Forensic Science International: Digital Investigation xxx (xxxx) xxx

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi

Forensic analysis of privacy-oriented cryptocurrencies

Wiebe Koerhuis [a], Tahar Kechadi [b], Nhien-An Le-Khac [b, *]

[a] National High Tech Crime Unit, Driebergen, the Netherlands
[b] University College Dublin, Belfield, Dublin 4, Ireland

ARTICLE INFO

Article history:
Received 3 June 2019
Accepted 31 October 2019
Available online xxx

Keywords:
Cryptocurrency forensics
Privacy-oriented cryptocurrency
Monero
Verge
Forensic artefacts

ABSTRACT

The privacy-oriented cryptocurrencies have built-in anonymity and privacy features that made them very difficult (nearly impossible) to trace funds back to a particular user or successfully seize funds present in a cryptocurrency wallet. Criminals use these currencies in different kinds of malware and DDOS extortion attacks to launder money. While academic research on Bitcoin is becoming more mainstream, the research on privacy-oriented cryptocurrencies is not very common. In this paper, we address the privacy-oriented cryptocurrencies Monero and Verge and investigate which valuable forensic artefacts the software of these cryptocurrencies leaves behind on a computer system. We examine different sources of potential evidence like the volatile memory, network traffic and hard disks of the system running the cryptocurrency software. In almost all sources of evidence there are valuable forensic artefacts. These artefacts vary from mnemonic seed phrases and plain text passphrases in the volatile memory to indicators of the use of a cryptocurrency in the captured network traffic.

1. Introduction

Cryptocurrencies are highly popular nowadays and the amount of money involved went to an all-time high of $826 Billion on January 2018 and 24h volume on May 2019 almost reached $73 Billion according to coinmarketcap.com (Coinmarketcap charts). These new trading practices attract criminals, as they can use cryptocurrencies to launder money, fund terrorism and buy illegal goods on Darknet markets (Europol, 2017). In the past few years when people were talking about cryptocurrencies, they almost always meant the Bitcoin currency. So far, Bitcoin is the most accepted cryptocurrency in the criminal world, and it is widely used in ransomware and DDOS extortion attacks (Europol, 2017). Recently, Europol stated that criminals are shifting to several alternatives like Monero, Zcash and Ethereum because of the privacy and anonymity options these cryptocurrencies may provide (Europol, 2017).

Monero, for example, is a privacy coin and its developers are working very hard to provide extra security, anonymity and privacy for their users. They use a non-public blockchain, untraceable addresses, and they obfuscate the transaction amount (Europol, 2017). This privacy-oriented cryptocurrency was created in April 2014. Monero is a fork[1] of the Bytecoin cryptocurrency and it is based on the Cryptonote protocol (Chan et al., 2017). The Cryptonote protocol includes several anonymity features, such as untraceable payments, unlinkable transactions and blockchain analysis resistance (Moseret al., 2018). Unlike Bitcoin, in Monero, it is very difficult or near impossible to link payments to a particular user and thus, it is very difficult for law enforcement to track and seize funds (Androulaki et al., 2013).

Verge cryptocurrency is another example. It was created in 2014 under the name DogeDarkCoin and was rebranded to Verge currency in 2016 (Verge Currency). Verge currency utilizes several anonymity features to provide privacy and anonymity for their users. Verge integrated TOR[1] and I2P anonymous network[2] technologies to obfuscate IP addresses (Verge Currency). In the beginning of 2018 Verge introduced the Wraith Protocol, another measure to provide anonymity for their users by using untraceable stealth addresses (CryptoRekt, 2017).

With a coin like Bitcoin all the transactions are publicly recorded in a blockchain and transactions and transaction amounts can be traced back to a Bitcoin public address e.g. a Bitcoin user (Androulaki et al., 2013) (Reid et al., 2013). With Monero and Verge, being private coins, the main problem for law enforcement is the

* Corresponding author.
E-mail address: an.lekhac@ucd.ie (N.-A. Le-Khac).

[1] https://www.torproject.org/.
[2] https://geti2p.net.

https://doi.org/10.1016/j.fsidi.2019.200891
1742-2876/

inability to trace payments and transactions of a particular user with publicly available blockchain information. Privacy-oriented coins are getting more attention from both developers and users than Bitcoin as it does not really provide anonymity (Europol, 2017) (Conti et al., 2018) (Meiklejohnet al., 2016) (Androulaki et al., 2013).

Most forensic analyses on cryptocurrency in the literature have been focusing on Bitcoin (Conti et al., 2018) (Meiklejohnet al., 2016) (Van der Horst et al., 2017) (Bonneau et al., 2015) (Haigh et al., 2019) (Zollner et al., 2019) (Doran, 2014). There is very little research concerning privacy-oriented cryptocurrencies. Hence, the objective of this paper is to study the forensic acquisition and analysis of artefacts that the privacy-oriented cryptocurrencies leave on a computer system.

In this paper we focus on two popular privacy coins: Monero and Verge. We analyse the forensic valuable artefacts that can be found on a hard disk and/or volatile memory of a seized computer and/or in the network traffic (wiretap) from a computer running the Monero or Verge cryptocurrency wallet application. We also discuss the forensic acquisitions and analysis of Monero and Verge privacy cryptocurrencies.

The remainder of the paper is organised as follows. We present the concept of privacy-oriented cryptocurrencies, as well as Monero and Verge with their valuable forensic artefacts in Section 2. Section 3 reviews the literature of privacy-oriented coins. Details of our approach is described in Section 4. Sections 5 and 6 describe the experiments and discuss the results, respectively. Finally, Section 7 concludes the paper and highlights some future directions.

2. Privacy-oriented cryptocurrencies

2.1. Anonymity and privacy

An important aspect of maintaining privacy and anonymity in cryptocurrency is preventing others from knowing the user public address. One should avoid publishing their public address together with other personal identifiable information (***PII***) if one wants to remain anonymous (99Bitcoins, 2019). This is a bit challenging because if one wants to get paid one has to give out their public address. A solution to this problem would be the use of so-called “one-time addresses”. A simple use of a one-time address is to manually generate a new address for every transaction (Conti et al., 2018). With stealth addressing this one-time address generation procedure is someway automated and the implementation of this mechanism varies a bit for cryptocurrency. A stealth address is not visible on the blockchain with a blockchain explorer and should only be known between the sender and the receiver (Moseret al., 2018) (CryptoRekt, 2017). The one-time address generated by the stealth address mechanism is recorded on the blockchain and it should not be linkable to other transactions or the stealth address by others (CryptoRekt, 2017) (Alonso, 2017).

With a blockchain explorer everybody is able to lookup what transactions a public address was part of (Conti et al., 2018). This is not the case with a Monero public address, the Monero public address is never recorded in the blockchain. Only one-time addresses are recorded in the blockchain and those addresses are not easily linkable or traceable (Moseret al., 2018). Another crucial aspect for privacy and anonymity is hiding a real IP-address. This can be achieved with anonymous network such as TOR. TOR encrypts network traffic and sends it across nodes around the world to hide the sender real IP address.

To summarise, there are three main aspects to obtain anonymity in the cryptocurrency world, which are:

- The use of an anonymity network like TOR or I2P.
- The use of one-time/stealth addressing.
- Never publish ***PII*** together with a public or stealth address.

2.2. Anonymous cryptocurrencies

There are several privacy-oriented and anonymous cryptocurrencies with different features at the moment. In this section, we investigate two popular cryptocurrencies. The first currency is Monero XMR (Monero, 2019). It is listed in the top 10 cryptocurrencies on CoinMarketCap (Coinmarketcap charts) and it is largely used by cybercriminals (Europol, 2017) (Reed, 2018) (Mursch, 2019) in different forms of crimes. The second cryptocurrency we investigated is Verge XVG. In the beginning of 2017, its value was around $0.000019 per coin. At the end of 2017, during the peak of the market, the value of a Verge coin skyrocketed to $0.30 per coin (Coinmarketcap charts); a gain of around 1,582,000%. According to (Williams, 2017), the main reasons of this giant increase in value is the growing interest in privacy coins and the fear of missing out. Another reason is evading paying tax on cryptocurrency gains (Williams, 2017). As May of 2019, its value is still at $0.0099.

2.2.1. Monero

Monero is a fork of the cryptocurrency Bytecoin and it is based on the Cryptonote protocol (Chan et al., 2017) (Monero, 2019). The Cryptonote protocol includes several anonymity features like untraceable payments, unlinkable transactions and blockchain analysis resistance (Moseret al., 2018). A complete overview of the Cryptonote protocol is already addressed in (Moseret al., 2018) (Alonso, 2017). It uses a ‘private’ blockchain and so-called stealth addressing and as a result, transactions cannot be linked to a particular user (Monero, 2019). Unlike Bitcoin, with Monero you are not able to query someone's public address on the blockchain to lookup transactions and balance because public addresses are not recorded in the blockchain (Alonso, 2017).

When a Monero wallet is created, four different keys and a public address are generated. Monero uses a private and public view keys, a private and public spend keys and just one public address for receiving payments (Alonso, 2017). In addition, for each transaction a one-time address is generated and only those addresses are recorded in the blockchain (Alonso, 2017) (Monero, 2019).

In (Alonso, 2017) the authors described how the Monero software (version 0.11.1.0) works and how the underlying cryptographic algorithms provide anonymity and privacy to the Monero users. One conclusion is that the ring size[3] is an important factor for traceability, bigger ring sizes (a.k.a. *mixins*) make it more difficult to trace transactions (Alonso, 2017). In the latest release (Monero version 0.12.0.0), the minimum ring size was increased from 4 to 7 to provide a better privacy (Monero, 2019). Authors in (Moseret al., 2018) also addressed the traceability of transactions on the Monero blockchain. They stated that they can trace the spending of coins by deduction and most of the time the newest input is the real input of a transaction (Moseret al., 2018).

The studies reported in (Moseret al., 2018) (Alonso, 2017) (Monero, 2019) do not explore forensic artefacts that Monero software can leave on computer systems and their main focus is on blockchain analysis and the cryptographic fundaments of Monero.

2.2.1.1. Monero-valuable forensic artefacts. Monero currency wallet software contains interesting forensic artefacts that can be useful for law enforcement. The value of these artefacts varies from user

[3] https://getmonero.org/resources/moneropedia/ring-size.html.

Please cite this article as: Koerhuis, W et al., Forensic analysis of privacy-oriented cryptocurrencies, Forensic Science International: Digital Investigation, https://doi.org/10.1016/j.fsidi.2019.200891

attribution to full access to a user's wallet.

As stated earlier, when creating a Monero wallet, the software creates four different keys and one public address. The public address contains 95 characters and it always starts with the number '4'. A Monero public address looks like this: *496ZzMcZk8U2bcaLGo YSsSiDC3iNisMvibPYVXt34EbfDWozXpC7R28HbyxioHNfV7Q63yDLgYxugCzAC43cGFB1jErn84*. A Monero wallet has only one public address and to receive funds the user must communicate their address to the sender (Monero, 2019) (OpenOne Labs and "F are al, 2019). A public address can identify a wallet present on a suspect's computer system and therefore its forensic value is significant.

There is also another kind of public address called an *integrated* address, which contains a payment id set by the receiver to distinguish different payments. An *integrated* address is 106 characters long and also starts with the number '4' and looks like this: *4JoF1AS4MPz**2bcaLGoYSsSiDC3iNisMvibPYVXt34EbfDWozXpC7R28THbyxioHNfV7Q63yDLgYxugCzAC43cGFB**27oNtufZkrTEcV7mUo*. The 77 characters (in bold) of the integrated address correspond to the public address of the wallet and therefore the integrated address can be traced back to a public address (OpenOne Labs and "F are al, 2019). An integrated address and a payment id can be of great forensic value because they can identify a public address and thus a wallet. For example, with a payment id you can look up a transaction on the blockchain and find a corresponding transaction id.

Monero uses transaction ids to record transactions on the blockchain. With a transaction id we can see what one-time addresses are involved in a transaction (Monero, 2019). A typical transaction id looks like this: *4485151e06b936e56ce7f5f132c1026608bca716c23bfa4e4ad88a6155a88aa6*. It consists of 64 hexadecimal characters. The value of this artefact lies in the facts that a transaction id can be used to look up transactions on the blockchain. However, due to the nature of the Monero blockchain it does not reveal a lot of useful information.

The *private spend key* consists of 64 hexadecimal characters and it is used for spending Monero and it is also used for viewing older transactions that the owner had spent (OpenOne Labs and "F are al, 2019).

The forensic value is substantial because with a private spend key an investigator can recover a wallet without knowing the passphrase or seed phrase. A private spend key looks like this: *d17f7ee37fc904cd04692b0db2a8cc003008de7975d7b0ed7c1212b9892cca03*.

The *private view key* can be used to view all the transactions the owner received from other addresses. A private view key is made of 64 hexadecimal characters and it is derived from the private spend key (OpenOne Labs and "F are al, 2019). A private view key looks like this: *33c71dd92b44bfb25d9adfa25e0b7efb4565edac457a2e0bc1cbc021484d5f05*. With this key, we can see all the incoming transactions to owner's wallet without having the private spend key. Therefore, a forensic investigator is able to see all the incoming transactions to a wallet.

The *public spend key* and *public view key* are used to create the public address of the wallet. So, in theory the owner can recreate the public address of a wallet if the owner is able to retrieve both public keys (OpenOne Labs and "F are al, 2019). Forensically, this artefact is not really important because if we are able to retrieve both public keys the chance that we come across the public address is more likely, as it is stored in plain text in several places.

When creating a wallet, the software creates a 25-word long sentence called a Mnemonic seed, which is used to create the private spend key (OpenOne Labs and "F are al, 2019). With the seed we can recover the wallet without knowing the passphrase and thus an investigator is able to get control over a wallet when the seed is known. This is really very important for a forensic investigator. A Mnemonic seed looks like this: *exquisite dumb athlete crazy costume roared dinner tether growing summon fazed dual also hijack circle apricot eavesdrop gills tawny uptight flower wolf eskimo frying crazy*. The last word of the seed is used as a checksum and is present earlier in the seed (Monero, 2019).

During the wallet creation process, it asks for a passphrase to encrypt and protect the wallet key file. However, a passphrase is not mandatory, and you can skip this step by leaving the passphrase fields blank. When opening the wallet or sending Monero to another address you need to enter the passphrase to gain access to the wallet and be able to send Monero. With the passphrase an investigator can gain full access to the funds inside and this is very interesting to the forensic investigator.

The Monero wallet consists of three files, a wallet file, a key file and a public-address text file. With the key file and the passphrase, we can recover a wallet. The public-address text file only contains the public address of the wallet and can be used to link a public address to a wallet. The actual wallet file is encrypted and not readable. These three files do have forensic value because they can be used to identify and recover a wallet (only if the passphrase is known).

Other valuable forensic artefacts are indicators that reveal the presence of Monero software running on a computer system. An indicator of the use of Monero can, for example, capture Monero related DNS traffic present in a wiretap.

2.2.2. Verge

DogeDarkCoin currency was created in 2014 and was rebranded to Verge currency in 2016. Verge currency uses several anonymity features to provide privacy and anonymity for their users. Its developers integrated TOR and I2P in its wallet software design to obfuscate IP addresses of its users (Verge Currency). Obfuscating IP addresses is crucial for privacy according to (Bonneau et al., 2015). In early 2018, Verge introduced the Wraith Protocol; another approach for providing anonymity. With Wraith protocol it is possible to use a stealth address for receiving funds (CryptoRekt, 2017). A stealth address cannot be queried with a blockchain explorer and it is not recorded in the blockchain. Transactions made to 'normal' Verge public addresses are visible for the public and can be traced back to a public address on the Verge blockchain (Verge Currency) (CryptoRekt, 2017).

Verge currency uses stealth addressing which allows a user who wants to send funds to create a one-time address based on a provided stealth address by the receiver. Only the sender and receiver can see that a generated one-time address belongs to the receiver and only the receiver is able to recover and spend the money at that one-time address (CryptoRekt, 2017). The cryptographic foundation of the one-time address generation lies in the Elliptic Curve Diffie-Hellman algorithm (Hankerson et al., 2003), and is outside the scope of this paper.

With the introduction of the Wraith protocol Verge was able to integrate TOR in the core wallet software. TOR is enabled by default and there is no possibility to establish a connection outside of the TOR network. Verge wallet software also uses TOR for encrypting connection and all network traffic should be fully encrypted (CryptoRekt, 2017).

To the best of our knowledge there is no academic research conducted on the Verge core wallet software, Verge blockchain and on the artefacts this software may have. There are many discussions about Verge and other currencies on the news and technology websites and forums from people who have some kind of interest in Verge or other competing currencies.

2.2.2.1. Verge - valuable forensic artefacts. Verge currency wallet

Please cite this article as: Koerhuis, W et al., Forensic analysis of privacy-oriented cryptocurrencies, Forensic Science International: Digital Investigation, https://doi.org/10.1016/j.fsidi.2019.200891

software contains interesting forensic artefacts that can be useful for law enforcement. These artefacts vary from user attribution to full access to the wallet of a user. We conducted an investigation of this currency and deduced the following.

The Public address is like the wallets home address, if we know this address, we can send money to this address. A wallet can have multiple public addresses. The public address is recorded in the Verge blockchain when this address is part of a transaction (sender or receiver). Also, this address needs to be announced when a user wants to receive payments (Verge Currency). A Verge public address always starts with a capital 'D' and contains 34 characters. A Verge public address looks like this: "*DHtq9Yxw5XDZMAC Av8S1a1kFS6Xv ZQNeBM*".

When a Verge public address exists on the blockchain it is part of a transaction. A forensic investigator is, therefore, able to trace the flow of Verge currency on the blockchain and see the amount of Verge present on that address. With the Private address (a.k.a. private key) we have full control over the funds belonging to that address. A wallet can have multiple private addresses with corresponding public addresses. A Verge private address starts with a capital 'Q' and contains 52 characters. A Verge private address looks like this: "*QRtTSDRB2nAj6rFdLCzop kexL234qrsVqNpHAAyksFo9Lh TiuFbc*".

A Verge private address never leaves the wallet, it can only be obtained from the wallet itself. The forensic value of a private address is substantial because with the private address you have full control over the funds belonging to that address and therefore you are able to seize the coins present on that address. A Stealth address is not recorded publicly in the Verge blockchain. When two users perform a stealth transaction, the receiver needs to generate a stealth address within the wallet software and then communicates this address to the sender. The sender uses this stealth address as the recipient address for the transaction (CryptoRekt, 2017). A Verge Stealth Address starts with the characters 'smY' and has 102-character string that consists of a public view key and a public send key (CryptoRekt, 2017). A Verge stealth address looks like this: *smYo2Ey6E-*
dA4g-
bTJqQXd14XPfu7FDq6ATW9Y6vsY9yBH1HJChQ86vgocCjeZEttx-
gHYJzSDZQq2qjNsjcjFsUm6CXmfAd9NDkqM5Jp. The forensic value of a stealth address is less than a normal public address because a stealth address is not recorded publicly in the blockchain and cannot be used to lookup transactions on the blockchain. But a stealth address announced on for example a Darknet market can potentially identify a user.

A Wallet password or passphrase is used to protect the wallet of a user. During setup of the Verge wallet software there is no option to protect the wallet with a password. After setup a user can choose to protect the wallet with the "Encrypt wallet" option. This option implies that the wallet. dat file gets encrypted and protected with a password. A wallet password is a valuable forensic artefact because it gives full control over a wallet file and thus the ability to seize the funds present in that wallet.

A Transaction ID or Tx id is a unique identifier of a transaction on the blockchain. With a transaction id it is possible to lookup a transaction with a blockchain explorer (https://verge-blockchain.info/) and examine the details of that transaction like the sender, receiver and amount of Verge sent. Also, with a transaction id it is possible to identify stealth transactions within the wallet software and look up which public address was used to send a certain amount of XVG with. In short, transaction ids are unique identifiers of transactions which can identify a particular transaction made between users of the Verge currency and therefore an important forensic artefact.

An XVG transaction id is a 64 characters long hexadecimal string and it looks like this: *f4393787e70802b370235bcb7e6654b399a 6860eebf81cfa0efe5cf143a032d8*.

The Verge core wallet does not use a so-called seed phrase. The Verge Electrum Tor wallet does use a seed phrase, but this piece of software is outside the scope of this paper. The Electrum software is used by many different cryptocurrencies and is already researched by others (Van der Horst et al., 2017). The actual Verge wallet. dat wallet file is a valuable piece of evidence and contains all the private and public keys. With an unencrypted wallet. dat file an investigator has full control over a wallet and over the funds inside. Other valuable forensic artefacts are indicators that reveal the presence of Verge software running on a computer system, such as. Verge DNS traffic present in a wiretap.

3. Related work

Cryptocurrencies are a fairly new phenomenon and mass-adoption has yet to come. This has pushed academics to conduct research on privacy-oriented cryptocurrencies. There is significant research work on the number one cryptocurrency called Bitcoin like (Conti et al., 2018) (Meiklejohnet al., 2016) (Androulaki et al., 2013) (Reid et al., 2013) (Van der Horst et al., 2017) (Bonneau et al., 2015) (Zollner et al., 2019) (Neilson et al., 2016) but there is very few research works on cryptocurrencies that focus on privacy and anonymity like Monero and Verge. The papers (Moseret al., 2018) (Alonso, 2017) focused on the Monero privacy-oriented currency but they only discuss the cryptographic implementations on how to achieve the privacy and anonymity and not on forensic artefacts these coins may leave on computers. On news and technology websites we can find many non-academic articles about cryptocurrencies Monero and Verge.

There is some academic research about the Monero currency and how this currency works but we did not find any academic research about the Verge currency.

In (Alonso, 2017) the authors show that it is possible to conceal the sender and receiver of the transaction on the public blockchain without losing a way to verify the transaction for the community. It also addresses in depth how the Monero currency and software work. Another work about Monero entitled 'An Empirical Analysis of Traceability in the Monero Blockchain' is presented in (Moseret al., 2018). The authors addressed the traceability of transactions on the Monero blockchain. They stated that they can trace the spending of coins on the blockchain by deduction and most of the time the newest input is the real input of a transaction. Both studies focus on the traceability of coins on the blockchain and on the cryptographic fundaments of the currency.

To the best of our knowledge, there is no academic research about the Verge currency. The official Verge 'Blackpaper' (CryptoRekt, 2017) describes broadly how the currency works and what privacy measures are built-in, but this article is more a marketing piece than a research reference work.

There is academic writing on Bitcoin forensics like (Van der Horst et al., 2017) which focus is on the Windows platform and only addresses the Bitcoin clients Electrum and Bitcoin Core and it does not discuss other operating systems or cryptocurrencies. This research concludes that valuable forensic artefacts like private addresses, seed phrases and transaction ids are present on the analysed systems. The paper (Montanez, 2014) does address other coins than Bitcoin but this research main focus is on mobile operating systems and the wallets used on those mobile operating systems, and it also does not include the Monero or Verge currencies. In 'A forensic look at Bitcoin' (Zollner et al., 2019) the authors investigate several different Bitcoin wallets installed on a Windows 7 operating system. Their findings conclude that there are interesting forensic artefacts present on the system, but they

Please cite this article as: Koerhuis, W et al., Forensic analysis of privacy-oriented cryptocurrencies, Forensic Science International: Digital Investigation, https://doi.org/10.1016/j.fsidi.2019.200891

did not search for passwords or seed phrases or other useful artefacts. Also, this study did not include a Linux operating system or other coins than Bitcoin.

Most research works on cryptocurrency focus on the Bitcoin currency and the corresponding Bitcoin blockchain. Forensic research on cryptocurrencies is conducted on Windows or mobile operating systems like iOS and Android and it is not yet conducted on a Linux operating system as of our knowledge. This motivated us to conduct this study on privacy-oriented cryptocurrencies.

4. Approach

4.1. Challenges

The knowledge and experience of law enforcement agencies to successfully investigate, seize and trace virtual currencies continue to grow, but these capabilities are often limited to the Bitcoin currency and not to the other emerging currencies like Monero (Europol, 2017). For example, Europol encourages law enforcement agencies and private companies to continue to develop their knowledge about the emerging cryptocurrencies (Europol, 2017).

Cryptocurrencies are widely used by criminals and in March 2017 a malware discovered by a researcher from Avast probably was the first ransomware that uses Monero as a payment option (Abrams, 2018). Another cybercriminal use of Monero for example are forms of malware that secretly mine[4] Monero on your computer like they discovered in (Mursch, 2019) and (Reed, 2018). About 5% of all Monero coins in circulation are mined maliciously according to (Grunzweig, 2018). Cryptocurrencies are also used in money laundering, drug sales, sex trafficking, child exploitation and terrorism funding (Europol, 2017) (Orcutt, 2018).

Private companies, such as Chainalysis (https://www.chainalysis.com/) provide blockchain analytic tools that work perfectly for Bitcoin because Bitcoin has a public and open blockchain (Orcutt, 2018). But privacy-oriented cryptocurrencies with built-in privacy and anonymity make this a very complex or near impossible exercise (Moseret al., 2018) (Alonso, 2017).

4.2. Forensic process

According to (Casey, 2009) the basic forensic process consists of 4 stages. Collection (1) of evidence, the Examination (2) and Analysis (3) of the collected evidence to extract valuable forensic artefacts and the Reporting (4) on the findings from the Examination and Analysis stage.

We used these four steps to structure the experiment and extract forensic artefacts from different sources. These sources of evidence are the network traffic flowing to and from the virtual machine, the volatile memory of the virtual machine in different states and the virtual hard disk of the virtual machine. We will discuss these sources and the collection of these sources later on in this paper.

The workflow, as shown in Fig. 1, starts with the collection stage which consists of the identification of the evidence; in this case the virtual machine. In this first stage the different sources of evidence (volatile memory, network traffic and hard disk) are identified and acquired. In the next two stages these sources of evidence are examined and analysed.

During these stages the sources of evidence are examined and analysed with pre-known values as described earlier. In the fourth and last stage the outcome of the earlier stages like the valuable forensic artefacts are interpreted and reported in Section 6.

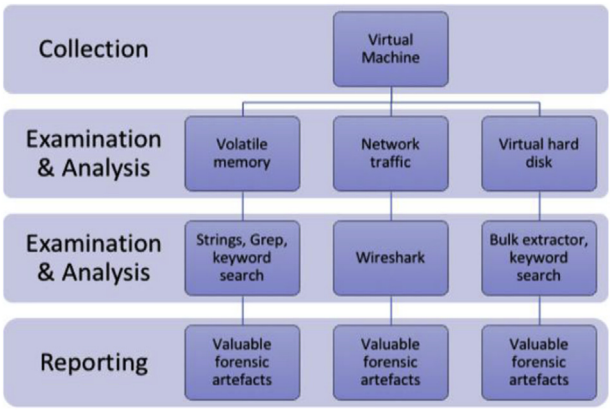

Fig. 1. Forensic workflow.

The forensic process utilised in this experiment includes three different disciplines from the digital forensics work field. Namely, network traffic forensics, volatile memory forensics and 'normal' hard disk forensics.

5. Experiments - setup

5.1. Environment setup

We use the virtual machine software VMWare Fusion version 10.1.1 on a MacBook Pro (Retina, Mid, 2012; Core i7, 16 GB) and running the operating system macOS High Sierra version 10.13.4. The guest OS, running as a virtual machine, is Ubuntu 16.04 LTS configured with 2 GB of RAM with all the software updates applied to date. For both the Monero and Verge experiments we use a separate virtual machine to avoid any contamination.

5.1.1. Monero setup

Monero GUI version 0.11.1.0 with *monerod* 0.12.0.0. The GUI software used in the Monero experiment is not the most recent software to date at that time. We did use the *monerod* daemon file from the newest release, version 0.12.0.0. We installed the Monero software on the fully up-to-date Ubuntu virtual machine and let the software locally synchronise the complete blockchain with the command,./monerod before creating a wallet. This command starts the Monero daemon and creates a copy of the complete blockchain in the directory:/home/{user}/.bitmonero/lmdb/. This directory contains two files: *data. mdb* and *lock. mdb*, the first file represents the blockchain and the second file is an 8K sized lock file without any forensic use. The Monero blockchain file is around 60 GB in size and grows constantly while updating the blockchain.

5.1.1.1. Wallet creation. After synchronisation of the blockchain, we started the Monero software with the./start-gui command. If there is not a wallet available in the default wallet location *(/home/{user}/Monero/wallets/)* the software shows a wizard to create a new wallet. For the experiment we created a wallet named M0n3r0Wall3t.wallet with the passphrase This-Is-The-Monero-Password, both the wallet name and the password are rather unique and should not give any false positives when searching for it. The wallet is stored in the*/home/{user}/Monero/wallets/{wallet-name}*/directory and contains three files. The wallet file, a key file and a plain text file with the Monero Public Address. During the wallet creation process the software generated a 25-word long phrase called the mnemonic seed and this seed can be used to

[4] Monero mining malware: malware that uses your computing power to create new Monero coins.

Please cite this article as: Koerhuis, W et al., Forensic analysis of privacy-oriented cryptocurrencies, Forensic Science International: Digital Investigation, https://doi.org/10.1016/j.fsidi.2019.200891

recover a lost wallet without the need for a password as discussed earlier. After the creation of a wallet the software is ready to use and the wallet can receive funds from another wallets address.

5.1.1.2. Address creation. With the Monero software we cannot generate new public addresses to receive funds. A wallet has only one public address for receiving funds. We can create a new integrated address for every transaction instead and this integrated address also includes a big part of your public address as described earlier.

5.1.2. Verge core set up

The software used in the Verge experiment is the most recent software and has version 4.0.2.0. We installed the Verge software on the fully up-to-date Ubuntu virtual machine and let the software locally synchronise the complete blockchain with the command, *./src/qt/VERGE-qt*. While invoking this command Verge automatically creates a wallet. dat file if no wallet file is present. Verge stores the blockchain files and the wallet file by default in the/*home/{user}/.VERGE*/directory. This directory contains several interesting files like several log files and the actual wallet file.

5.1.2.1. Wallet creation. Wallet creation with the Verge software is fairly easy compared to Monero. The only thing a user needs to do is run the *./src/qt/VERGE-qt* command and a wallet file is created automatically after the full blockchain is loaded. But this 'user-friendly' process also incurs a serious security flaw. A Verge wallet is neither encrypted nor password protected by default.

While this is good news for law enforcement it is not so good for the privacy of the user because everybody with physical access to the wallet file can spend all the funds present in that wallet. As described in the previous paragraph, the Verge wallet is located in the/*home/{user}/.VERGE*/directory and has the filename wallet. dat. In the Verge software there is an option to encrypt the wallet file. The password used for the encryption of the wallet is: This-Is-The-Verge-Password. In this experiment we examined the wallet file in encrypted state only because this is the most secure and probably common state of the wallet we can encounter. And with a non-encrypted password less wallet we already have full control over the funds inside, so there is no need to investigate an unprotected wallet.

5.1.2.2. Address creation. Verge software uses stealth addresses to 'hide' transactions on the blockchain. A Verge user can create a stealth address within the Verge software to receive a payment. For this experiment we created one stealth address and one normal public address with a corresponding label *Stealth_Address_#1* and *Normal_Address_#1*. The Verge software creates a public address when creating a wallet, this public address has not got a label.

5.2. Volatile memory image

The memory images are obtained by suspending the virtual machine and copying the . *vmem* file to a designated folder on the host system for examination. The virtual machine is powered off and rebooted after every memory creation action to clear up the memory. There is some differences in memory images for the Verge and Monero experiment because of the differences in wallet functions. For the Verge experiment we created 6 different memory images (Table 1).

For the Monero experiment we created 7 different memory images (Table 2).

After the creation of the memory images we created two different text files from the memory.

5.3. Network traffic capture

During transactions we captured all the network traffic to and from the virtual machine by using Wireshark. For the Verge experiment we captured the network traffic during sending and receiving of funds and continued capturing until the transaction had 6 confirmations. We also made a capture during wallet start-up to look for indicators that can identify the use of the Verge software.

For the Monero experiment we made traffic captures during sending and receiving of funds and waited until the transaction was fully confirmed (around 20 min). The Monero software also includes a resolve option, which functions like DNS and is called OpenAlias (https://openalias.org/), we also made a network traffic capture during a resolve action. To look for indicators of the use of Monero software we made a network traffic capture during wallet start up.

5.4. Disk image analysis

After the Monero and Verge experiment we closed the wallets and made a snapshot of the virtual machines to preserve the current state of the virtual hard disks for further investigation. We copied these virtual hard disks to a designated location for investigation.

6. Experiments and results

This section is divided in two main subsections in which we describe the results of the Monero experiment and the Verge experiment. In both subsections the results are categorised based on the resources: memory research, network capture research and disk analysis. Finally, we discuss the experimental results.

6.1. Monero experiment results

6.1.1. Memory images

For Monero, 8 images are created in different states of the virtual machine. After every memory creation action, the virtual machine is shut down and rebooted. The forensic artefacts found in each memory image are listed as follows.

Memory image 1 (*1_after_creation.vmem*):

This image is created after the creation of the wallet, the following artefacts are found in this image:

- Wallet passphrase (ASCII and UTF16 format)
- Mnemonic seed phrase (only stored in UTF16 format)

We did not find the private keys in memory, maybe they are present in memory in a different format.

Memory image 2 *(2_after_unlock.vmem)*:

After rebooting the virtual machine, we unlock the wallet with the passphrase and open the wallet, after a few seconds we then create this memory image, the following artefacts are found in this image:

- Wallet passphrase (ASCII and UTF16 format)
- Public address of own wallet (ASCII format)

Memory image 3 (*3_after_receiving.vmem*):

The third image was created after receiving one transaction from a different wallet. The following artefacts are found in this image:

- Wallet passphrase (ASCII and UTF16 format)

Please cite this article as: Koerhuis, W et al., Forensic analysis of privacy-oriented cryptocurrencies, Forensic Science International: Digital Investigation, https://doi.org/10.1016/j.fsidi.2019.200891

Table 1
Verge images.

Image	Wallet state	Wallet Locked/Unlocked
1._after_creation.vmem	Never used encrypted wallet	Locked
2._after_unlock.vmem	Never used encrypted wallet	Unlocked
3._after_address_creation.vmem	Wallet with custom created addresses	Locked
4._after_receiving.vmem	Wallet with custom addresses and incoming transactions	Unlocked
5._after_sending.vmem	Wallet with custom addresses and incoming and outgoing transactions	Locked
6._after_lock_and_quit.vmem	Wallet with custom addresses and incoming and outgoing transactions	Locked

Table 2
Monero images.

Image	Wallet state	Wallet Locked/Unlocked
1._after_creation.vmem	Never used encrypted wallet	Unlocked
2._after_unlock.vmem	Never used encrypted wallet	Unlocked
3._after_receiving.vmem	Wallet with incoming transactions	Unlocked
4._after_receiving_integratedaddress.vmem	Wallet with incoming transactions	Unlocked
5._after_sending.vmem	Wallet with incoming and outgoing transactions	Unlocked
6._after_sending_paymentid.vmem	Wallet with incoming and outgoing transactions	Unlocked
7._after_resolving.vmem	Wallet with incoming and outgoing transactions	Unlocked
8._after_wallet_close.vmem	Wallet with incoming and outgoing transactions	Locked

- Transaction ID of incoming transaction with amount XMR received
- Public address of own wallet

Memory image 4 (*4_after_sending.vmem*):
The fourth image was created after sending a transaction to another wallet. The following artefacts are found in this image:

- Wallet passphrase (ASCII and UTF16 format)
- Transaction ID of outgoing transaction with amount XMR
- Transaction ID of earlier transaction
- Public address of receiving party
- Public address of own wallet

Memory image 5 (*5_after_sending_paymentid.vmem*):
The fifth image was created after sending a transaction with a full payment id. This image contains the following artefacts:

- Wallet passphrase (ASCII and UTF16 format)
- Transaction ID of outgoing transaction with amount XMR
- Public wallet address of receiving party
- Public address of own wallet
- Full payment id of transaction
- Transaction IDs of earlier transactions

Memory image 6 (6_after_receiving_integratedaddress.vmem)
The sixth memory image was created after receiving a transaction with an integrated wallet address, this address includes a payment id within the public address. The following artefacts are found in this image:

- Wallet passphrase (ASCII and UTF16 format)
- Transaction ID of incoming transaction with amount XMR
- Public address of own wallet
- Public address of earlier receiving party
- Short payment id of transaction
- Transaction IDs of earlier transactions
- Full payment id of earlier transaction

The integrated address is not present in this memory image.
Memory image 7 (*7_after_resolving.vmem*):
The seventh image is created after an OpenAlias resolve action within the wallet software. The Monero donation address is resolved. The following artefacts are found in this image:

- Wallet passphrase (ASCII and UTF16 format)
- Public address of own wallet
- Public address of earlier receiving party
- Short payment id of earlier transaction
- Transaction IDs of earlier transactions
- Full payment id of earlier transaction
- Resolved public address with a description of the resolved address

Memory image 8 (*8_after_wallet_close.vmem*):
The last memory image was created after closure of the wallet. The following artefacts are found in memory after closure of the wallet software:

- Wallet passphrase (only UTF16 format)
- Public address of own wallet
- All transaction IDs of earlier transactions
- Full payment id of earlier transaction
- Short payment id of earlier transaction (belongs to integrated address transaction)

In every memory image there are several valuable forensic artefacts. In addition, in every memory image the wallet passphrase was present and that is valuable from a forensic perspective. For example, with a wallet passphrase you are able to gain full control over a wallet and all the funds inside.

6.1.2. Network traffic

All the network traffic during the experiment is captured with Wireshark with a capture filter. In total we created 6 different network capture files. One during wallet start-up and unlock, two during receiving funds in different ways, two during sending of funds and one capture file during resolving an OpenAlias address.

In the network traffic captures, there are many indicators confirming that Monero wallet software is running on the captured host. There is large Monero related DNS traffic which indicates the presence of the Monero wallet software. There is also traffic that looks like blockchain synchronisation packets because it includes block hashes. The only forensic value this traffic has is the ability to

Please cite this article as: Koerhuis, W et al., Forensic analysis of privacy-oriented cryptocurrencies, Forensic Science International: Digital Investigation, https://doi.org/10.1016/j.fsidi.2019.200891

prove that a Monero client is running on a system. We also did a search with all the keywords we used in the memory image experiment. There was only one hit and that was the donation public address of the Monero project. Because the OpenAlias mechanism is based on DNS this traffic is not encrypted, so the public donation address of the Monero project was visible in the network traffic. An interesting fact about the data returned is that it also included a Bitcoin address for Bitcoin donations.

6.1.3. Disk analysis

For the forensic disk analysis, we created a raw disk from the virtual machines hard disk. vmdk and mounted the raw disk as read-only. Then we used the program Bulk Extractor (https://www.forensicswiki. org/wiki/Bulk_extractor) to search for forensic artefacts. For this search, we used the keyword file with all the transaction and wallet related keywords as used in the memory image analysis.

There are only two interesting files with search hits found on the system, namely:

- *monero-wallet-gui.log*
- *M0n3r0wall3t.address.txt* (*M0n3r0wall3t* is the name of the wallet)

The first file is the general log file from the Monero GUI client. This file contains a lot of valuable forensic artefacts. In this file the following forensic artefact were present:

- Public address of own wallet
- Transaction ids of all transactions
- Amounts of XMR received and send

The *M0n3r0wall3t.address.txt* file present in the wallet directory contains the public address of the wallet and this is the only content this file has.

There are other interesting files found on the system that did not generate search hits like the wallets encrypted keys file. This file is present in the wallet directory and is called *{wallet_name}.keys*. This file contains the private keys of the wallet in encrypted format and can be used to recover a wallet when the passphrase is known. The passphrase was not found on the disk image but the memory image experiment revealed the passphrase in all images so this wallet file can be recovered. We did not find any other useful files with forensic use to gather information about the activities of a user's wallet.

6.2. Verge experiment results

During the Verge experiment we discovered an interesting fact while sending a transaction to a stealth address. When a sender initiates a transfer of funds to a stealth address the public one-time address is recorded in the transaction details of the sender. This public one-time address is added to the receiver's wallet and is visible in the blockchain. This one-time address can now be linked to the receiver and if the receiver wants to spend the coins present on that one-time public address that transaction is then linkable too.

6.2.1. Memory images

As mentioned in Section 5.2, we created 6 different memory images in different states of the virtual machine. After every memory creation, the virtual machine is shut down and rebooted. The forensic artefacts found in each memory image are as follows.

Memory image 1 *(1_after_creation.vmem)*:

After creation of the wallet and the manual encryption of the wallet with the predefined wallet passphrase we created the first memory image. The following artefacts are found in this image:

- Standard public address of wallet

The passphrase of the wallet was not found in memory. The public address found is the standard public address the software created during creation of the wallet.

Memory image 2 *(2_after_unlock.vmem)*:

The second memory image was created after the wallet was unlocked with the wallet passphrase via the debug console. When opening a wallet with the Verge software it does not require a passphrase to open the wallet even if it is encrypted. The passphrase is only needed when you add something to the wallet, for example a new public address or if you want to make a transaction. The wallet was unlocked with the following command: *walletpassphrase This-Is-The-Verge-Password 900 false*. The 900 represents 900 s that the wallet is unlocked and the false means that the wallet is completely unlocked.

In this memory image the following artefacts were found:

- Standard public address of wallet
- Wallet Passphrase (UTF16 format)

Memory image 3 (*3_after_address_creation.vmem)*:

This memory image was created after adding two new addresses to the wallet. A normal public address and a stealth address. Both addresses were created with a corresponding label *Normal_-Address_#1* and *Stealth_Address_#1*.

In this memory image the following artefacts are found:

- Standard public address of wallet
- Both created labels
- Newly created normal public address
- Newly created stealth address

For this memory image we did not unlock the wallet with the *walletpassphrase* command. When you add a new address to a wallet the Verge software asks for the passphrase before creating and adding the new address. This passphrase was not found in the memory image.

Memory image 4 (*4_after_receiving.vmem*):

The fourth memory image was created after the wallet received and confirmed two different transactions. The first transaction was sent to the newly created normal public address and the second transaction was send to the newly created stealth address. Both addresses were created in the previous step. The following artefacts were found in this memory image:

- Standard public address of wallet
- Both created labels
- Newly created normal public address
- Newly created stealth address
- Linked normal (one-time) public address to stealth address
- Wallet passphrase (UTF16 format)

When receiving funds, send to a stealth address, the Verge software automatically creates a new normal public address with the stealth address as a label. This newly created normal public address is also recorded in the Verge software of the sending party. This is the actual address the funds are send to and this address is recorded publicly in the blockchain. The wallet was unlocked with the *walletpassphrase* command. There was not any transaction id's present in this memory image.

Memory image 5 (*5_after_sending.vmem*):

Please cite this article as: Koerhuis, W et al., Forensic analysis of privacy-oriented cryptocurrencies, Forensic Science International: Digital Investigation, https://doi.org/10.1016/j.fsidi.2019.200891

This memory image was created after sending funds to two different addresses, a normal public address and a stealth address. The following artefacts were found in this memory image:

- Standard public address of wallet
- Both created labels
- Newly created normal public address
- Newly created stealth address
- Linked normal public address to stealth address (from memory image 4)
- Recipient stealth address with corresponding linked public address
- Recipient normal public address
- Transaction label, normal public address (this label and the recipient address is automatically added to the wallets address book.)

Also, in this memory image there was not any transaction ids present in memory. The wallet passphrase is also not present. This time we did not unlocked the wallet with the *walletpassphrase* command but filled in the passphrase when sending the funds just like we did with memory image 3. It seems that the Verge software does not store the plain text passphrase in memory when using this option.

Memory image 6 (*6_after_lock_and_quit.vmem*):

The last memory image was made after we manually locked the wallet and closed the Verge wallet software. We locked the wallet with the *walletlock* command within the debug console. The following artefacts were present in memory after locking and closure of the wallet.

- Standard public address of wallet
- Both created labels
- Newly created normal public address
- Newly created stealth public address
- Linked normal public address to stealth address (from memory image 4)
- Transaction label, normal public address (from memory image 5)
- Recipient normal public address (from memory image 5)
- Wallet passphrase (UTF16 format)

After manually locking the wallet and closing the wallet software the wallet passphrase is still available in memory. For this memory image we did unlock the wallet with the *walletpassphrase* command.

6.2.2. Network traffic

All the network traffic during the experiment is captured with Wireshark with a capture filter. In total we created 3 different network capture files. One during wallet start-up, one during receiving of funds and one during sending of funds.

The Verge wallet software uses the TOR network for all the network traffic of the Verge wallet software. We did not find any Verge related forensic artefacts in the network captures of the wallet software because all traffic is sent encrypted over the TOR network. From a privacy point of view the use of the TOR network is a good feature.

6.2.3. Disk analysis

Similar to Monero disk forensics, we also created a raw disk from the virtual machines hard disk. vmdk and mounted the raw disk as read-only. Then we used the program Bulk Extractor to search for forensic artefacts. For the search we also used the keyword file with all the transaction and wallet related keywords as used in the memory image analysis.

There are several interesting files present on the examined disk. All the interesting files are directly related to the Verge wallet software and are located in the hidden. VERGE directory inside the *users*' home directory. The interesting files are named:

- .VERGE/wallet.dat
- .VERGE/debug.log
- .VERGE/database/log.*

The wallet. dat file is the actual wallet file with all the private keys, this file can be used to recover a wallet or use the wallet on a different system. During the creation of the wallet we encrypted the wallet with the encrypt wallet option from the Verge wallet software. In this case encryption is a big word because the wallet still contains plain text addresses and doesn't look completely encrypted. Also, when an encrypted wallet is used on a different system you do not need a passphrase to open the wallet. Hence an investigator can see the complete detailed transaction history and all the public and stealth addresses a wallet contains without knowing the passphrase. In plain text the encrypted wallet. dat file contains the following forensic artefacts:

- Standard public address of wallet
- User created public and stealth address with corresponding labels
- Public address linked to stealth address of wallet
- Public address of recipient.
- All user created labels present in the wallet

To complete this list the debug. log file contains almost the same data as the wallet. dat file with some interesting additions. The debug. log file contains all the transaction ids (incoming/outgoing) and only the amounts of Verge send of the outgoing transactions of the wallet. This information was not retrieved from the memory images and this is the first location this data is encountered in plain text.

Inside the database directory there are some log files with some interesting forensic artefacts. These artefacts can also be found elsewhere, and these files contain the same information as the encrypted *wallet. dat* file contains in plain text.

6.3. Discussion

The ability for an investigator to 'follow the money' can be achieved partially with transaction ids, public addresses and of course completely with the seizure of a wallet because all the transactions are mathematically bound to a wallet. In addition to wallet passphrase, private key or mnemonic seed phrase, indicators of the use of a cryptocurrency in for example network traffic from a wiretap can also be a valuable asset for the preparation of forensic research.

Valuable forensic artefacts are present in almost all sources we analysed. The only source in which we did not find any relevant forensic artefact was the network traffic captures from the Verge experiment. This is the direct result of the use of the build-in TOR software for the encryption of the network traffic of the Verge wallet software. However, valuable forensic artefacts were present in all memory images from the Monero experiment as well as in the Verge experiment. But there are some important differences.

To open the Monero wallet software we need to enter the passphrase and probably therefore this passphrase is present in all memory images and even after closure of the wallet software. This is different with the Verge wallet software. With the Verge software we do not have to enter the passphrase to open the wallet and look

Please cite this article as: Koerhuis, W et al., Forensic analysis of privacy-oriented cryptocurrencies, Forensic Science International: Digital Investigation, https://doi.org/10.1016/j.fsidi.2019.200891

Table 3
Summary of valuable artefacts found.

Artefact/Source	Memory	Network	Disk	Value
Passphrase	Verge/Monero	–	–	Wallet seizure
Private keys	–	–	–	Wallet seizure
Mnemonic seed	Monero	–	–	Wallet seizure
Public address of wallet	Verge/Monero	–	Verge/Monero	Indicator of use, transaction identifier
Public address involved in transactions	Verge/Monero	–	Verge/Monero	Indicator of use, transaction identifier
Stealth address	Verge	–	Verge	Indicator of use, transaction identifier
Transaction ids	Monero	–	Verge/Monero	Indicator of use, transaction identifier
Transaction amounts	Verge	–	Verge/Monero	Indicator of use, transaction identifier
Labels	Verge	–	Verge	Indicator of use, transaction identifier
External indicators of use	–	Monero	–	Forensic research preparation

at the transactions or the balance of the wallet. You do have to enter your passphrase when you want to transfer funds. But this passphrase was not found in memory, the only time Verge wallet software stored the passphrase in memory was when we manually unlocked the wallet with the *walletpassphrase* command. So, the seizure of a Verge wallet is only possible when the wallet was unlocked with the *walletpassphrase* command and in our opinion, there is no real need for manually unlocking the wallet. The seizure with the passphrase from the Monero wallet is possible in all analysed situations. Private keys which give an investigator the ability to seize funds from a wallet were not present in memory at all in both experiments.

Other than Verge (Verge does not use a seed), Monero uses a mnemonic seed and this seed is present in memory right after the wallet is created. With this seed an investigator is able to recover a wallet and thus seize the contents. In our opinion this is not a very likely state an investigator seizes the memory of a computer system in real life.

Other valuable forensic artefacts like public addresses, stealth addresses, transaction ids and transaction amounts are present in different sources on the examined systems. In Table 3 we give an overview of all the forensic artefacts found.

In both the Monero and Verge experiments there are valuable forensic artefacts in different sources of evidence with forensic value varying from the seizure of the wallet and thus the funds inside to indicators of the use of specific cryptocurrency software. The memory images were created in different states of the analysed systems and therefore in most circumstances practically useful. With the results from the experiments an investigator now has knowledge what artefacts should be present in different memory images and thus can search for those artefacts with for example regular expressions. Not all artefacts follow a pre-defined pattern and therefore searchable with a regular expression, for example a passphrase. A solution to this problem can be to generate a wordlist from all the strings present in memory and use that wordlist to brute-force the wallet software to gain access. Be sure to include wide space characters as well.

The results from the disk analysis experiments are only practically useful if the disk is not fully encrypted and files can be examined in an unencrypted state. Network captures from the Verge experiment are not practically useful because all the traffic is encrypted due to the use of the built-in TOR software. Network captures from the Monero experiments can be practically used and forensic artefacts are available in these captures. Especially when starting up the wallet application there is a lot of relevant artefacts and DNS traffic was also found in the network captures.

Some other researchers (Van der Horst et al., 2017) (Zollner et al., 2019) (Montanez, 2014) did investigate what kind of forensic artefacts cryptocurrencies can leave on a computer system. These researchers conducted their works on the Windows operating system or mobile operating systems, and they investigated multiple wallet applications of the Bitcoin cryptocurrency. None of the researchers researched the wallet software of the privacy-oriented cryptocurrencies.

7. Conclusion and future work

This paper contributed to the forensic and academic world in several ways. It provided a forensic investigation conducted on a Linux operating system on the forensic artefacts these cryptocurrencies wallet software may leave on a computer system. This provides an investigator important knowledge on the presence of certain forensic artefacts needed for example the seizure of a wallet and thus the funds inside. On the other hand, it also addresses the need for further research in the emerging world of cryptocurrencies. This paper shows that even privacy-oriented cryptocurrencies can still be traceable, and the wallets can be seized because of the shortcomings in the wallet software. In our opinion there is no need for plain text passwords in volatile memory and certainly not after the wallet software is closed. Log files present in both experiments contain a lot of interesting information, for the sake of privacy and anonymity these log files can easily be encrypted to provide better privacy. Also the *wallet. dat* file of the Verge experiment is not fully encrypted, this is also an important source of information that can be fixed pretty easily with a software update. As said earlier cryptocurrencies are fairly new and software updates come out regularly so these shortcomings can be addresses in future releases.

Future research can be done on a Windows or MacOS systems as well as Web Wallets installed on privacy preserving web browsers (Warren et al., 2017). Different operating systems use different file systems and therefore it is possible that some file systems contain different artefacts than others. Future research can also be done on the source code of both currencies because both are open source projects. Valuable information about the inner workings of the software could be retrieved from the source code.

References

99Bitcoins. Top seven ways your identity can Be linked to your Bitcoin address [Online]. Available: https://99bitcoins.com/know-more-top-seven-waysyour-identity-can-be-linked-to-your-bitcoin-address/. (Accessed 21 May 2019).

Abrams, L.. Star trek themed kirk ransomware brings us monero and a spock decryptor! [Online]. Available: https://www.bleepingcomputer.com/news/security/star-trek-themed-kirkransomware-brings-us-monero-and-a-spock-decryptor/ [Accessed May 2018].

Alonso, K.M., 2017. Monero - Privacy in the Blockchain. Universitat Oberta de Catalunya.

Androulaki, E., Karame, G.O., Roeschlin, M., Scherer, T., Capkun, S., 2013. "Evaluating User Privacy in Bitcoin," in Financial Cryptography and Data Security. Okinawa.

Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J.A., Felten, E.W., July 2015. Research perspectives and challenges for Bitcoin and cryptocurrencies. In: IEEE Symposium on Security and Privacy, CA, USA.

Casey, E., 2009. Handbook of Digital Forensics and Investigation. Academic Press. October 2009.

Chan, S., Chu, J., Nadarajah, S., 2017. J. Osterrieder. "A statistical analysis of cryptocurrencies". J. Risk Financ. Manag. 10 (2), 12.

Please cite this article as: Koerhuis, W et al., Forensic analysis of privacy-oriented cryptocurrencies, Forensic Science International: Digital Investigation, https://doi.org/10.1016/j.fsidi.2019.200891

Coinmarketcap charts. https://coinmarketcap.com/.
Conti, M., Kumar, S.E., Lal, C., Ruj, S., 2018. A survey on security and privacy issues of Bitcoin. IEEE Communications Surveys & Tutorials 20 (4), 3416–3452.
CryptoRekt, 2017. Verge Currency Blackpaper v3.0.
Doran, M.D., 2014. A Forensic Look at Bitcoin Cryptocurrency" ProQuest LLC (2014) UMI 1554438, 2014.
Europol, 2017. IOCTA 2017, Internet Organised Crime Threat Assessment. Europol, Den Haag.
Grunzweig, J.. The rise of the cryptocurrency miners," paloalto networks [Online]. Available: https://research.center.paloaltonetworks.com/2018/06/unit42-rise-cryptocurrencyminers/ [Accessed April 2019].
Haigh, T., Breitinger, F., Baggili, I., 2019. If I had a million cryptos: cryptowallet application analysis and a trojan proof-of-concept. In: Breitinger, F., Baggili, I. (Eds.), Digital Forensics and Cyber Crime. ICDF2C 2018. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 259. Springer.
Hankerson, D., Menezes, A.J., Vanstone, S., 2003. Guide to Elliptic Curve Cryptography. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
Meiklejohn, S., et al., 2016. A fistful of bitcoins: characterizing payments among men with No names. Commun. ACM 59 (4), 86–93.
Monero. Monero | Monero - secure, private, untraceable. Monero, [Online]. Available: https://getmonero.org/ [Accessed May 2019].
Montanez, A., 2014. Investigation of Cryptocurrency Wallets on iOS and Android Mobile Devices for Potential Forensic Artifacts. Department of Forensic Science, Marshall University.
M. Moser et al., "An empirical analysis of traceability in the Monero blockchain," Proceedings on Privacy Enhancing Technologies, vol.2018 (3), pp:143-163.
Mursch, T.. Large cryptojacking campaign targeting vulnerable Drupal websites [Online]. Available: https://badpackets.net/large-cryptojackingcampaign-targeting-vulnerable-drupal-websites/ [Accessed May 2019].
Neilson, D., Hara, S., Mitchell, I., 2016. Bitcoin forensics: a tutorial. In: Jahankhani, H., et al. (Eds.), Global Security, Safety and Sustainability - the Security Challenges of the Connected World. ICGS3 2017. Communications in Computer and Information Science, vol. 630. Springer.
OpenOne Labs, 2019. WTF are all of these Monero Keys?? [Online]. Available: https://openonelabs.com/wtf-are-these-monero-keys/ [Accessed May 2019].
Orcutt, M.. Criminals thought Bitcoin was the perfect hiding place, but they thought wrong [Online]. Available: https://www.technologyreview.com/s/608763/criminals-thought-bitcoin-was-theperfect-hiding-place-they-thought-wrong/ [Accessed April 2018].
Reed, T.. New Mac cryptominer uses XMRig, 22 5 2018. [Online]. Available: https://blog.malwarebytes.com/threat -analysis/mac-threat-analysis/2018/05/newmac-cryptominer-uses-xmrig/ [Accessed May 2019].
Reid, F., Harrigan, M., 2013. An analysis of anonymity in the Bitcoin system. In: Altshuler, Y., Elovici, Y., Cremers, A., Aharony, N., Pentland, A. (Eds.), Security and Privacy in Social Networks. Springer, New York, NY.
Van der Horst, L., Choo, K.-K.R., Le-Khac, N.A., 2017. Process memory investigation of the Bitcoin clients Electrum and Bitcoin core. IEEE Access 5 (1). https://doi.org/10.1109/ACCESS.2017.2759766.
Verge Currency. Privacy as a choice. A secure and anonymous cryptocurrency [Online]. Available: https://vergecurrency.com.
Warren, C., El-Sheikh, E., Le-Khac, N.A., 2017. Privacy preserving internet browsers – forensic analysis of browzar. In: Daimi, K. (Ed.), Computer and Network Security Essentials. Springer, Cham. https://doi.org/10.1007/978-3-319-58424-9_21.
Williams, S.. 3 cryptocurrencies that rose by more than 100,000% in 2017" Fool.com, 29 12 2017. [Online]. Available: https://www.fool.com/investing/2017/12/29/3-cryptocurrencies-that-rose-by-morethan-100000-i.aspx [Accessed April 2019].
Zollner, S., Choo, K.-K.R., Le-Khac, N.A., 2019. An automated live forensic and postmortem analysis tool for Bitcoin on Windows systems. IEEE Access 7. https://doi.org/10.1109/ACCESS.2019.2948774.

Please cite this article as: Koerhuis, W et al., Forensic analysis of privacy-oriented cryptocurrencies, Forensic Science International: Digital Investigation, https://doi.org/10.1016/j.fsidi.2019.200891