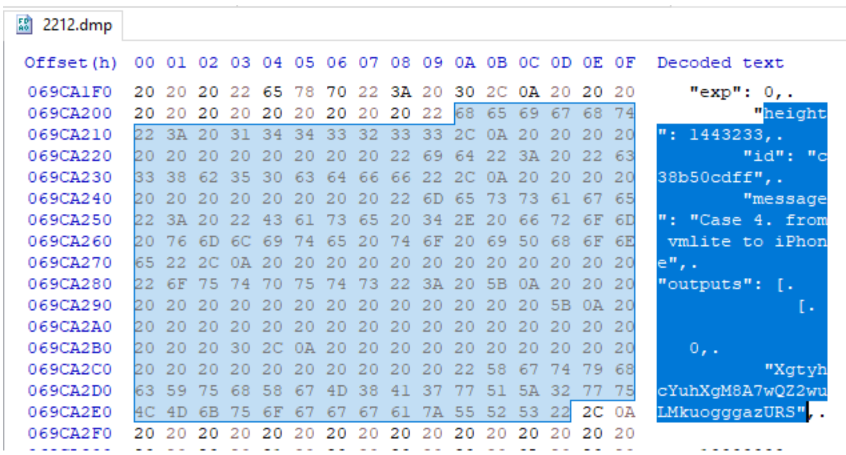


Figure 86. D. Electrum – Mem. analysis. Case 4. Message and recipient address included in the third outgoing IS transaction.

The unstructured analysis shows in Figure 87, the transaction ID *2839983d0e43a1ef6b4e2d37baecaaed6f542bde334414ccb4b1cf20f10514d1* but again separated from the rest of information from the transaction making it difficult to correlate.

```
8296447-1362d474d384b311245f1f0363b18e33c7e0fec917f2560a43ac9e285fda7433:0
8296448-b13ba4f5e4be8093f052dc679c86027d737706f0c5bd5798e504d7ba1f813cb5
8296449:2839983d0e43a1ef6b4e2d37baecaaed6f542bde334414ccb4b1cf20f10514d1:0
8296450-c82bc29
8296451-88bc0b6
```

Figure 87. D. Electrum – Mem. analysis. Case 4. The ID of the third outgoing IS transaction.

No trace of private keys or wallet password was found.

4.4.1.5 Case 5

The structured analysis shows in Figure 88 the message *Case 5. from vmlite to iphone. Private Send* used in the transaction. The message was located without any other information that will allow the investigator to relate it to the transaction itself. In this case, it was found due to the previous knowledge of it.

```
5488.dmp

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
078F6B30  E4 3B 7F BC 3A 01 00 00 00 00 00 00 00 00 00 00  ä;.¼:...........
078F6B40  43 61 73 65 20 35 2E 20 66 72 6F 6D 20 76 6D 6C  Case 5. from vml
078F6B50  69 74 65 20 74 6F 20 69 70 68 6F 6E 65 2E 20 50  ite to iphone. P
078F6B60  72 69 76 61 74 65 20 53 65 6E 64 00 00 00 00 00  rivate Send.....
078F6B70  00 00 00 00 00 00 00 00 30 77 01 C5 3A 01 00 00  ........0w.Å:...
078F6B80  FC FF FF FF FF FF FF FF 00 00 00 00 00 00 00 00  üÿÿÿÿÿÿÿ........
```

Figure 88. D. Electrum – Mem. analysis. Case 5. The message included in the fourth PS transaction.

In like manner, the unstructured analysis shows the transaction ID *923be98575fd49b07ee0da0393eaa2e85e341675c27ff35a410e22474f415cbd* separated from the rest of the information about the transaction as it was shown in the previous cases.

```
27262346-1s07,
27262347:Found PrivateSend 923be98575fd49b07ee0da0393eaa2e85e341675c27ff35a410e22474f415cbd
27262348-16:f8e57efeeefca37bb953298e26931965505b23771646a69dcee3e454d12d518a:1:1616663178
27262349-K:VB
```

Figure 89. D. Electrum – Mem. analysis. Case 5. The ID of the fourth outgoing PS transaction.

4.4.1.6 Case 6

The following options were explored using the wallet application:

- Make a backup of the wallet called *"testttu_wallet.backup"*.

- See the HD seed. It was not saved on disk.
- Export the private keys in CSV format in the file called electrum-dash-private-keys.csv.
- Execute the commands *electrum help*, *help*, *history*, *version*, *list_wallets* and *listaddressesss* from the embedded console of the wallet application.

By default, the application adds the .backup extension when a backup file is created. From the point of view of an investigator, this information is helpful because he/she can make searches of any possible backup on the entire disk. Figure 90 shows the MFT record with the creation of the backup file *testttu_wallet.backup*.

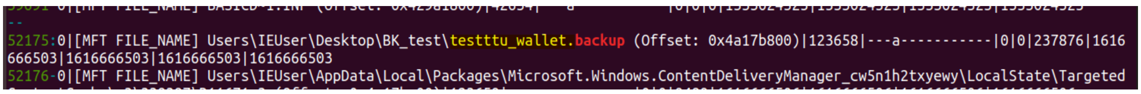
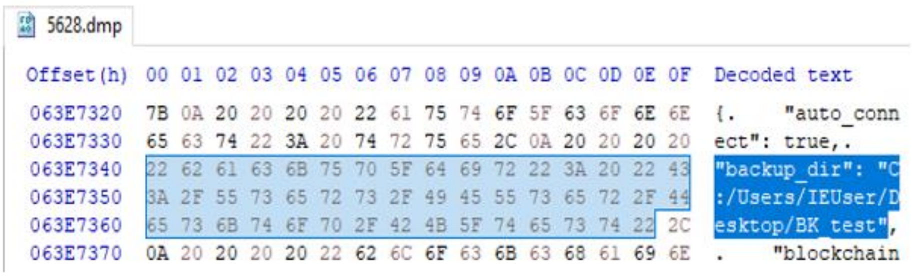


Figure 90. D. Electrum – Mem. analysis. Case 6. MFT record of the creation of the backup file.

Another interesting finding related to wallet backup is that the application leaves traces of the destination path by searching the parameter *backup_dir*. The details are illustrated in Figure 91.



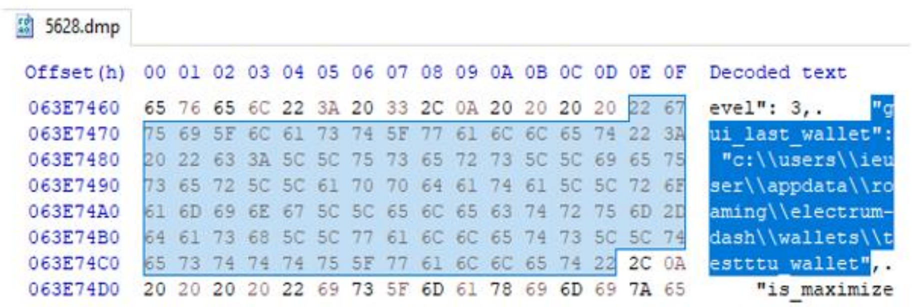


Figure 91. D. Electrum – Mem. analysis. Case 6. The path where the wallet backup was saved.

Also, in the dumped PID 5626 from the offset 0x041d7f90 to 0x04272faq, information about the wallet can be obtained in JSON format. When this text was exported to a file, it showed in total 11151 lines of information that includes most of the previous information displayed from case 1 to 5. Figure 92 depicts the most relevant parts of the mentioned file.

Figure 92. D. Electrum – Mem. analysis. Case 6. Wallet information in JSON format showing addresses and transaction details.

The relevant data mentioned above includes the list of *change* and *receiving* addresses created by the wallet application. Likewise, the transaction ID and the message included in this one is under the *lables* section. Finally, more details about the transactions are shown under the section *payment_requests*.

The backup of the private keys can be observed in the MFT record illustrated in Figure 93, but since there is not a standard name or extension given by the application for this file, it would be challenging to identify for the investigator.

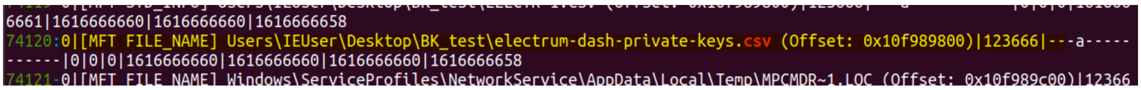


Figure 93. D. Electrum – Mem. analysis. Case 6. MFT record of the creation of CSV file with the private keys.

Finally, the executed commands such as *help*, *electrum help*, *history*, *version*, *list_wallets* and *listadddresses* on the embedded console were found under the parameter *qt-console-history* and illustrated in Figure 94. This finding is very interesting because it allows the investigator to determine what other actions were taken by the user. However, the results of these commands were not located in the memory file.

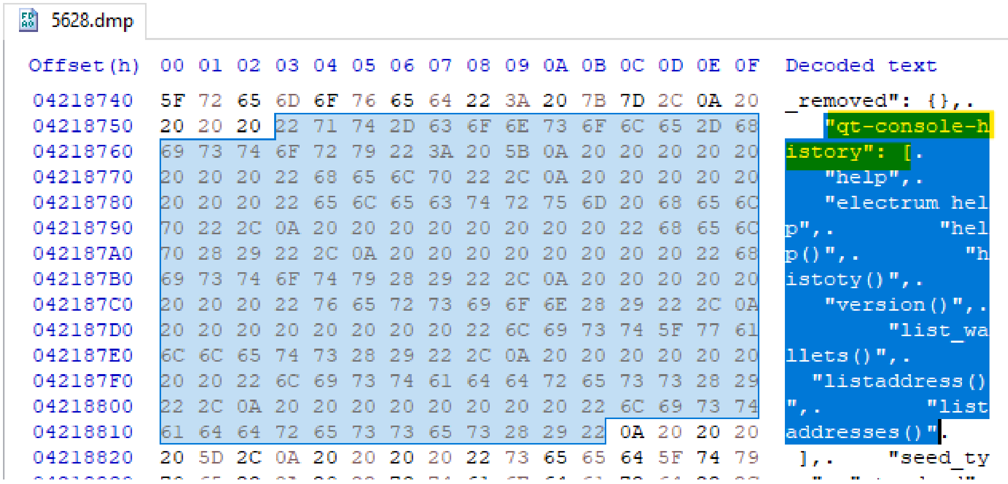


Figure 94. D. Electrum – Mem. analysis. Case 6. Parameter qt-console-history shows the commands executed by the user.

4.4.2 Disk Files

Relevant information was found in the following files:

- The *config* file stores the basic configuration of the local wallet.
- The *recent_servers* file stores the nodes or network where the local wallet connects.
- The *Wallets* folder contains the wallets files.

The first file contains information such as what was the latest used wallet shown in the parameter *gui_last_wallet* and what were the recently opened wallets in the parameter *recently_open*. For instance, Figure 95 depicts that the *testttu_wallet* was the latest opened wallet from the applications while the *wallet_2*, *wallet_3* and *testttu_wallet* were recently opened.

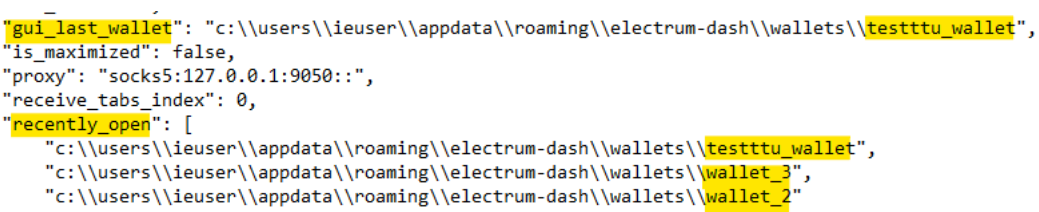


Figure 95. D. Electrum – Disk analysis. Config file content.

The second file contains the list of nodes where the wallet is connected to make transactions. In this file an onion domain can also be observed, that is used by the application. Figure 96 shows the details.

```
[
    "178.62.234.69:50002:s",
    "electrumx-mainnet.dash.org:50002:s",
    "hyhwaxmckqakwjde.onion:50002:s",
    "165.232.38.144:50002:s",
    "drk.p2pay.com:50002:s"
]
```

Figure 96. D. Electrum – Disk analysis. Nodes where the wallets connect.

The last folder contains the wallet files that have all the information regarding the wallet, such as transaction history and addresses, as was explained in previous cases. The path by default is *C:\Users\[User]\AppData\Roaming\Electrum-DASH\wallets*, but this can be defined by the user. Figure 97 illustrates the details.

```
C:\Users\IEUser\AppData\Roaming\Electrum-DASH\wallets>dir
 Volume in drive C is Windows 10
 Volume Serial Number is B4A6-FEC6

 Directory of C:\Users\IEUser\AppData\Roaming\Electrum-DASH\wallets

03/27/2021  09:14 PM    <DIR>          .
03/27/2021  09:14 PM    <DIR>          ..
03/27/2021  09:14 PM           237,876 testttu_wallet
03/27/2021  07:42 PM             2,504 wallet_2
03/27/2021  07:50 PM             2,524 wallet_3
               3 File(s)        242,904 bytes
               2 Dir(s)  21,203,910,656 bytes free
```

Figure 97. D. Electrum – Disk analysis. Default path where the wallet files are stored.

4.4.3 Network Files

The network traffic captured during the 6 cases was encrypted, and there was no evidence of DNS queries such as the Dash Core showed. The reason for this is because the Tor service starts automatically with the operative system and encrypts the traffic even when the wallet application has not been started yet.

To identify what exactly the wallet application does without Tor, the service was disabled, and then the wallet initialized. The network traffic shows connections to the IP addresses *178.62.234.69* and *165.232.38.144*, and DNS queries to the domains *hyhwaxmckqakwjde.onion*, *electrumx-mainnet.dash.org* and *drk.p2pay.com*. These domains and IP addresses are configured in the *recent_servers* file shown in Figure 94.

Finally, the following table summarizes the findings in the four wallets analysed.

Table 7. Findings in the four wallet applications analysed.

Artefacts		**Zcash**		**DASH**	
		Fullnode	**Lite**	**Core**	**Electrum**
Memory	Local addresses	✔	✔	✔	✔
	External addresses	✔	✖	✖	✖
	Transaction ID	✔	✔	✔	✔
	Transaction amount	✔	✖	✔	✖
	Transaction timestamp	✔	✖	✔	✔
	Transaction fee	✔	✖	✖	✖
	Private keys	✔	✔	✖	✖
	Mnemonic phrase	✖	✔	✔	✖
	Wallet password	✖	✔	✖	✖
	Memo field	✔	✖	N/A	N/A
	Seed (Base64)	✖	✖	✖	✔
	Xpriv	✖	✖	✖	✔
	Xpub	✖	✖	✖	✔
	Transaction message	N/A	N/A	✖	✔
Disk	Transaction ID	✔	✖	✔	✖
	Local addresses	✔	✖	✔	✖
	External addresses	✖	✔	✔	✖
	Memo field	✖	✔	N/A	N/A
Network	DNS queries	✔	✔	✔	✔
	IP connections	✖	✖	✖	✔

5 Discussion

This section will elaborate on the results obtained during the analysis of the four wallet applications starting from the Zecwallet Fullnode, later the Dash Core, then Zecwallet Lite and finally the Dash Electrum. Moreover, a comparison with a previous study [36] that also analyses other privacy-oriented cryptocurrency wallets will be made.

The findings presented in this section are accompanied by keywords that will make the searches straightforward for the investigator or the person that will use this document. For instance, it can be mentioned that some information was found with the keyword *txid*, then the investigator can use this *txid* word or keyword to find relevant information.

Zecwallet Fullnode does not require the user to create a password to open the application or before spending the funds. Transparent and shielded addresses from the local wallet were present in memory in all the cases, but only case 4 (outgoing private address between the VMfullnode and iPhone) showed the sending address, and it was a straightforward identification using the regular expressions listed in Appendix A. Furthermore, information about the transactions is present in JSON format, which makes it effortless for the investigator to identify such information. The AddressBook.json file that stores the user's contacts was spotted under the keyword *label*. One way to locate the transaction ID is through the *AddToWallet* keyword, but this one will show only the ID. Another way to find the transactions with more detail is through the keyword *txid* that will show not only the transaction ID but the amount, the fee, the confirmations, the memo field, the timestamp and the sending and/or receiving address.

Once the investigator achieves obtaining the transaction ID, this can be used to gather more information regarding the transaction on the blockchain. However, the investigator must consider what type of transaction was performed under that transaction ID. For instance, if the ID belongs to a private transaction, the blockchain will show only general information such as the date, the ID, and the fee. On the other hand, if the ID belongs to a public transaction, besides the general information on the blockchain, this also will show the recipient address and the amount, giving the investigator more clues about the destination of the funds. The Zcash transaction types are illustrated in Figure 1.

An important thing to point out regarding the memo field is that this will be visible only to the recipient address. For instance, in case 2, the private transaction between iPhone and VMfullnode, the memo field content is visible from the wallet recipient and the memory acquisition file. However, in case 4 and 6, the private and shielding transaction, respectively, the memo content is visible again from the recipient's wallet, but in memory, even when the field is present, and this one is decoded, the information does not return the original message. Table 7 shows a summary of the presence of the memo field produced in cases 2, 4 and 6.

Table 8. Memo field presence in memory files.

Case	**Transaction type**	**Direction**	**Is the message present in the recipient's wallet?**	**Is the memo field present in-memory file?**	**Is the memo field readable after it was decoded?**
Case 2	Private	Receive	Yes	Yes	Yes
Case 4	Private	Sent	Yes	Yes	No
Case 6	Shielding	Sent	Yes	Yes	No

Another important aspect to mention regarding the memo field is that this becomes relevant when its content has any information that can identify the source or destination of the transaction. In the discussion of the Lite version, this will be explained.

Private keys and their corresponding addresses (public keys) were found in memory only in case 8 after the execution of command *z_exportwallet*, and these can be obtained using the regular expression listed in Appendix A for the case of transparent addresses and with the keyword *secret-extended-key* for the case of shielded addresses.

Besides, the MFT records retain not only valuable metadata on files application but also shows their content. For instance, the user's contacts stored in the file AddressBook.json can be seen in these on the MFT records and are illustrated in Figure 11. In like manner, the default configuration and the additional parameter added to execute the z_exportwallet command in case 8 are also observed from the MFT records and illustrated in Figure 18. Nevertheless, only metadata such as the access date is shown in the case of the files zecwallet_transactions.csv and zcash-cli.exe as illustrated in Figures 6 and 19, respectively.

The artefacts obtained from the disk files did not differ much from the ones collected in memory. The debug.log file shows the transaction IDs if searches are done with the keyword *AddToWallet*. One way to identify the outgoing transaction is by the combination of the keywords *txid* plus *z_sendmany*, which is an RPC command used to send money. The external IP address used by the computer was also found in this file under the keyword *advertising*. Furthermore, the wallet.dat showed the transparent addresses under the keyword *purpose*. In like manner, files AddressBook.json and zcash.conf were also found with the same content spotted in the MFT records in memory.

In other words, if the investigator succeeds in recovering the wallet.dat file, that would be good progress for the investigation since restoring this file to another computer will allow the investigator to access the entire wallet transaction history, funds, and private keys from transparent and private addresses. So, investing time and effort in this part of the analysis would save additional effort.

On the other hand, **Dash Core** does not require the user to use a password before opening the wallet application or spending the funds. In memory, the transaction ID was found under the keyword *AddToWallet*, but no more information regarding this one was shown. Moreover, the sending address was found with the regular expression listed in Appendix A and under the keyword *Address*. Only case 3 (outgoing IS transaction from VMfull to iPhone) showed some details regarding the transaction using the keyword *Amount*. In general, correlating the ID, sending address, and the amount would be difficult for the investigator since this information is dispersed when analysing the memory file making no sense. However, the backup of the wallet was located with the keyword *wallet*, but this is a general keyword since it shows many results, but the trick is to look for the message illustrated in Figure 60 that shows the path and file name even when this can be different every time the user decides to create a new backup.

Likewise, in the memory file, the MFT records show evidence (as illustrated in Figure 66) that the file dash.exe has been used. This file has been identified only in case 5, where the user has obtained the mnemonic phrase from the wallet. Even when this finding does not represent information that can provide the investigator with clues regarding some transaction, it can tell the investigator that there exists the probability that in the memory acquisition, there is the presence of the 24-word mnemonic phrase to restore the wallet and access the complete information this contains. If that is the case, the 24-word mnemonic phrase can be found using the keyword *hdchain*. However, if the wallet is later encrypted (like in case 6), the mnemonic phrase will not be present in the wallet.dat file, and this option can be discarded.

Disk artefacts are also interesting since it is possible to see the addresses in the wallet.dat file. The local addresses can be found using the keyword *receives* while the external addresses with the *send* keyword. Figure 69, it is shown how these addresses can be found. Also, the 24-word mnemonic phrase was in the file; nevertheless, if the wallet file is encrypted, also the mnemonic phrase. The debug.log did not offer many details other than the transaction ID, and this can be found using the *AddToWallet* keyword, similar to the memory findings. Network findings showed information limited to DNS queries due to the traffic being encrypted by the application.

Even when the Dash Core wallet shows slight information in memory, by recovering the transaction ID, the investigator can obtain the full information from the blockchain considering that DASH works like Bitcoin regarding how data about the transaction is publicly available on the blockchain. However, if the transaction ID belongs to a private send transaction, even though the information is public, DASH uses the coin mixing technique precisely to provide anonymity to the users making it difficult to trace who was the real sender of the transaction. On the other hand, focusing the efforts on recovering the wallet.dat file or any backup of this file, or recovering the mnemonic phrase from memory, could be more beneficial since this would allow the investigator to obtain full access to the wallet information and private send transactions could be traced. Finally, if the wallet.dat file is recovered but was previously encrypted by the user, the investigator will still have access to the information, but the password will be required to spend the funds.

The **Zecwallet Lite** wallet application, when executed for the first time, automatically generates the mnemonic phrase composed of 24-words. These words are visible in memory only once, but after rebooting the VM, they were not present in the following cases. When the wallet was encrypted, the processes related to the wallet applications were not present during the memory analysis; nevertheless, it was still possible to identify evidence of the installation of the application in the MFT records. Likewise, the password used for the encryption of the wallet was found in the memory files of case 3 and 5, where the user inputs the password to spend the funds, and in case 5, where the user inputs the password to export the private keys. However, finding the password will be difficult for the investigator since there was not a keyword to make it identifiable or easy to locate.

Transparent and shielded local addresses were located using the regular expressions in Appendix A, while the transaction ID using the keywords *Added to wallet* and *txid*. Moreover, a way to identify an incoming transaction is by the keyword *receiving sapling output to,* but this one will show the recipient address and not the transaction ID. Private keys were not found during the analysis of the first four cases; however, these were present in memory when the user executed the *export all private keys* option in case 5 and the way to identify them was by using the regular expressions of Appendix A that will show the transparent and shielded addresses with their corresponding private keys.

The findings in the MFT records were similar to those in Zecwallt Fullnode. Again, the content of the AddressBook.json shows the user's contacts in memory as depicted in Figure 41, while in the case of the file zecwallet_transaction.csv only the headers are only displayed without any content, and it is illustrated in Figure 42.

Disk files analysed show interesting information related to the transactions. In the zecwallet-light-wallet.dat file, local transparent addresses are present and illustrated in Figure 50; also, the content of the memo field is displayed. Even though it is not possible to determine to what transaction the message belongs to just by reading it, it would be possible to identify the sending address as long as the user marks the *reply-to* check option while doing the transaction, as is illustrated in Figure 52. In Figure 55, it can be seen how the memo field works and why this is relevant in the investigation. The content of this field is visible only from the recipient wallet. Therefore, if the user marks the *Include Reply-To address* before sending the funds to the recipient address, this will include the sending address to the memo field. If that is the case, in the recipient's wallet, this message will be shown from the memory acquisition (See Table 8.) and/or disk acquisition. Shielded addresses or private keys were not found during the analysis. Furthermore, zecwallet-light-wallet.debug.log file shows the transaction ID when searches are done using the keyword *Txid* with the first letter capitalized. Also, the AddressBook.json shows the same content displayed in the memory analysis. Results on network file acquisitions do not change much if compared with the Fullnode version since the network traffic is also encrypted and only DNS queries are visible.

Unlike the Fullnode version, the Lite version shows less information regarding the transactions in memory files making the transaction ID the most valuable information recoverable from these files. As it was also explained in the Fullnode version, the investigator must consider the type of transaction used by the user to understand how much information can be obtained from the blockchain. Nevertheless, if the investigator succeeds in recovering the zecwallet-light-wallet.dat from the disk, this could be restored, and the investigator could have access to the entire wallet information, but still, the password used to encrypt the wallet will be needed to spend the funds or export the private keys.

Dash Electrum wallet application by default has the encrypt wallet option marked when the user is following the installation steps, inducing him/her to create a password from the very beginning of the creation of the wallet. The password, the mnemonic phrase or any private key related to the addresses were not found during the entire analysis. However, lots of information in JSON format was found in memory, as is illustrated in Figure 92 from case 6 (exploring additional options from wallet application). From that information, the investigator can easily differentiate between the change addresses and the receiving addresses. Likewise, transactions can be found under the section *labels* and *payment_requests*. Also, from the section *keystore*, information such as xpriv and xpub can be identified.

Despite the user can choose the path and name of the file when doing a backup of the wallet, making it challenging for an investigator to guess where the file was saved, the application leaves a trace in memory about what the user's selection was using the key *backup_dir*. Likewise, this information can also be seen in the MFT records but trying to find the backup file from there would be challenging for the investigator. A similar situation happens when exporting the private keys in CSV format; since no trace was found in memory, again looking for that information in the MFT records will be demanding. On the other hand, the set of commands executed from the embedded console of the wallet application (case 6) can be found in memory with the keyword *qt-console-history*; nevertheless, the results were not located.

Information obtained from the disk file analysis gives valuable information to the investigator. In the config file, illustrated in Figure 94, can be seen what the latest open wallet was and what were the recent open wallet files. With this input, the investigator can make searches of those files in the entire disk. The content of the recent_servers file shows the IP addresses and domains where the wallet application will connect, also it includes the .onion domain used by the application when the user accepts to install the Tor proxy. The wallets folder is considered valuable information since it is the default location where the application stores the wallet files; however, this can change based on the user's decision when this creates the wallet for the first time.

Since the wallet application installs the Tor proxy, the entire network traffic is encrypted even when the application is not running. To identify the application's behaviour in terms of network traffic, the Tor service was disabled from the Windows Services option, and the results showed that all the network traffic goes to the IP address and domains listed in the file recent_servers.

Despite inducing the user to create a password from the beginning to protect the wallet and not showing the password or any private keys, during the six memory files obtained in the forensic acquisition, the Dash Electrum wallet application stores a considerable amount of data in JSON format in memory regarding the transactions and addresses. Likewise, even if the investigator achieves recovering the wallet file, it would be necessary to have the password used the first time the wallet was created to have access to the information because, unlike the Dash Core version, Dash Electrum requires the user first to input the password to open the file. In other words, memory acquisition analysis becomes the most important part when dealing with this version of the wallet application.

When comparing the results obtained from a previous study focused on Monero and Verge, also considered privacy-oriented cryptocurrencies, the findings in memory files are similar to those found in the Zcash and Dash analysis [36]. For instance, the passphrase, transaction IDs, transaction amounts and mnemonic phrases were obtained from the forensic analysis. However, also some differences can be noticed, such as Zcash Fullnode presents the entire transaction in JSON format or Dash Electrum shows a considerable amount of information about the wallet that includes the transaction, addresses, xprv and xpub in plaintext.

Regarding the disk findings in Monero and Verge, the artefacts that can be obtained from there are also similar to those obtained in Zcash and Dash. This is because the applications encrypt and/or protect with an additional password (passphrase) the wallet file, which is the most important file in all the analysed wallets due to the fact that if the investigator gains access to it, it can be said that the case is solved since all the information can be found there. Also, the files that contain general debugging information are in plaintext and contain the IDs or, in some cases, the addresses that can be used later to do searches in the blockchain. Maybe the difference in this part is how the application works since Monero creates a text file that contains the addresses, and it is not present in Zcash and Dash. Network findings in these wallets are poor since all the applications encrypt the traffic and the only readable information are the DNS queries.

It can be said that there exist more similarities than differences because the application wallets work almost in the same way, protecting the information that is contained in the wallet or if the wallets use a mnemonic phrase to restore it in case the wallet is damaged, or if the application uses a password to allow the user to have access to the information or spend the funds. However, the big difference comes when talking about the protocol itself and how Zcash, Dash, Monero and Verge record the information of the transactions in the blockchain and what techniques they used to provide anonymity and privacy to the users.

6 Conclusion

This study has demonstrated the valuable forensic artefacts that can be obtained from Zecwallet Fullnode, Zecwallet Lite, Dash Core and Dash Electrum wallet applications. The analysis has shown that information can be collected from the structured analysis, which consists of the scanning of processes on memory files, and the unstructured analysis that consist of the use of regular expressions and keywords.

Most of the evidence collected during the analysis was obtained from the memory acquisition, meaning that in case of not being able to access the disk of the local computer for different reasons, the memory analysis will provide considerable information about the transaction history, contacts, etc. Therefore, this part of the study probably is the most relevant during the investigation.

Network analysis did not provide much information due to the traffic being encrypted, and information cannot be extracted or analysed. On the other hand, despite the disk analysis contributing with some interesting findings, it can not be compared with the amount of data that can be found in memory analysis. However, if the investigator manages to acquire the important files like the wallet itself and restores it in another computer, most of the investigation will be accomplished. But if the wallet requires a password to be opened, like is the case of the Dash Electrum, recovering the file will be useless since the password needs to be used to have access to the information.

The goal of the artefacts obtained during the memory, network, and disk forensic analysis; is to provide the investigator with helpful information that can be correlated in the blockchain to identify the source and destination of the involved parties after a transaction has been done. Moreover, facilitating the search for information with the provided keywords recollected during the study. Finally, the use of free tools during the entire analysis can be considered a limitation since there exist commercial tools such as EnCase, which specializes in forensic investigations, that could provide more information to the study.

6.1 Future Research

The future work can include the new versions of the wallet application and the versions available for Linux and macOS versions. Considering that operative systems work differently from each other since the filesystem they use is different, new artefacts could be obtained from the studies. In like manner, the multi-currency wallets that can store Zcash, Dash, Bitcoin and some others can also be part of future forensic analysis. Finally, the mobile versions for Android and iOS can be part of a future forensic analysis.

References

[1] "What Are the Most Traded Cryptocurrencies? | Plus500." https://www.plus500.com/Trading/CryptoCurrencies/What-are-the-Most-Traded-Cryptocurrencies~2 (accessed Apr. 05, 2021).

[2] J. P. Buntinx, "The Role of Cryptocurrency in Crime - Darknet Activity Soars » NullTX," *NullTX*, Jun. 08, 2018. https://nulltx.com/the-role-of-cryptocurrency-in-crime-darknet-activity-soars/ (accessed Apr. 05, 2021).

[3] R. Wolfson, "Tracing Illegal Activity Through The Bitcoin Blockchain To Combat Cryptocurrency-Related Crimes," *Forbes*. https://www.forbes.com/sites/rachelwolfson/2018/11/26/tracing-illegal-activity-through-the-bitcoin-blockchain-to-combat-cryptocurrency-related-crimes/ (accessed Apr. 05, 2021).

[4] "What is the Deep and Dark Web?," *www.kaspersky.com*, Jan. 13, 2021. https://www.kaspersky.com/resource-center/threats/deep-web (accessed Apr. 15, 2021).

[5] "INTERNET ORGANISED CRIME THREAT ASSESSMENT (IOCTA) 2020," *Europol*. https://www.europol.europa.eu/activities-services/main-reports/internet-organised-crime-threat-assessment-iocta-2020 (accessed Apr. 05, 2021).

[6] M. Morell, J. Kirshner, and T. Schoenberger, "Report: An Analysis of Bitcoin's Use in Illicit Finance," *The Cipher Brief*, Apr. 13, 2021. https://www.thecipherbrief.com/report-an-analysis-of-bitcoins-use-in-illicit-finance (accessed Apr. 16, 2021).

[7] "Alt-Right Groups and Personalities Involved In the January 2021 Capitol Riot Received Over $500K In Bitcoin From French Donor One Month Prior." https://blog.chainalysis.com/reports/capitol-riot-bitcoin-donation-alt-right-domestic-extremism (accessed Apr. 19, 2021).

[8] "Attorney General William P. Barr Announces Publication of Cryptocurrency Enforcement Framework," Oct. 08, 2020. https://www.justice.gov/opa/pr/attorney-general-william-p-barr-announces-publication-cryptocurrency-enforcement-framework (accessed Apr. 08, 2021).

[9] C. C. Editor, "PII - Glossary | CSRC." https://csrc.nist.gov/glossary/term/PII (accessed Apr. 21, 2021).

[10] E. Silfversten, M. Favaro, L. Slapakova, S. Ishikawa, J. Liu, and A. Salas, *Exploring the use of Zcash cryptocurrency for illicit or criminal purposes*. RAND Corporation, 2020. doi: 10.7249/RR4418.

[11] "FinCEN Advisory, FIN-2020-A006," p. 8.

[12] Michael Doran, "SANS Institute: Reading Room - Forensics." https://www.sans.org/reading-room/whitepapers/forensics/paper/36437 (accessed Apr. 10, 2021).

[13] A. Turner and A. S. M. Irwin, "Bitcoin transactions: a digital discovery of illicit activity on the blockchain," *J. Financ. Crime*, vol. 25, no. 1, pp. 109–130, Jan. 2018, doi: 10.1108/JFC-12-2016-0078.

[14] Y. Wu, A. Luo, and D. Xu, "Forensic Analysis of Bitcoin Transactions," in *2019 IEEE International Conference on Intelligence and Security Informatics (ISI)*, Shenzhen, China, Jul. 2019, pp. 167–169. doi: 10.1109/ISI.2019.8823498.

[15] A. Pinna, R. Tonelli, M. Orrú, and M. Marchesi, "A Petri Nets Model for Blockchain Analysis," *ArXiv170907790 Cs*, Sep. 2017, Accessed: Apr. 10, 2021. [Online]. Available: http://arxiv.org/abs/1709.07790

[16] A. Lr and D. Ao, “Bitcoin Investigations: Evolving Methodologies and Case Studies,” *J. Forensic Res.*, vol. 09, no. 03, 2018, doi: 10.4172/2157-7145.1000420.
[17] D. A. Orr and D. M. Lancaster, “Cryptocurrency and the Blockchain: A Discussion of Forensic Needs,” *Int. J. Cyber-Secur. Digit. Forensics*, vol. 7, no. 4, p. 420+, Oct. 2018.
[18] L. Van Der Horst, K.-K. R. Choo, and N.-A. Le-Khac, “Process Memory Investigation of the Bitcoin Clients Electrum and Bitcoin Core,” *IEEE Access*, vol. 5, pp. 22385–22398, 2017, doi: 10.1109/ACCESS.2017.2759766.
[19] “privacy.” https://dictionary.cambridge.org/dictionary/english/privacy (accessed Apr. 10, 2021).
[20] “A guide to GDPR data privacy requirements,” *GDPR.eu*, Feb. 22, 2019. https://gdpr.eu/data-privacy/ (accessed Apr. 10, 2021).
[21] K. A. Wallace, “Anonimity,” *Ethics Inf. Technol.*, vol. 1, no. 1, pp. 21–31, 1999, doi: 10.1023/A:1010066509278.
[22] “How It Works,” *Zcash*. https://z.cash/technology/ (accessed Apr. 10, 2021).
[23] “What is Digital Forensics | Phases of Digital Forensics,” *EC-Council*. https://www.eccouncil.org/what-is-digital-forensics/ (accessed Apr. 10, 2021).
[24] K. Kent, S. Chevalier, T. Grance, and H. Dang, “Guide to integrating forensic techniques into incident response,” National Institute of Standards and Technology, Gaithersburg, MD, NIST SP 800-86, 2006. doi: 10.6028/NIST.SP.800-86.
[25] U. Feige, A. Fiat, and A. Shamir, “Zero-knowledge proofs of identity,” *J. Cryptol.*, vol. 1, no. 2, pp. 77–94, Jun. 1988, doi: 10.1007/BF02351717.
[26] “What are zk-SNARKs?,” *Zcash*. https://z.cash/technology/zksnarks/ (accessed Apr. 10, 2021).
[27] “The Encrypted Memo Field,” *Electric Coin Company*, Dec. 05, 2016. https://electriccoin.co/ja/blog/encrypted-memo-field/ (accessed Apr. 15, 2021).
[28] “Monero vs zcash vs dash: which is the most anonymous?,” *Comparitech*, Apr. 04, 2018. https://www.comparitech.com/crypto/anonymous-cryptocurrency-monerto-zcash/ (accessed Apr. 10, 2021).
[29] D. Hamilton, “Investing in Dash Crypto – Everything You Need to Know,” *Securities.io*, Aug. 19, 2020. https://www.securities.io/investing-in-dash-everything-you-need-to-know/ (accessed Apr. 10, 2021).
[30] B. Academy, “What is PrivateSend? Anonymous payments with DASH,” *Bit2Me Academy*, Jun. 21, 2019. https://academy.bit2me.com/en/what-is-privatesend-dash/ (accessed Apr. 10, 2021).
[31] “Cryptocurrency Wallet Guide: A Step-By-Step Tutorial,” *Blockgeeks*, Feb. 27, 2017. https://blockgeeks.com/guides/cryptocurrency-wallet-guide/ (accessed Apr. 14, 2021).
[32] “Getting Started With Bitcoin - A simple guide for beginners.” https://learnmeabitcoin.com/beginners/getting-started (accessed Apr. 14, 2021).
[33] S. Jokić, “Analysis and security of crypto currency wallets,” *Zb. Rad. Univ. SINERGIJA*, vol. 19, no. 4, May 2019, doi: 10.7251/ZRSNG1801102J.
[34] “HD Wallets.” https://learnmeabitcoin.com/technical/hd-wallets (accessed Apr. 14, 2021).
[35] A. Biryukov and S. Tikhomirov, “Security and privacy of mobile wallet users in Bitcoin, Dash, Monero, and Zcash,” *Pervasive Mob. Comput.*, vol. 59, p. 101030, Oct. 2019, doi: 10.1016/j.pmcj.2019.101030.
[36] W. Koerhuis, T. Kechadi, and N.-A. Le-Khac, “Forensic analysis of privacy-oriented cryptocurrencies,” *Forensic Sci. Int. Digit. Investig.*, vol. 33, p. 200891, Jun. 2020, doi: 10.1016/j.fsidi.2019.200891.

[37] T. Volety, S. Saini, T. McGhin, C. Z. Liu, and K.-K. R. Choo, "Cracking Bitcoin wallets: I want what you have in the wallets," *Future Gener. Comput. Syst.*, vol. 91, pp. 136–143, Feb. 2019, doi: 10.1016/j.future.2018.08.029.
[38] Reith, Mark, Carr, Clint, and Gunsch, Gregg, "An Examination of Digital Forensic Models," vol. 1, no. 3, p. 12, 2002.
[39] S. Rahayu, Y. Robiah, and S. Sahib, "Mapping Process of Digital Forensic Investigation Framework," vol. 8, Jan. 2008.
[40] R. Tabuyo-Benito, H. Bahsi, and P. Peris-Lopez, "Forensics Analysis of an On-line Game over Steam Platform," in *Digital Forensics and Cyber Crime*, vol. 259, F. Breitinger and I. Baggili, Eds. Cham: Springer International Publishing, 2019, pp. 106–127. doi: 10.1007/978-3-030-05487-8_6.
[41] T. Y. Yang, A. Dehghantanha, K.-K. R. Choo, and Z. Muda, "Windows Instant Messaging App Forensics: Facebook and Skype as Case Studies," *PLOS ONE*, vol. 11, no. 3, p. e0150300, Mar. 2016, doi: 10.1371/journal.pone.0150300.
[42] R. McKemmish and Australian Institute of Criminology, "What is forensic computing?" Australian Institute of Criminology, Canberra, 1999.
[43] "Wallet Backup Instructions — Zcash Documentation 4.3.0 documentation." https://zcash.readthedocs.io/en/latest/rtd_pages/wallet_backup.html#using-z-exportwallet-z-importwallet (accessed Apr. 15, 2021).
[44] "Advanced topics — Dash latest documentation." https://docs.dash.org/en/stable/wallets/dashcore/advanced.html (accessed Apr. 15, 2021).
[45] "What is Tor? A beginner's guide to the privacy tool," *the Guardian*, Nov. 05, 2013. http://www.theguardian.com/technology/2013/nov/05/tor-beginners-guide-nsa-browser (accessed Apr. 15, 2021).
[46] "Zcash.conf Guide — Zcash Documentation 4.3.0 documentation." https://zcash.readthedocs.io/en/latest/rtd_pages/zcash_conf_guide.html (accessed Apr. 15, 2021).
[47] "New Release: 4.1.0," *Electric Coin Company*, Nov. 10, 2020. https://electriccoin.co/blog/new-release-4-1-0/ (accessed Apr. 15, 2021).
[48] G. Tankersley, "Foundation DNS Seeders Are Live," *The Zcash Foundation*. https://www.zfnd.org/blog/foundation-dns-seeder/ (accessed Apr. 15, 2021).

Appendix

A. Regular expressions and Keywords

Table 9. List of regular expressions and keywords used in Zcash and Dash

	Regex/Keyword	Used To	Zcash	DASH
Regex	((t1)([a-zA-z\d]{33}))$	Search transparent addresses	•	
	((zs)([a-zA-Z\d]{76}))$	Search shielded addresses	•	
	(^X[a-zA-Z0-9]{33})$	Search addresses		•
Keywords	AddToWallet	Show transaction ID	•	
	txid/Txid	Show transaction ID	•	
	secret-extended-key	Locate the private key of the shielded address	•	
	z_sendmany	Shows incoming transactions in combination with txid	•	
	advertising	Locate the IP address used by the wallet in debug file	•	
	purpose	Locate the transparent address in the wallet file	•	
	Added to wallet	Locate the transaction ID	•	
	AddressBook.json	Show the contact list in MFT	•	
	label	Show the contact list	•	
	zcash.conf	Show the configuration file in MFT	•	
	AddToWallet	Locate the transaction ID		•
	Address	Locate the sending address of a transaction		•
	Amount	Shows the amount of the transaction		•
	wallet (Fig. 60)	Show the path of the backup file		•
	hdchain	Shows the mnemonic phrase in D. Core		•

	receives	Locate the local addresses in the wallet file		•
	send	Locate the external addresses in the wallet file		•
	labels	Show information of transactions		•
	payment_requests	Show information of transactions		•
	keystore	Show the xprv, xpub		•
	change	Show the list of change address		•
	Receiving	Show the list of receiving address		•
	backup_dir	Show the path of the backup file		•
	qt-console-history	Show executed commands		•

B. File Hashes

Table 10. File hashes.

Filename	SHA256
Zecwallet Fullnode	
Case 1	
01_02082021.mem	155DDABDD7A7F2FF9D3689488542C96631ADA807751BAA5E0585DEFCB83FF4E5
01_02082021.raw	6A90462C8BD5908030B76BBB2D64723F49B631F72D1A14847E159998D0DC667B
Case 2	
02_12022021.mem	86FA0613D8CA9AE0E6FC11ECD31DB93E72170487173B413EB466C989B62D5547
enet_01_12022021.pcapng	AA08B51EA32F55EBD175641251702E4CFAF41B9BC494A7B2CD0AB41765A7D5CE
02_12022021.raw	A12D1BF4AF4A676387D84DB4E6CAF1703A7F852ACD0120EA992505F868D7B4E4
Case 3	
03_12022021.mem	2D2C71D086092B1CE424242969D305C91EDEECA2954DBB951C8B57C03F2A3667
enet_02_12022021.pcapng	3029228D88599FC20E4A93FAB19886D2E1D9569D47156ED91C5B68B174497370
03_12022021.raw	7D96435766B90F855FF334E272D9D2D186CD7CFD600041469B68EF24474DC415
Case 4	
04_13022021.mem	2DFD2CB6E6514C7DE6C1FF505395F8E4A05CF65BEDBA9E1FA634AFCE7AF00AE6
enet_01_13022021.pcapng	7208ED348C1E81B02CA1A5F2113EE571CA04A2B03FBD73E3820E0FB2B777C959

04_13022021.raw	786536915624D3013FD2B82AB909B8CA0EEB5C481941D2DE881B94E74440817C
Case 5	
05_14022021.mem	B5A830D2478B8BB8A56F84344EE9361366B4894A8D2DABA2DF218612D457706A
enet_01_14022021.pcapng	040FC61086E8D59FE76E500FF5B29B60F47B2B0EDCB9F0F9E7030E25CE80BAAA
05_14022021.raw	33F5B6FB711592898F8D60004E0AD9582C2EE3B7B457E3084BF04A532FE6A065
Case 6	
06_14022021.mem	17B0C3100D76F20360EB3CA7B93A86D938078CF0755E1FEEB9A12B50F0EA4C20
enet_02_14022021.pcapng	351E58FF095FDA9053B3DA03C36AE14D3E511CB46959E25D78666AF6FC2E59C8
06_14022021.raw	AC181A8754CDC1BDAA1F6DF348E0C6D3D1839D6EF8C483C4D73CBE4F961B6162
Case 7	
07_14022021.mem	8F354FAFAD458E50171F3DE55BD49449DB7660791A56FBBE4FA8DDA4E4BC7B41
enet_03_14022021.pcapng	F8FB27F1E688FE09CB5983F4EA86C5F008B7220A73F6EA267AA3F55E991E40BD
07_14022021.raw	6EBE83D05B127FB6329EAFC495C1232A7BAB1D31DD6731A55EFB4DC256F6CA9D
Case 8	
08_03032021.mem	78E93537DDFF927D6A559EB073F4B3D5CB710454D8BE900CE8F783D11A2DA38D
enet_01_03032021.pcapng	6F874602C24F3F4E5E7DA6C5A41B78FFC38275D1D934B816D856022A79530F1E
08_03032021.raw	59B36F1588F3082CAD14FE4BFDE1791ECFE36B9D02174B4C1676DD848E00D4AD
Zecwallet Lite	
Case 1	
enet_01_0311.pcapng	45776FDC83F789809D37BA47C256FBBD4CCC21D4C07C2585F51404AA256F13F8
01_0311.raw	6CA07528EDD60E2E156F7CDAED49C3920A38C48A06953659FEE09287DE70C34B
01_0311.mem	5FEB4DB2FAE671560E7735163A87BDF4A8423F06B4FBC84A87348350374D3DED
Case 2	
enet_02_0311.pcapng	CC58C285D13A694368EA98DB8D8AAFA2EFB80883B7259E1DEAFFD36AC721E6CE
02_0311.raw	8997F48646B48D3E619D76C74ABB4EA892F3496A9665816E15F298F380CCBFF4
02_0311.mem	09A262F11D34754878E93EE8A78DB5132D8DE13144BA4595D436BB88F859511B
Case 3	
enet_03_0311.pcapng	AD6DEC83D673E96C9480397194BD7E5533267906D5AF5D5BD37C791B3FAF13B5
03_0311.raw	78EFFBFA5B89B0B64B197B5FE9CE52874C34DA80FE18B07C59DF42F2218504E3
03_0311.mem	39A415CFB540E2DE0F234F0732408A80074B9994C065C5C19D86FD37D435E46B
Case 4	
enet_04_0311.pcapng	EC6F1962152A126BDE0AB6453DB2771E9EC38DC994D6C2AB24DFEC518F859332
04_0311.raw	EAE8C1000FC29D4FE03AF1ED9964AC55989291ACA88E88BEF8111163243708FD
04_0311.mem	C89BA114474AE3CBFE7F9776015ABC15236B6F20228A2D244E5F0364B5BF1BFA
Case 5	

enet_05_0311.pcapng	3CEDAEFC578695BA51A22F570F0727C247341C95B28E2853CF7D7511CB2E5663
05_0311.raw	3741DD2B149798D87D3E66E22BC2324386AF8DA18BBE9B002C512672581E2522
05_0311.mem	ED26C39C0773D3A9259A8906A63ECE82924800D26A0DC8E42D38FA18254A81A3
Dash Core	
Case 1	
01_03202021.raw	CCB777F7FE720688DBFA5CFDB582A8A199547BF4C0ECC13D32AF938D4493F0AF
01_03202021.mem	67FF5070EDEE92FFC53EADA740B56A361EDDE6CEF47251E844EC41E8FE26A79E
Case 2	
02_03202021.raw	3565C11FC457D768E6CC4025572FE53688B7C4C2346E17A99DC91B985B69F1A6
enet_02_03202021.pcapng	1F3BE60B51DA4755D2DDFB1FCD50974914B780C0E61F9E47CC4C4449E442EB01
02_03202021.mem	01B85A2D7F3C9E7A0A103AC9F820922BFA12E7FE9D180FA10F2985295B039651
Case 3	
03_03212021.raw	53F3C724489A60CC8899A2DEF0C8713582CABF64659B126F8E01CCAF45CED420
enet_03_03212021.pcapng	CAE039D29553C073F8F05BEF96BCB9D669851EBBEE9DC4DC55677F9FB54038B7
03_03212021.mem	A139AF2CF0E42238BD4AA1BD496658715902F8DE2359A8F9BAC942DCEC510548
Case 4	
04_03212021.raw	6DEE3EA5EF83A22EBB22FF838BA55DDF450B5B1F3B1F85F692516CAD5B9C43D2
enet_04_03212021.pcapng	868D6C66D16478D2287478CA72388F22E65BD78ADFE7BC390DC33A046D028A05
04_03212021.mem	5597CB602D5847AA709C091263B3AF69E30CF61B7EF4250835F50628291C8206
Case 5	
05_03242021.raw	78C3E59AE824E252F9C1CFA0052BBBCE821A3FB161168E32E1A94E77A1FD72D7
enet_05_03242021.pcapng	E86A9B1400407D6DE619D55A9E386223E8281728FA8480B7D4FDBC7A123A8017
05_03242021.mem	348560843D34606B3D5EFA23BE2980FE2711C812101C4C1E10B08D928517CC57
Case 6	
06_03242021.raw	A699A121D5F5418692ED0936622E839D278F7DAB5CD28C7448B0929A5305F535
enet_06_03242021.pcapng	4356A5A0EC4D316482C97847BCA2B9787244D77B5DE0DD8C002E011A5701BD47
06_03242021.mem	7B6FA82A51556980CF6D5F78C62C46BB4975A374BF71DC6939CD1EDE813B46AE
Dash Electrum	
Case 1	
enet_01_03242021.pcapng	8AFE4FA50C7728B0F423385CCC5A84079752DB49FC1248105AA6CED5B1B6F1AA
01_03242021.raw	6FE6E5F4AB27A95FE8A10499E6CE33FD61C2BF7958ABB5D8E43B6735C7792A92
01_03242021.mem	20F9C5F04262946C54190BBB0FF9732D63B925860EA5C88660870CFDB026F783
Case 2	
enet_02_03242021.pcapng	DE5B4395DD4C05E0F42FE6F8C2A5AFC4260450E2F5431A338F4C78E5ADF33073
02_03242021.raw	EBAFE40E688FFE72FB27C0D72C0334DFACCCA9FDEFEDC1B28AB765A272A4D428

02_03242021.mem	AAD950F4F58B1012B5A8AEDE89AE92CE47139815A3F1C968DDEE2637D584B835
Case 3	
enet_03_03242021.pcapng	EA721ECF8BBB029651F58BA13CD8AD2D2B82C119B4CA29BFA341D476A2130199
03_03252021.raw	C03623A3A19531C921B58A1DD9915432E36DAE3984219030C11DB0BD5AE9A29C
03_03252021.mem	575A73984C70F4FF35C36F160227E781B660B7D7AA16760D3846C0DA93A60023
Case 4	
enet_04_03252021.pcapng	FBCBD7319829B3D4FB700318574F2AFE6DD75ABB49F630A683488F73B1910355
04_03252021.raw	7A885B322DB9809EB5A12FBE4EC136669845C97D1A21315485D7A94CA6FEF654
04_03252021.mem	5BEBE0F03A8FD43BA9A916909991C3847A91CDC90A2E3425FD0692640DD8530C
Case 5	
enet_05_03252021.pcapng	62734C9903CB3836FF0FB8985184BDAFAFBB5EF466D8C009F398E1E0EF652F61
05_03252021.raw	DBBD6966173C17EB2265A22C786C8EA2AE3443490A54A32BC41134BFCB2A441A
05_03252021.mem	7C3B7CE4E797180B2142D39E1F823EEB432FF59F2C3AA1B3E4D343DCADDC1A6F
Case 6	
enet_06_03252021.pcapng	enet_06_03252021.pcapng
06_03252021.raw	CB2C52B64A840C85BA2403C816BD52F912A445F4C479C96856B74191087BEC0B
06_03252021.mem	D0C7BD990CFFC933D39E48934CCCCD96D1B584D12FE4FFD7D6011F32B4FE7448

C. Zecwallet Fullnode Transactions

Table 11. Zecwallet Fullnode transactions

Case	Time (UNIX format)	Date	Transaction ID (txID)	Direction	Amount (ZEC)	Recipient Address	Memo
Case 7	1613327516	Feb 14 2021 08::31 pm	441479f39c59ec4e171bd6f952d238fc60d341670a46ad607f3438d27400c4a7	send	0.00069	t1dv9Gzg8tWphFLuTdwBrSipkjbduVospqa	
Case 6	1613322189	Feb 14 2021 07::03 pm	f011ca4db4810b61c4e5beee53bf4d2938f486a7cc84639a94525f6c7edef107	send	0.0002	zs13tem6fljqf5kskn0kvgeqcxrhat7tj37w9l5w5vt0mmnuamsxchqlqptqrvvhz97g5zxg6670mu	'from T vm to Z iphone'
Case5	1613319958	Sun 14 Feb 2021 16:25:58	b48591f1cabd46509a66b937fe0b7905085da5a882cb343f863604d8464c28bf	send	0.00007	t1dv9Gzg8tWphFLuTdwBrSipkjbduVospqa	
Case 4	1613255265	Sat 13 Feb 2021 22:27:45	43baa44e9f1335a15e5c5412584b2e001def	send	0.0006	zs13tem6fljqf5kskn0kvgeqcxrha	'from Z vm to z iphone'

			74d94a76ddcc30b22fee15f79289			t7tj37w9l5w5vt0mmnuamsxchqlqptqrvvhz97g5zxg6670mu	
Case 3	1613161378	Feb 12 2021 10::22 pm	25bc98a33f1c33d81ed3bed427aeecb211605cb95ef36288f64a6bf538efeb35	receive	0.001	t1gxPPoGQuy6PT5QJFdC8wEjP7hUETG3Yrw	
Case 2	1613149966	Feb 12 2021 07::12 pm	25ee0e307e63efb06f07c0574de8dabddb245fcbd6e252eaa4709746da31de32	receive	0.00000001	zs1e4jvjsaft625y28jtcm9vyehak7u0jzlyqsr0y43y308y8ntdvvev37g7maq37seyljkxtsflfu	'From Z i to Z vm. JM'

D. Zecwallet Lite Transactions

Table 12. Zecwallet Lite transactions.

Case	Time (UNIX format)	Date	Transaction ID (txID)	Direction	Amount (ZEC)	Recipient Address	Memo
Case 5	1615483861	Mar 11 2021 07::31 pm	472dfe803c95ca5f2efea17b5797365717b1629a66859499fb16bf3d96624e5a	sent	0.344755	t1dv9Gzg8tWphFLuTdwBrSipkjbduVospqa	---
Case 4	1615480992	Mar 11 2021 06::43 pm	fb9d975ad5a2a09ebff448a47318c4b8a04e59be761a551a1d8ac904a27232aa	sent	0.344755	zs13tem6fljqf5kskn0kvgeqcxrhat7tj37w9l5w5vt0mmnuamsxchqlqptqrvvhz97g5zxg6670mu	'From "Z" vmlite to Z iphone Reply-To: zs1zr0v...'
Case 3	1615476458	Mar 11 2021 05::27 pm	87a64652f0046e31247ec33c590a05a108440b329bec8b278761fa06d5d09642	receive	0.09	t1QbX4ec2KBjAhyN1QM1gqqHGtF7P66iz6h	---
Case 3	1615476006	Mar 11 2021 05::20 pm	b7694872d104f5b9f57d9fad6ced02f278969d5c18865cc69d50d843516b2cca	receive	0.0999	zs1zr0v2y48jqazu3rhjdnv4msrx6wrfsk8xumnzyqxpt5fhu9d4n3r8y5wdwsnu9fw5784g2n4jrt	‘
Case 2	1615470390	Mar 11 2021 03::46 pm	066e1bd24b796e76be202ab99ffa87688	receive	0.24991	zs1zr0v2y48jqazu3rhjdnv4msrx6wrfsk	'From Z vmfullnode to Z vmlite. Case2. Reply-To:

			bdf032e281c97d58a2fa2a1fb71e584			8xumnzyqxpt5fhu9d4n3r8y5wdwsnu9fw5784g2n4jrt	zs1mycjdvvrlseegn7jtlz95p7g09j0y972fh3l8vl23czgm0ye9hrzy6l4l98ru8ez7745wqwunpm'
Case 2	1615469860	Mar 11 2021 03::37 pm	2850f2152523bdff6f48d7ab475718785e56c947cd2967b1b5f8d3cb7ec072aa	receive	0.2499	t1QbX4ec2KBjAhyN1QM1gqqHGtF7P66iz6h	---

E. Dash Core Transactions

Table 13. Dash Core transactions.

Case	Date	Type	Label	Adrress	Amount (DASH)	Direction	Transaction ID
Case 4	2021-03-21T10:45:47	PrivateSend	iPhone addr. PS	XogciEjYTBsczMER4dub1wqVf745GhZp25	-0.20700207	sent	0fb2f2f0aa1a925840f7af278a536db0ab800f921cd209adb857bbcf787b038a
Case 3	2021-03-21T00:29:10	Sent to	iPhone addr	Xr2D3wLMyThxHLtoQFxBK71h1B7Ptr9Wtn	-0.32	sent	e84c10b95087eaacd4f6bb21dadaa3ee410790c5c107e7bb6d973c9103ab55b3
Case 2	2021-03-20T22:50:07	Received with	---	XtaXbvRWspeVDE1YPA4z93Fa2JvubBdS4J	0.646905	receive	d1b97eff84da15e1b10d95f2bdbf23feffb0e2af18e2465959c1b90dc58b25d7

F. Dash Electrum Transactions

Table 14. Dash Electrum transactions.

Case	Date	Type	Label	Fee	Amount (DASH)	Direction	Transaction ID
Case 5	3/25/2021 01:53:32	PrivateSend	Case 5. from vmlite to iphone. Private Send	0.00004484	-0.11100111	Sent	923be98575fd49b07ee0da0393eaa2e85e341675c27ff35a410e22474f415cbd
Case 4	3/25/2021 00:28:10	InstantSend	Case 4. from vmlite to iPhone	0.00000339	-0.10000339	sent	2839983d0e43a1ef6b4e2d37baecaaed6f542bde334414ccb4b1cf20f10514d1

Case 3	3/24/2021 15:15:48	PrivateSend	Case3. from VMfull to VM lite		0.04999266	receive	18f9302c6ef900eaf69b40d7fceba495cc9cb971b25769dba4c1f3051b5f6f21
Case 2	3/24/2021 13:42:29	InstantSend	From VMfull to VMlite. Cas1		0.14999774	receive	b13ba4f5e4be8093f052dc679c86027d737706f0c5bd5798e504d7ba1f813cb5