Visualising Bitcoin Flows of Ransomware: WannaCry One Week Later

Stefano Bistarelli[1], Matteo Parroccini[1], and Francesco Santini[1]

Department of Mathematics and Computer Science, University of Perugia, Italy
[stefano.bistarelli, matteo.parroccini, francesco.santini]@unipg.it

Abstract

Because of its pseudo-anonimity and decentralisation characteristics, bitcoin payments are often a tool utilised by ransomware: this kind of malware infects a victim computer by encrypting some/all its data and/or denying the access to it. Then, the victim has to pay a given amount of bitcoins to see all the blocked functionalities restored. The goal of this paper is to visualise these bitcoin transactions, and in particular we focus on the effects of one of such ransomware, i.e., *WannaCry*, one/two weeks after its diffusion. We exploit *BlockChainVis*, a tool for visualising flows of bitcoins through the use of Visual Analytics.

1 Introduction

The white-paper on Bitcoin appeared in November 2008 [5], written by a computer programmer using the pseudonym "Satoshi Nakamoto". His invention is an open-source, peer-to-peer digital currency. Money transactions do not require a third-party intermediary, with no traditional financial-institution involved in transactions: the Bitcoin network is completely decentralised. A complete transaction record of every bitcoin and every Bitcoin user's encrypted identity is maintained on a public ledger, called the *block-chain*. For this reason, Bitcoin transactions are thought to be *pseudonymous*, not completely anonymous.

The actors in the Bitcoin network are the *users* who own a *wallet* associated with a couple (or more) of private/public cryptographic keys. A private key is usually a 256 bit random number, and by using the *Elliptic Curve Digital Signature Algorithm* (*ECDSA*) [2], a 512 bit public key can be obtained from it. Afterwards, from the public key it is possible to obtain a Bitcoin *address*, e.g., applying an hashing function on it. Users use these keys to sign the transactions they generate in order to transfer their money to other users; transactions are then broadcast to the Bitcoin peer-to-peer network.

Transactions represent the mechanism that allows a user to cede money to another user. A user can prepare a new transaction referring to the ones through which she received money, called the (multiple) *inputs* of this new transaction. The *output* of a transaction describes the destination of bitcoins instead. There can be multiple *outputs*, allowing a owner to make multiple payments at once; one output often represent the change w.r.t. a previous transaction.

Miners keep the block-chain consistent, complete, and unalterable: they repeatedly verify and collect newly broadcast transactions into a new group of transactions, called a *block*. In order to validate a block, a miner needs to compute a random nonce that becomes part of a block and makes it have a hash that starts with a given amount of zeroes (i.e., the *proof-of-work*). This proof is easy to verify, but extremely time-consuming to generate.

Ransomware [3] is a software that performs a cryptoviral extortion attack that encrypts data until a ransom is paid to a given bitcoin address. Thus, ransomware leads to a denial-of-access attack that prevents users from accessing files on the infected computer. Some sadly-famous names of such software are *Cryptolocker*, *Cryptowall*, *TeslaCrypt*, and *Locky*. For what concerns Cryptolocker, it affected 500,000 users until 2014, and an analysis indicates that only 1.3% of

all the users hit by the malware paid the ransom of 400$.[1] A more recent and very effective piece of ransomware, which started to spread on May 12th 2017, is WannaCry.

In this work we show how *BlockChainVis* [1] (Sec. 2) visualises all the Bitcoin transactions that have as output one of the addresses ascribed to WannaCry. BlockChainVis is dedicated to the visual analysis of flows of bitcoin transactions. Since the block-chain is an example of Big Data, a straightforward visualisation in its entirety is not very significant. Hence, we have exploited some techniques from *Visual Analytics* (*VA*) [6] to filter out undesired information, with the purpose to obtain a forensic-tool to efficiently and visually analyse the block-chain and help investigations.

2 BlockChainVis

BlockChainVis [1] is a client-server Web-application. It consists of a back-end (server-side) and a front-end (client-side). The client can be directly tested online with any browser. The main technologies used for the back-end are *OrientDB*[2], *PHP*, *Node.js*[3], and *Bitcore*[4], while the ones for the front-end are *HTML5*, *CSS3*, *Javascript*, and *D3.js*[5].

As a first step, the tool can download the entire block-chain on the back-end; in fact, to intensively work on it, after some initial attempts we discarded the idea to use a *block-explorer* because of their current limitations (traffic volumes and omission of some information). Block-explorers are Web-sites that allows for reviewing information about the block-chain, by using dedicated Web-services. Therefore, we opted for *Bitcore*, which is a *full*[6] Bitcoin-client. Bitcore (its block-chain) can be queried by using *Insight API*, and the result is presented as a *JavaScript Object Notation* (*JSON*) file, which is a simple text-document where the basic structure are a set of name-value pairs and an ordered list of values.

The second step consists in extracting the desired information from the block-chain in order to populate a relational database: *Postgres*. The PHP script that accomplishes this task is *getBlock.php*, which takes as input a range of blocks. Starting form the first one, by calling the API offered by BitCore, the script extracts all the information related to the block (the result of the invocation is a *json* file) and encodes it in a PHP object to better handle it.

The back-end of BlockChainVis is implemented on a machine with 128Gbyte of RAM, two processors Intel(R) Xeon(R) CPU E5-2620 v4 2.10GHz 8 core (for a total of 16 cores and 32 threads); in particular, the implementation consists of three different virtual machines running, *i)* Bitcore, *ii)* OrientDB, and *iii)* software dedicated to visualisation.

Big Data analytics examines large amounts of data to uncover hidden patterns, correlations and other insights. The block-chain can be considered as Big Data: by mid October 2017, the block-chain contains over than 262 million transactions, for more than 137,000Mbyte. For this reason we turned our attention to *Visual Analytics* [6] (*VA*), that is the science of analytical reasoning facilitated by interactive visual-interfaces. The main aim of VA is to help the visualisation of problems whose size, complexity, and need for closely coupled human-machine analysis may make them otherwise intractable.

Being VA task-oriented [6], we identified nine main tasks: *i)* to find miners; *ii)* find transaction sources and understand how they are connected; *iii)* find the main addressees of trans-

[1] http://www.bbc.com/news/technology-28661463.

[2] http://orientdb.com/orientdb/. A successive version of BlockChainVis will also use PostgreSQL as relational Data-base: https://www.postgresql.org.

[3] https://nodejs.org/en/.

[4] https://bitcore.io/.

[5] https://github.com/aaronpowell/db.js/.

[6] Full nodes download every block in the block-chain.

actions; *iv)* find the "richest" and "poorest" addresses; *v)* find the addresses with a break-even budget; *vi)* find bitcoin flows from an arbitrary address; *vii)* find bitcoin flows from a set of different addresses; *viii)* filter the block-chain on intervals of time or block identifiers; *ix)* filter the block-chain on specific transaction amounts, or on the number of involved addresses.

3 Case Study

WannaCry (alternatively *WannaCrypt*, *WanaCrypt0r 2.0*, *Wanna Decryptor*) is a ransomware computer-worm that targets the family of *Microsoft Windows* operating systems. It was first discovered on May 12th 2017 (Friday), and it has infected more than 200,000 computers in over 150 countries.[7] The same day, a security-researcher arrested the spread of WannaCry when he discovered some traffic directed to an unregistered domain from a copy of WannaCry he was testing. By registering *iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com*, he stopped the infection by using this "kill switch" designed to control it.[8]

WannaCry demands that the victim to pay a ransom of 300$ in bitcoins at the time of infection, which doubles to 600$ after three days. After seven days without payment, data is permanently unrecoverable. This ransomware encrypts nearly any important file type a user might have on her computer, e.g., .png, .zip, .jpg, .docx, and .rtf. WannaCry exploits a vulnerability in Microsoft's implementation of the *Server Message Block* (*SMB*) protocol, in order to take control of the target and infect it (and then encrypt victim's data).

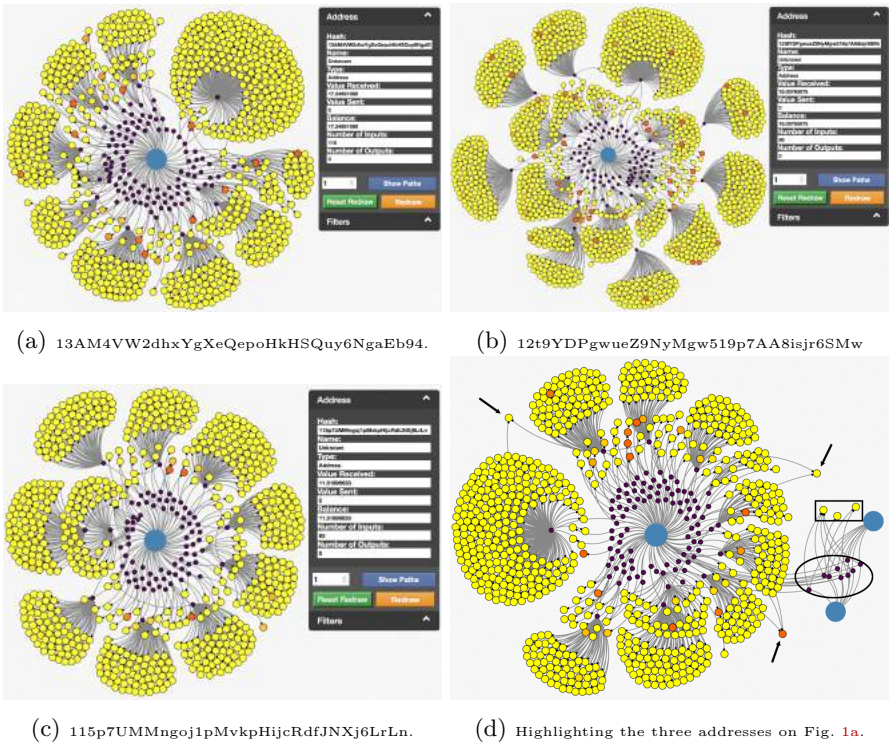
In our study, we visualise the flows of money related to the top three known WannCry Bitcoin addresses by incoming budget, where a victim is required to send the ransom.[9] We report these flows in Fig. 1, where we filter out all the blocks outside the range 465960-466949, that is we consider all the transactions mined between 2017-05-12 00:21:05 and 2017-05-18 09:42:34. In addition, we filter out all the inputs of the transactions, in order to only focus on final money-destinations. Figure 1a, Fig. 1b, and Fig. 1c highlight the transactions that respectively concern such three addresses (at the centre of each image), i.e., all the transactions for which one of the outputs is one of these three addresses. Figure 1d shows Fig. 1a by highlighting all the three incriminated addresses in the same image (bigger nodes). In such images, collected by using BlockChainVis (see Sec. 2), smaller dark nodes represent transactions, and all larger and lighter ones represent the involved addresses of such transactions.

The sum of money moved to such three addresses during the first week is $17.247 + 16.037 + 11.518 = 44.802$Ƀ. We do not provide amounts in dollars/euros because of high fluctuations of Bitcoin in 2017. The total number of inputs that concerns all three of them is 110+96+82 = 288.

From the flows reported in Fig. 1 we notice some interesting features. First, most of the transactions just have one output to only one of the three incriminated addresses (no payer splits the ransom among them), where they transfer an amount of bitcoins very close to 300$ or 600$ at the exchange rate of that period (i.e., exactly what requested by WannaCry): between around 0.15Ƀ and 0.18Ƀ (resp., 0.3Ƀ-0.36Ƀ). By considering all the 288 transactions, there is only one of them (for the address in Fig. 1b) that paid a higher ransom: 1.99Ƀ, which correspond to 11 infected machines.

If we instead consider the whole history until May 26th (two weeks later the diffusion), we notice that *i)* no transaction is dated before May 12th, and *ii)* there is still no outbound transaction at the moment (also visible in Fig. 1 for the first week): ransom funds remain unspent. Moreover, we notice that the number of input transactions increases from 288 to 333,

[7] https://blog.kaspersky.com/wannacry-ransomware/16518/.
[8] http://tinyurl.com/k7ea9y2.
[9] Addresses: https://github.com/GregorSpagnolo/WannaCrypt.



(a) 13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94. (b) 12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw

(c) 115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn. (d) Highlighting the three addresses on Fig. 1a.

Figure 1: The flows towards three different addresses used by WannaCry.

with only new 45 transactions in the second week, and the total balance of the three nodes increases from 44.802Ƀ to 50.14Ƀ: 89% of the ransoms has been collected during the first week. Considering these 333 transactions, 177 transfer $[0.1, 0.2)$Ƀ, while 41 move $[0.3, 0.4)$Ƀ as ransom. The second group is concentrated in the last period of the studied time-interval, meaning that victims that paid after three days, actually paid twice the ransoms, exactly following WannaCry instructions (unless those few cases where they pay for two infected machines). In addition, 85 of such 333 transactions are less than 0.01Ƀ ($\sim$ 2$ at that time). Therefore, in total we estimate that no more than $333 - 85 = 248$ victims paid the requested ransom during the first two weeks, focusing the three investigated addresses. Our hypothesis is that such many low-value transactions may collect payment errors or first attempts, hidden messages (see in the following), or just the will to appear in such a list.

Looking at Fig. 1, some transactions show a high number of outputs, visually corresponding to the "flowers with many petals"; for instance, there are 12 of such many-output transactions in Fig. 1d, and the largest of them has 246 outputs. For all the 12 "flowers", most of these outputs receive a lesser amount of bitcoins, while a few addresses receive more than the ransom. If we investigate the largest of them, it moves a total amount of 147.83Ƀ, but 144 addresses (out of 246) receive less than 0.1Ƀ. Six of these receivers belong to *Poloniex.com*[10], a US-based cryptocurrency exchange and lending service provider. Some other addresses refer to different betting, investing, or wallet services, e.g. *Cubits.com*[11].

[10] https://poloniex.com.
[11] https://cubits.com.

A second characteristic is the presence of three addresses (pointed by arrows in Fig. 1d) that are a common output of two/three different transactions used to pay a ransom. One of them belongs to Poloniex.com. The second address is linked only to the two transactions used to pay two ransoms, and has a low unspent budget (0.45฿). The third address has been involved in 1,073 transactions and received more than 169฿ (with a current null balance): the last transaction is towards an online gambling platform.

A third feature observable in Fig. 1d is a set of 9 transactions (grouped by an ellipses) that moved some bitcoin to all the three WannaCry addresses. However, such amounts do not correspond to what due for a ransom, but they are less than 1$. Seven (out of 9) of such transactions have a fourth output (one of them has also a fifth); such addresses are inside a rectangle in Fig. 1d. These transactions consists in messages sent to WannaCry addresses to reach high visibility. Five of them only have one input address, but all 5 addresses start with the string "1DoDiK". One more of transaction has four input addresses, each of them containing part of an insult: "You are a ****". Finally, a transaction has two inputs whose sub-strings advertise a different crypto-currency: "Use ****".

4 Conclusion and Future Work

We have introduced BlockChainVis, a tool for visualising the Bitcoin block-chain and help digital forensics-investigations. Then we have proposed a case study concerning the visualisation of all the ransoms paid in one week due to the WannaCry ransomware. Related work as [4], excluded from this paper for the sake of brevity, can be found in [1].

We are currently extending BlockChainVis to encompass explicit features that are oriented to digital forensics. For instance, we will try to identify *mixing services* (also called *tumblers*), which can be used to mix money of a ransomware address with other users' money, intending to confuse the trail back to the original source and thus launder money [3]. Moreover, we will also characterise other unexpected flows of money, for instance immediately reporting newly created addresses whose incoming balance has rapidly increased: this could help to quickly find addresses linked to ransomware effects.

References

[1] Stefano Bistarelli and Francesco Santini. Go with the -bitcoin- flow, with visual analytics. In *Proceedings of the 12th International Conference on Availability, Reliability and Security*, pages 38:1–38:6. ACM, 2017.

[2] Don Johnson, Alfred Menezes, and Scott Vanstone. The elliptic curve digital signature algorithm (ECDSA). *International Journal of Information Security*, 1(1):36–63, 2001.

[3] Amin Kharraz, William K. Robertson, Davide Balzarotti, Leyla Bilge, and Engin Kirda. Cutting the gordian knot: A look under the hood of ransomware attacks. In *Detection of Intrusions and Malware, and Vulnerability Assessment DIMVA*, volume 9148 of *LNCS*, pages 3–24. Springer, 2015.

[4] Christoph Kinkeldey, Jean-Daniel Fekete, and Petra Isenberg. BitConduite: Visualizing and Analyzing Activity on the Bitcoin Network. In Anna Puig Puig and Tobias Isenberg, editors, *EuroVis 2017 - Posters*. The Eurographics Association, 2017.

[5] S. Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. http://www.hashcash.org/papers/hashcash.pdf, 2008. [Online; accessed 21-July-2016].

[6] Pak Chung Wong and Jim Thomas. Visual analytics. *IEEE Comput. Graph. Appl.*, 24(5):20–21, 2004.

Appendix

The application used to study WannaCry, named *BlockChainVis*, is accessible from the link http://www.dmi.unipg.it/blockchainvis/. If user and password are required to access, please use *user* and *test* respectively. A screenshot of the first page is reported in Fig. 2. From here it is possible to access to visualise by transaction id, address id, or the whole archipelago of islands. The archipelago view displays all the islands of the archipelago of transactions. An island is a connected component of a graph, where each couple of nodes is connected through a path, and each of the nodes is not connected to any other vertex of the super-graph.

Figure 2: A screenshot of the first page that appears to the user, where it is posible to visualise by transaction id, address id, and the whole archipelago of islands.

As it can be seen from Fig. 3(a), the number of islands is too large to be useful. For this reason we have created four slide-bars, each one operating on a different data-filter:

- A block-interval filter, by date or by height position (from-to) in the block-chain. Only the transactions in such blocks are visualised in the archipelago.
- A filter on the number of transactions: only the islands with the specified minimum and maximum number of transactions are shown.
- A value-based filter: it specifies the minimum and maximum amount of bitcoins considering all the transactions of a single island (i.e., their sum).
- A filter on the number of miners: it specifies the minimum and maximum number of miners in each visualised island.

In Fig. 3(b) we lighten the visualisation w.r.t. Fig. 3(a) by only showing islands with 2-10 miners. We obtain 354 islands (20% of the total): most of the islands only have one miner.

By clicking on any island of the archipelago, a summary of its statistics pops-up. It is possible to enter into an island and visualise all its transactions. In Fig. 4 we show the view of a single island. Such a visualisation employs an oriented graph: a node can represent either a transaction or an address, and each transaction may have 1-n outgoing edges and 1-n incoming

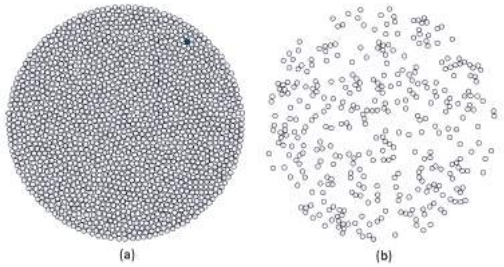


Figure 3: (a) The whole Bitcoin archipelago, and (b) by keeping only islands with 2-10 miners (min-max).

edges. The graph is bipartite: transactions can only be connected to addresses, and vice-versa. Larger nodes are addresses: they are lighter if their budget is balanced (bitcoin inputs equal to outputs). Smaller nodes represent transactions, and their colour is darker if the amount of transferred bitcoins is larger.

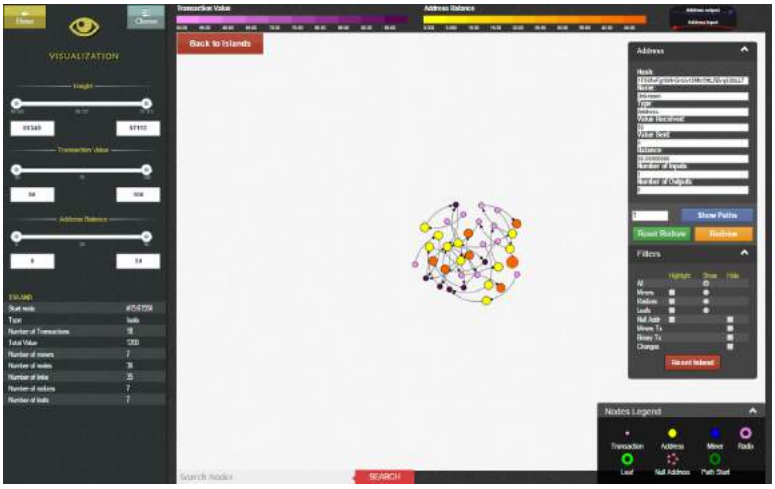


Figure 4: The view of an island of transactions.

From the page in Fig. 4 it is possible to apply different filters to highlight or exclude information form the visualisation. To this aim, BlockChainVis has some filters to *i)* show only the roots of an island, *ii)* only the leaves, *iii)* hide the transactions with a fee, *iv)* collapse binary transactions, and *v)* collapse changes: with *iv)* we hide the nodes that represent the transactions with only one input and only one output, while with *v)* we hide the transactions that return a change to the same address. The effect of some of the implemented filters is shown in Fig. 5.

In order to study WannaCry we have visualised the three different Bitcoin addresses known to belong to criminals. The, we have applied the previous filters to highlight destination nodes, in order to obtain the images in Fig. 1.

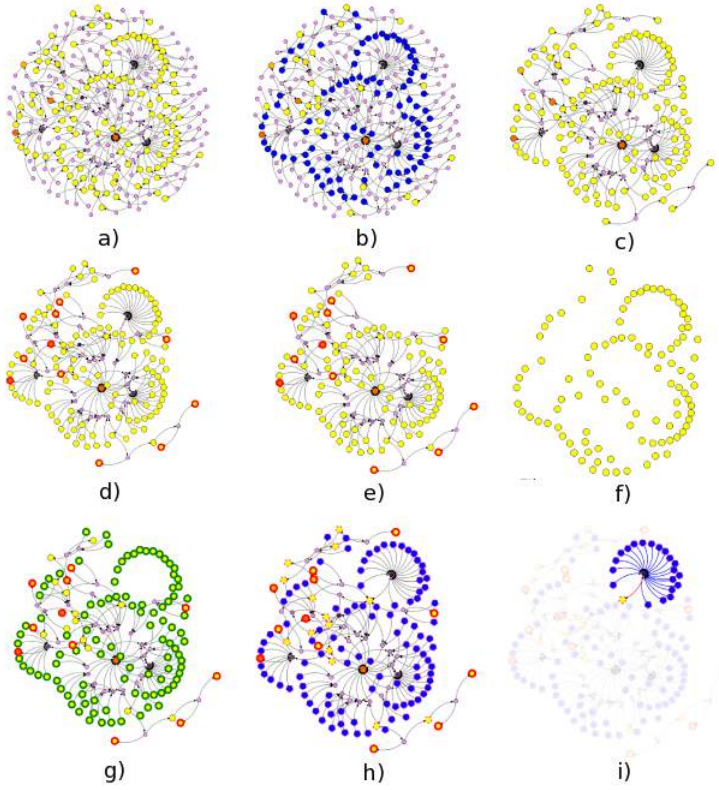


Figure 5: An example of filter application on (a) the initial graph: (b) by highlighting miners, (c) hiding coinbase transactions (rewarding the miners), (d) highlighting leaves, (e) applying a transaction-value interval, (f) showing only roots, (g) visualising paths (most of the two-colour nodes are the same roots in (f), the others are the leafs), (h) combines b and f together (darker nodes highlight the same miners in (b)), (i) focuses on a given transaction.