EGRET: Extortion Graph Exploration Techniques in the Bitcoin Network

Silivanxay Phetsouvanh, Frédérique Oggier and Anwitaman Datta
Nanyang Technological University, Singapore
Email: silivanx002@e.ntu.edu.sg, frederique@ntu.edu.sg, anwitaman@ntu.edu.sg

***Abstract*—The Bitcoin network is a complex network that records anonymous financial transactions while encapsulating the relationships among its pseudonymous users. This paper proposes graph mining techniques to explore the relationships among wallet addresses (pseudonyms for Bitcoin users) suspected to be involved in a given extortion racket, exploiting the anonymity of the Bitcoin network to collect and launder money. Starting around Bitcoin addresses of potential interest, neighborhood subgraphs are analyzed in terms of path length and confluence to detect suspicious Bitcoin flow and other wallet addresses controlled by the suspected perpetrators. We show with a dataset of the Ashley Madison blackmail campaign from August 2015 how the mechanisms can be used both to estimate the amount of money that was extorted by the suspected perpetrators under the specific blackmail campaign, and also estimate the amount of money handled by them during the same period of time.**

I. Introduction

Bitcoin is a cryptocurrency [1] relying on a decentralized back-end infrastructure to publicly record all transactions. Yet the users involved in these transactions retain anonymity via cryptographic mechanisms. Studying and understanding Bitcoin related activities is of interest for purposes as diverse as policy framing, investment planning, or forensic investigation, since Bitcoin data is both publicly available for anyone to analyze, and representative of an actual financial ecosystem.

Because of their anonymous nature (wallet *addresses* are used for transactions, but the address owner is hidden, and a single person may control arbitrarily many addresses), cryptocurrencies have become a de facto medium for illegal activities such as online black markets (for selling e.g. drugs or stolen credit card information), carrying out extortion, receiving and laundering money. Studying the latter in terms of detecting Bitcoin flow involving suspicious wallet addresses and in turn singling out new addresses potentially controlled by the (same group of) perpetrators is the focus of this paper.

Analysis and forensics of the Bitcoin network, especially in the early years, mostly relied on the idea of clubbing co-paying addresses together [1], [2], [3], [4], [5], [6], and guessing the address used to collect the change in a transaction (e.g., [7], [8]). More precisely, when inputs from multiple wallet addresses are combined together as inputs to a single Bitcoin transaction, and there are at most two outputs for such a transaction (one for the intended payee, one to collect the change back), then it was assumed that all the input addresses of the transaction belong to a single entity. Likewise, when there are two outputs, the output with the smaller amount is considered as a 'shadow' address to collect 'change', while the output with the larger amount is considered to belong to a third party to whom money is being paid [7]. Someone deliberately trying to blindside such a heuristic could create two outputs, one very small, and one large, and the heuristic would only club the address receiving the smaller output with the input addresses, ignoring the other output address. In fact, we witness this particular evasive behavior in the dataset that we shall study (in Section IV). We propose a path confluence based address aggregation mechanism which is robust against such an evasion, since it follows all the flows of money.

Furthermore, clubbing co-paying addresses extends to the case where some of the addresses are reused across multiple transactions, since then multiple such sets of aggregated addresses with non-empty intersection can be coalesced together. The proliferation of mixing (a process where multiple users come together to create a multi-input multi-output transactions) complicates forensics, since it obfuscates the linkage between a particular input of a transaction to a specific output, making it difficult to trace the flow of money. In analyzing the efficacy of the clubbing co-paying addresses heuristic, [4] notes that it is surprisingly effective, but that false positives are indeed caused by transactions which involve mixing.

Even though there is currently no absolute mechanism for deanonymization of Bitcoin users (see [9], [10] for surveys on both privacy enhancement and deanonymization techniques), the use of side (*off-chain*) information [9], [11] to identify real world identities has proven to be effective, and could be used in conjunction with the above heuristics. For instance, side information regarding a wallet address (associated with a set of wallet addresses allegedly involved in laundering money stolen from the now defunct Bitcoin exchange Mt. Gox) led to the identification (and eventual arrest) of Alexander Vinnik [12].

Gathering addresses belonging to the same (group of) users and possibly identifying them using side information is of relevance in the context of extortion scams. There are numerous and increasing amount of incidents, typically following the same pattern playing on potential victims' guilt: blackmail to not expose evidence of improper behavior (e.g. affair, porn watching, etc). The blackmail following Ashley Madison data breach [13] falls in that category, and is used as a case study for this paper. Ransomwares form another type of popular means of extortion (e.g., the WannaCry ransomware extortion [14]).

Our methodology to study Bitcoin extortion scams is as follows. We first identify some address/transaction subgraphs

1	Inputs: $\varnothing$ Outputs: 25.0 $\rightarrow$ addr$_A$
2	Inputs: 1[0] Outputs: 17.0 $\rightarrow$ addr$_B$, 8.0 $\rightarrow$ addr$_A$
3	Inputs: 2[0] Outputs: 8.0 $\rightarrow$ addr$_C$, 9.0 $\rightarrow$ addr$_B$
4	Inputs: 2[1] Outputs: 4.0 $\rightarrow$ addr$_D$, 2.0 $\rightarrow$ addr$_B$, 2.0 $\rightarrow$ addr$_A$
5	Inputs: 3[1], 4[1] Outputs: 11.0 $\rightarrow$ addr$_B$
6	Inputs: 3[0], 5[0] Outputs: 3.0 $\rightarrow$ addr$_D$, 7.0 $\rightarrow$ addr$_C$, 9.0 $\rightarrow$ addr$_B$

TABLE I: Some Examples of Transactions.

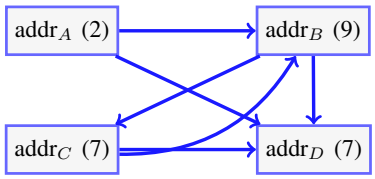

Fig. 1: An address network corresponding to a set of Bitcoin transactions. This is for the sake of illustration, when a transaction has several inputs and outputs, it is not always possible to reconstruct the list of exact transactions.

of the Bitcoin network that seem relevant to the scam to be studied (see Section II for background on the Bitcoin network and how we choose these subgraphs). We then look at the subgraph of interest as a realization of a random graph with a specific in-degree/out-degree node distribution, and look for abnormal patterns in terms of path behavior. To do so, we extend the work from [15] that studies the expected behavior of (power-law) graphs in terms of average path length, to the case of directed graphs (detailed in Section III). Nodes on the Bitcoin address graph that are connected to each other by a distance that is statistically significantly shorter than the expected average behavior as per the analysis, as well as nodes with high number of shortest paths among them can be discerned as outliers, and indicate that these nodes are involved in Bitcoin money flows with confined circulation. We accordingly design an algorithm (described in Section IV) to trace the confinement of money flows by establishing confluence of flow paths over the address network. This also enables us to identify new addresses to investigate, that need to be working in tandem with the original detected nodes, for realizing such specific circulations, and aggregate the new suspects with the original ones. Furthermore, by studying the transaction network, but considering only nodes and edges that involve the found addresses, we can estimate the total volume of money being handled by the (group of) users who are controlling the wallet addresses being analyzed.

The contributions of this paper are thus as follows:
(1) We extend the analysis from [15] to study the expected path lengths in directed graphs in order to identify outliers, specifically, close-knit nodes.
(2) We design a path confluence algorithm that provides a new way to aggregate suspicious Bitcoin addresses.
(3) We study one specific extortion - the Ashley Madison blackmail campaign from August 2015 [13] - to show step-by-step how the methodology can be applied. Specifically, we obtain an initial list of suspected addresses based on a particular amount of money that the extortionists have been demanding. We then establish a shortlist, by eliminating addresses which are way too far from each other on the address graph to be acting in tandem. We then focus on the shortlisted nodes, and neighboring nodes within a given radius, and establish if the shortlist of nodes have unexpectedly short distance among themselves according to the results of Section III. We conduct path confluence analysis, identification of further addresses and analysis of the transaction network (below) accordingly.
(4) We filter the corresponding Bitcoin transaction subnetworks to show information regarding only a properly aggregated subset of addresses that help us identify "sink" nodes, where money from the scam is being funnelled. We also estimate the amount of money controlled by the group of users controlling the suspicious addresses.

II. Directed Bitcoin (Sub)Networks

A. Bitcoin Background

The Bitcoin network is a peer-to-peer payment network, in which users send and receive the Bitcoin cryptocurrency, by broadcasting digitally signed messages to the network. Payments are grouped by transactions (see Fig. I for a toy example), and transactions are recorded into a public distributed database (ledger) called the blockchain.

An actual Bitcoin transaction comprises metadata (including the transaction hash), inputs and outputs [16]. The hash of the entire transaction is a unique ID for the transaction (in Figure I the transactions are labelled from 1 to 6 for simplification). Every input specifies a previous transaction hash, and the index of the previous transaction's output (labelled from 0) that is being spent. For example, the inputs of transaction 5 are 3[1] and 4[1], referring to the second output of transactions 3 and 4. Here, since the same address is used, this transaction is in fact a fund consolidation. Every output contains information that permits to confirm the recipient address, and the transaction amount. The sum of all the output has to be equal to or less (by a transaction fee) than the sum of the input amounts.

B. Bitcoin Network Models & Dataset

The Bitcoin network can be modelled in several ways: (1) as a heterogeneous (2-mode) graph, where nodes comprise transactions, identified by their unique hash, and Bitcoin wallet addresses (a user may own many wallet addresses), (2) as a directed transaction network (see Subsection IV-C), and (3) as a directed address network (see e.g. [17]), where each node represents a Bitcoin address, and there is a directed link between two nodes (Fig. 1) if there was at least one transaction among those addresses. This is the graph model we use next.

This paper will look at the Ashley Madison extortion incident that followed the data breach of Ashley Madison website. The website billed itself as an enabler of extra-marital affairs. In the extortion incident, the perpetrator(s) asked for a given amount of 1.05฿ from Ashley Madison

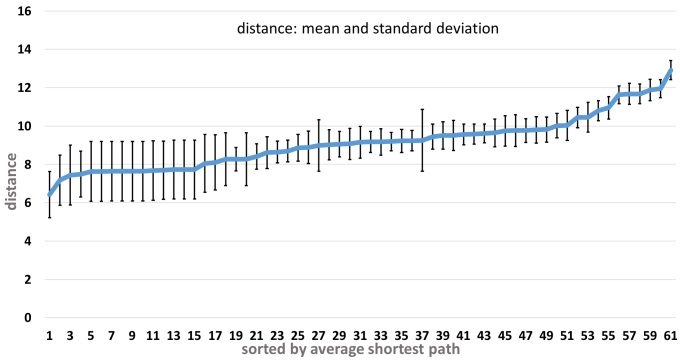

Fig. 2: Pair-wise distance statistics (average and standard deviation) among 61 suspect Bitcoin wallet addresses.

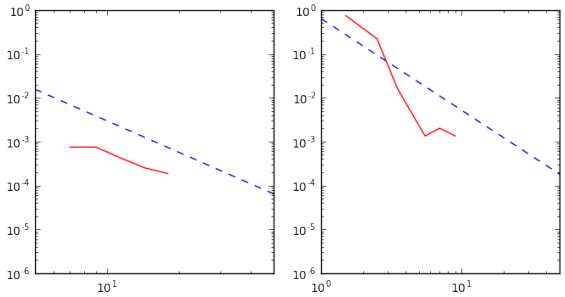

Fig. 3: Fitting a power law for G_1: on the left, the in-degree distribution with $\alpha_{in} = 2.38$, $x^{in}_{min} = 1$, and on the right, the out-degree distribution $\alpha_{out} = 2.08$, $x^{out}_{min} = 1$.

website data breach victims, so that their information was not brought to the attention of close friends and family. Thus, by looking at transactions of 1.05Ƀ for the period around 23 August 2015 when the extortion campaign was carried out, the author of [13] identified a list of 67 Bitcoin wallet addresses which could potentially have received payments for said blackmail. We start our analysis with these 67 addresses, and find that 61 of the addresses are connected to each other in an undirected variant of the address graph, while the other six are totally isolated. In Fig. 2 we show the distribution of pair-wise distances among the 61 addresses. Note that many addresses are still very isolated. Nevertheless, we studied graphs of radius 6 around each of them, to identify subsets of nodes within the specified distance: 39 of the nodes were each isolated from all other nodes in the group by a distance of at least 6. Only 22 of the original 61 nodes were identified to be close enough to each other. We focus the rest of our study around these. Our subsequent investigations are on subgraphs (of radii 4-5) centered around the node closest on an average to the aforementioned group of 61 nodes, namely *1G52wBtL51GwkUdyJNYvMpiXtqaGkTLrMv*. They contained the 22 suspected nodes.

III. Directed Random Graphs

For $G = (V, E)$ a directed graph with $n = |V|$ nodes, and E its set of directed edges, we denote by k_i^{in} the in-degree of vertex i, and by k_i^{out} its out-degree. Since every edge of G must leave some node and enter another one, it is well known (and sometimes referred to as the Handshake Lemma or Theorem) that $\sum_{i=1}^n k_i^{in} = \sum_{i=1}^n k_i^{out} = |E|$.

A. Power-Law Directed Graphs

A power-law distribution is a probability distribution given by $p(x) = Cx^{-\alpha}$, $x \geq x_{min}$ where C is a normalization constant, so that, in the discrete case, $\sum_{x \geq x_{min}} p(x) = C \sum_{x \geq x_{min}} x^{-\alpha} = 1$. For the continuous case, it is easy to see that $\int_{x_{min}}^{\infty} p(x)dx = C \int_{x_{min}}^{\infty} x^{-\alpha} dx = C \frac{x_{min}^{-\alpha+1}}{\alpha - 1}$, assuming that $\alpha > 1$ (otherwise the exponent $-\alpha + 1$ is positive and this quantify is infinite), from which $C = (\alpha - 1) x_{min}^{\alpha - 1}$.

The mth moment of a random variable X distributed according to $p(x)$ is given by $\langle X^m \rangle = \sum_{x \geq x_{min}} x^m p(x)$, or $\langle X^m \rangle = \int_{x_{min}}^{\infty} x^m p(x)dx = C \frac{x_{min}^{-\alpha+1+m}}{\alpha - m - 1}$ assuming that $\alpha > m + 1$, which simplifies to $x_{min}^m \frac{\alpha - 1}{\alpha - m - 1}$. In particular, the mean is given for $m = 1$.

A power-law (or scale-free) directed graph is a graph whose in-degrees $k_1^{in}, \ldots, k_n^{in}$ and out-degrees $k_1^{out}, \ldots, k_n^{out}$ are realizations of the random variables K^{in}, K^{out} which follow the respective power laws: $p_{in}(x) = C_{in} x^{-\alpha_{in}}$, $x \geq x_{min}^{in}$, $p_{out}(x) = C_{out} x^{-\alpha_{out}}$, $x \geq x_{min}^{out}$.

Again, since every graph edge must leave some node and enter another, the average in-degree and out-degree are the same: $\langle K^{in} \rangle = \langle K^{out} \rangle$. By the law of large numbers, $\sum_{i \in V} k_i^{out} = \sum_{i \in V} k_i^{in} = n \langle K^{in} \rangle$ and the probability q_{ij} that nodes i and j are connected by a directed edge (i, j) is

$$q_{ij} = \frac{k_i^{out} k_j^{in}}{\sum_{i \in V} k_i^{in}} = \frac{k_i^{out} k_j^{in}}{n \langle K^{in} \rangle}. \quad (1)$$

Power-law graphs are of interest in the context of the Bitcoin network. In [17], the authors show that over different one month periods after the fall of 2010 (a period that they call the trading phase), the in-degree and out-degree distributions do not change much (it is claimed that the network measures converge to their typical value by mid-2011) and are approximated by power-laws $p_{in}(x) \sim x^{-2.18}$ and $p_{out}(x) \sim x^{-2.06}$.

We thus consider two samples of the Bitcoin network, a (very) small sample $G_1 = (V_1, E_1)$ with $|V_1| = 4571$ and $|E_1| = 4762$, obtained by extracting a subgraph (considered undirected) of radius 4 around the given address of interest (*1G52wBtL51GwkUdyJNYvMpiXtqaGkTLrMv*) as discussed in Subsection II-B, and a small sample $G_2 = (V_2, E_2)$ with $|V_2| = 82663$, $|E_2| = 119190$ obtained by considering a radius 5 instead (this corresponds to the period of Aug 13 to Sep 6 2017), and explore whether they are power-law graphs. We checked the fit of the in-degree and out-degree distributions with respect to power laws for both G_1 and G_2 using [18]. The results are shown in Fig. 3 and 4. The algorithm in [18] looks for either the best fit, or the best fit given a choice of x_{min}. We chose to show the results obtained by fixing x_{min} to be 1, since we are more interested in the overall behavior of the distributions rather than wanting the best fit.

We do observe, as expected, that both graphs exhibit a power-law behavior for both their in-degree and out-degree

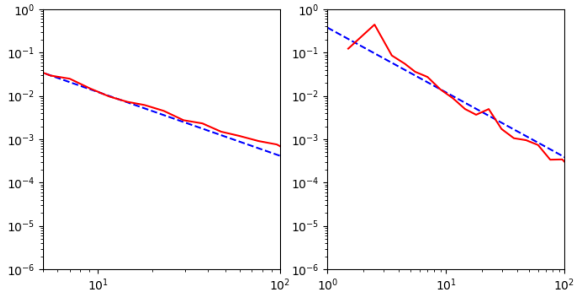

Fig. 4: Fitting a power law for G_2: on the left, the in-degree distribution with $\alpha_{in} = 1.47$, $x^{in}_{min} = 1$, and on the right, the out-degree distribution $\alpha_{out} = 1.493$, $x^{out}_{min} = 1$

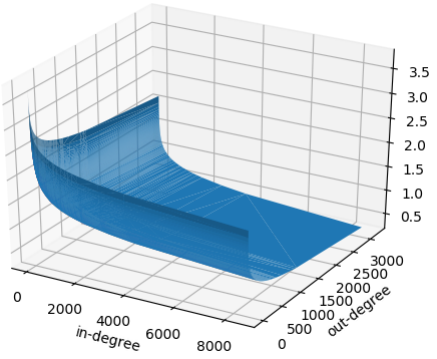

Fig. 5: Average path length for G_2 according to (2).

distributions. The exact values of α_{in} and α_{out} observed on both figures are however purely indicative, we do not draw conclusions compared to the values described in [17] since (1) we would need the value of x_{min} in [17] for a proper comparison, and more importantly (2), the graph sizes are not comparable. In fact, by extracting a subgraph of (much) smaller size, it is not clear that the graph characteristics are preserved, and by cutting the graph at a given radius, the boundary effects should be taken into accounts. The choice of a subgraph of radius r around a chosen address will be justified by the desired analysis on this specific address.

B. Average Path Length

We revisit the average path length analysis of [15] proposed for undirected graphs and extend the analysis for directed graphs. Let $G = (V, E)$ be a directed graph. An arbitrary sequence of vertices from V forms a directed walk on G, where vertices can be repeated, creating possibly directed cycles, and reaching a length which can possibly be arbitrarily large. In comparison, a directed path does not allow repetition of any vertex, thus it cannot contain any cycle. If there is a directed walk from i to j, skipping the cycles will give a path. Let $p_{ij}(l)$ be the probability that there is at least one walk of length l between i and j. Then in terms of statistical ensemble of random graphs [15], the probability $p_{ij}(l)$ of at least one walk of length l between i and j also captures the probability that the nodes i and j are neighbors of order not higher than l. Thus $p^*_{ij}(l) = p_{ij}(l) - p_{ij}(l-1)$ is the probability that i and j are at distance l from each other. Using (1) and Lemma 1 from [15], we have that $p_{ij}(l) \approx 1 - \exp(s_{ij}(l))$ where

$$s_{ij}(l) = -\sum_{v_1 \in V} \cdots \sum_{v_{l-1} \in V} q_{iv_1} q_{v_1 v_2} \cdots q_{v_{l-1} j}$$

since in the directed graph G, at any node of the walk, the next step does not depend on the previous one. Then

$$\begin{aligned} s_{ij}(l) &= -\sum_{v_1 \in V} \cdots \sum_{v_{l-1} \in V} \frac{k^{out}_i k^{in}_{v_1}}{n\langle K^{in}\rangle} \frac{k^{out}_{v_1} k^{in}_{v_2}}{n\langle K^{in}\rangle} \cdots \frac{k^{out}_{v_{l-1}} k^{in}_j}{n\langle K^{in}\rangle} \\ &= -\frac{k^{out}_i k^{in}_j}{n\langle K^{in}\rangle^l} \sum_{v_1 \in V} \cdots \sum_{v_{l-1} \in V} \frac{k^{in}_{v_1} k^{out}_{v_1} k^{in}_{v_2} \cdots k^{in}_{v_{l-1}} k^{out}_{v_{l-1}}}{n^{l-1}} \\ &= -\frac{k^{out}_i k^{in}_j \langle K^{in} K^{out}\rangle}{n\langle K^{in}\rangle^l} \sum_{v_1,\ldots,v_{l-2} \in V} \frac{k^{in}_{v_1} k^{out}_{v_1} \cdots k^{in}_{v_{l-2}} k^{out}_{v_{l-2}}}{n^{l-1}} \\ &= -\frac{k^{out}_i k^{in}_j}{n} \frac{\langle K^{in} K^{out}\rangle^{l-1}}{\langle K^{in}\rangle^l} \end{aligned}$$

and $p^*_{ij}(l) = 1 - \exp(s_{ij}(l)) - 1 + \exp(s_{ij}(l-1))$.

As in [15], averaging $p^*_{ij}(l)$ over all pairs of vertices one obtains the internode distance distribution $\langle\langle p^*_{ij}(l)\rangle_i\rangle_j$.

The expectation value $l_{ij}(k^{out}_i, k^{in}_j)$ for the average path length between i and j is

$$\begin{aligned} &\sum_{l=1}^{\infty} l p^*_{ij}(l) = \sum_{l=1}^{\infty} l(\exp(s_{ij}(l-1)) - \exp(s_{ij}(l))) \\ &= \sum_{l=1}^{\infty} l \exp(s_{ij}(l-1)) - \sum_{l=2}^{\infty} (l-1)\exp(s_{ij}(l-1))) \\ &= \sum_{l=0}^{\infty} \exp(s_{ij}(l)). \end{aligned}$$

This is happening because the graph can be arbitrarily large, and hence, so can be the length of a path. Since $\exp(s_{ij}(l))$ is a function of l which is non-negative, continuous, decreasing and Riemann integrable, we can invoke the following version of the Poisson summation formula [19, p.1]

$$\frac{1}{2} f(0) + \sum_{l=1}^{\infty} f(l) = \int_0^{\infty} f(t)dt + 2\sum_{n=1}^{\infty} \int_0^{\infty} \cos(2\pi n t) f(t) dt$$

with $f(l) = \exp(s_{ij}(l))$. Since $\cos(2\pi n t)$ is integrable and does not change sign over the unit semi-circle, the generalized mean value theorem for integrals yields that $2\sum_{n=1}^{\infty} \int_0^{\infty} \cos(2\pi n t) f(t) dt = 0$. Set

$$k = \frac{k^{out}_i k^{in}_j}{n\langle K^{out} K^{in}\rangle}, \; a = \frac{\langle K^{out} K^{in}\rangle}{\langle K^{in}\rangle}.$$

Then set $t = ka^l$, $dt = ka^l \ln a dl$, so that $\int_0^{\infty} \exp\left(-ka^l\right) dl = \int_k^{\infty} \frac{\exp(-t)}{\ln(a)t} dt$ and $l_{ij}(k^{out}_i, k^{in}_j) = \frac{1}{2}\exp(-k) - \frac{Ei(-k)}{\ln a}$. Furthermore [20, 8.214] $Ei(-k) = \gamma + \ln(k) + \sum_{m=1}^{\infty} \frac{(-k)^m}{m \cdot m!}$, where $\gamma \approx 0.57721$ is Euler's constant. Since $-k \approx 0$ in most of the cases, $\exp(-k) \approx \exp(0)$ and $Ei(-k) \approx \gamma + \ln(k)$ (computations done on the considered sample graphs confirm that the approximations are actually tight), we finally obtain

$$l_{ij}(k^{out}_i, k^{in}_j) \approx \frac{1}{2} - \frac{\gamma + \ln(\frac{k^{out}_i k^{in}_j}{n\langle K^{out} K^{in}\rangle})}{\ln(\frac{\langle K^{out} K^{in}\rangle}{\langle K^{in}\rangle})}. \qquad (2)$$

(k^{out}, k^{in})	(1,1)	(5,7)	(2807,196)	(8530,3196)
G_1	3.524	2.612	0.1335	NA
G_2	3.802	3.065	NA	0.254

TABLE II: Maximum and minimum average path lengths approximated by (2) for G_1 and G_2.

IV. Bitcoin Extortion Analysis

A. Average Length Analysis

We consider two samples of the Bitcoin network: the graphs $G_1 = (V_1, E_1)$ with $|V_1| = 4571$ and $|E_1| = 4762$, and $G_2 = (V_2, E_2)$ with $|V_2| = 82663$, $|E_2| = 119190$ obtained by extracting a subgraph (considered undirected) of respectively radius 4 and 5, around a given address of interest (*1G52wBtL51GwkUdyJNYvMpiXtqaGkTLrMv*). We limit the graph radius to study to 5, to keep the analysis tractable, particularly for manual computations. The graph $G_3 = (V_3, E_3)$ with radius 6 has $|V_3| = 1374498$ and $|E_3| = 4722501$.

Formula (2) can be used to evaluate the average path length in these graphs. In the case where in-degree/out-degree distributions are known to be power laws, closed form expressions for $\langle K^{in} \rangle$ can be used, assuming that $\alpha > 2$. However, since the derivation of (2) does not rely on the distributions to be power laws, we evaluate $\langle K^{in} \rangle$ and $\langle K^{out} K^{in} \rangle$ numerically (alternatively, since we know from the previous section that our graphs have distributions that are close to a power law, we could vary the choice of x_{min} to get the best fit in terms of power law, but still the closed-form formulas for the mean of a power law is available only for specific exponents α). To get a sense of (2), we provide some representative values in Table II: paths from nodes with out-degree 1 to nodes with in-degree 1 have approximated average lengths of 4, which appears meaningful, considering that the studied subgraphs have radii respectively 4 and 5, and that the nodes with these degrees are typically on the boundary (they are an artefact of cutting the edges at a chosen radius). The values for $(1, 1)$ are the maximum values found for G_1 and G_2. The minimum values are given for $(2807, 196)$ and $(8530, 3196)$ respectively: these are nodes with (very) high degrees, they are thus expected to be neighbors. Fig. 5 shows the average lengths determined using (2) for all the in-degree/out-degree combinations in G_2.

Since our analysis starts from a given address of interest (*1G52wBtL51GwkUdyJNYvMpiXtqaGkTLrMv* for the illustration), we look for anomalies in terms of path length around this node in the graph G_2. The most prominent pairs of nodes with paths shorter than expected are given in Table III. The first and the third columns give respectively the beginning and end of the path, with their in/out-degree. Since l_{ij} is an approximation based on (2), for sanity check, we compute the actual average path length between nodes of given degrees, and observe that l_{ij} is in fact a lower bound on the actual average length.

B. Confluence Analysis

We next consider another indicator, where instead of a length l shorter than average, we take into account the normalized length, which we define as

$$\tau(l) = \frac{l}{\text{no. of shortest paths between the node pair}} \quad (3)$$

where l is the length of the shortest path between two nodes. This can be used as a metric of how closely knit two nodes are. In fact, the idea could also be generalized to count paths capped within a distance, rather than using only shortest paths. But, as a first approach, we focus on the case where we use shortest paths only. Table IV lists the most prominently close knit nodes found through this heuristic.

Comparing the entrees in Tables III and IV, we identifed wallet addresses *1Nr2y7XD8c27tMQF5XrQJdfUWjCZ3zY5uy* and *1JujBBkRGAEm7JvdnCDfGw939cEtbuuWa2* to be repeated (among other addresses) across the tables. This piqued our interest to study these addresses, and we use them to showcase our methodology of using flow confluences to study relationship among wallet addresses, where we consider more generally short paths but not necessarily the shortest ones (and paths of different lengths could be involved in the confluence).

A key intuition we pursue is that the money flow may be confined and circulate among certain addresses, indicating in turn coordination among them. We thus design an algorithm that systematically finds path confluences over the directed address graph, as described next. Let $G = (V, E)$ represent a directed graph formed by Bitcoin addresses as vertices in V, and directed edges in E representing flow of Bitcoins from one address to another. We use the notation $\overleftrightarrow{G} = (V, \overleftrightarrow{E})$ to refer to an undirected graph obtained from G, such that $\forall (u, v) \in E$, $(u, v) \in \overleftrightarrow{E}$ and $(v, u) \in \overleftrightarrow{E}$. Likewise, we use the notation $\overleftarrow{G} = (V, \overleftarrow{E})$ to denote the directed graph obtained by reversing all the edge directions from G, i.e, $(v, u) \in \overleftarrow{E} \iff (u, v) \in E$.

Consider a shortest path d_{v_1,v_2} between two query vertices v_1 and v_2 in the undirected graph $\overleftrightarrow{G}$ (we abuse the notation to indicate both the path, which is an ordered list of nodes, and the unordered set). The query vertices are nodes of initial interest for whichsoever reasons. If d_{v_1,v_2} is not a path in G, it means that the directionality flips along d_{v_1,v_2}. Starting at each such flipping points (only v_z in the example graph shown in Fig. 6), we run a variation of the breadth first search (BFS) algorithm (starting independently at each flipping point), which returns all the nodes that are revisited: that would be v_5, v_9 and v_6 (because of the self-loop) in our example. Note that the BFS algorithm variation is run over G or $\overleftarrow{G}$, depending on whether the flipping point had outgoing or incoming edges respectively (the latter in our example). This helps us identify (a chain of) path confluences. For instance, from v_9 to v_5, and again from v_5 to v_z, with the second group of path confluences including our original nodes of interest v_1 and v_2. Note that v_6 would seem to have two distinct paths to v_z, one through v_1 and another through v_2. However they are not independent, since both share a single edge v_6, v_5 (unlike v_9 which had multiple paths to v_5). We eliminate from our consideration all the paths that involve any *cut edge*. An implementation of this algorithm has been added to the

in-address	k^{in}	out-address	k^{out}	length	l_{ij}	actual average
1EuRpZCEoyRKWcEpv6jD42N2CxJWrrETcD	2	15hWpb3m5VXdn9KVsS4rSMnrQQJLhXVyN4	16	2	3.084	5.403
1HZErPx5XPrp8mH98rKRjyazQLYN7j34Lk	2	15hWpb3m5VXdn9KVsS4rSMnrQQJLhXVyN4	16	2	3.084	5.403
17CcixA4fyEmscXPsZRshecXzJ52eoKNNY	2	15hWpb3m5VXdn9KVsS4rSMnrQQJLhXVyN4	16	2	3.084	5.403
1PUt7bm1ZU6BRT85yDJxd651Fgneveuzs3	2	15hWpb3m5VXdn9KVsS4rSMnrQQJLhXVyN4	16	2	3.084	5.403
14gkHySfCnoV4Fy4nCxoLUgavMP3DMXKsS	2	15hWpb3m5VXdn9KVsS4rSMnrQQJLhXVyN4	16	2	3.084	5.403
137ENC6fNM97tawPMAGgyn9BKLEmkTx7Xd	2	1J97MX1kCHMgePTBwB97KaHyjSYnkTyt7V	14	2	3.111	5.602
12bbs3MZAsd9mseXKTA93PtHVd2nSkvjDy	2	15WJMyfHeiD3rT8nvvrnyQs2THA3J3wxnF	11	2	3.161	5.328
1Hmjw6JcNDxCQuVo7pkUmRmJzRJe4yLze9	2	15WJMyfHeiD3rT8nvvrnyQs2THA3J3wxnF	11	2	3.161	5.328
14YeaK34GE68S6YASdtHR1iThj8c8hBQ1C	2	**1JujBBkRGAEm7JvdnCDfGw939cEtbuuWa2**	9	2	3.203	5.490
1MAwvr21q95UBu6y4WhRM47K5RBijQ164G	2	**1JujBBkRGAEm7JvdnCDfGw939cEtbuuWa2**	9	2	3.203	5.346
1Kbj3tzAbtHgJHt6h6G5PuvWYRZkWXAfhH	2	**1JujBBkRGAEm7JvdnCDfGw939cEtbuuWa2**	9	2	3.203	5.346
1KFc1tekeQtZg48pKbCEU2j7J14N2XFTNW	2	**1JujBBkRGAEm7JvdnCDfGw939cEtbuuWa2**	9	2	3.203	5.346
13Mj4nmV127symEJ6GTTb4kdTWyQ884QoN	2	**1JujBBkRGAEm7JvdnCDfGw939cEtbuuWa2**	9	2	3.203	5.346
1ETmDWjyto3gGegqch41PCeYJiVx9pmsuv	2	**1JujBBkRGAEm7JvdnCDfGw939cEtbuuWa2**	9	2	3.203	5.346
1ErwF3T6QCdZ75PBXZdyxXJPN2k7bBV9fd	2	1HZErPx5XPrp8mH98rKRjyazQLYN7j34Lk	2	2	3.514	5.566
1G52wBtL51GwkUdyJNYvMpiXtqaGkTLrMv	2	17CcixA4fyEmscXPsZRshecXzJ52eoKNNY	2	2	3.514	5.566
18euqRRpC2Zp9i9dwrT7Qp3M8jfbu9TUn6	2	14gkHySfCnoV4Fy4nCxoLUgavMP3DMXKsS	2	2	3.514	5.566
1Q3AvrmZ8cykHGQ7kCkSZaopnWiHPzc1Qw	2	14gkHySfCnoV4Fy4nCxoLUgavMP3DMXKsS	2	2	3.514	5.566
1Nr2y7XD8c27tMQF5XrQJdfUWjCZ3zY5uy	2	1PUt7bm1ZU6BRT85yDJxd651Fgneveuzs3	2	2	3.514	5.566
1BkUNiTfJRTTG1iycaWGZavmLnLVHorUUo	2	14YeaK34GE68S6YASdtHR1iThj8c8hBQ1C	2	2	3.514	5.566

TABLE III: Paths of length shorter than average, with their in- and out-addresses and respective in-degree k^{in} and out-degree k^{out}. The column *length* is the actual path length, l_{ij} is the value of (2), the actual average on G_2 is last.

in-address	k^{in}	out-address	k^{out}	length l	$\tau(l)$	l_{ij}
1ErwF3T6QCdZ75PBXZdyxXJPN2k7bBV9fd	2	15WJMyfHeiD3rT8nvvrnyQs2THA3J3wxnF	11	4	1.333	3.161
1ErwF3T6QCdZ75PBXZdyxXJPN2k7bBV9fd	2	**1JujBBkRGAEm7JvdnCDfGw939cEtbuuWa2**	9	4	0.8	3.203
1JBygewRQuHuh4qJnpQtb1qCfwNJ2zNrg5	2	15WJMyfHeiD3rT8nvvrnyQs2THA3J3wxnF	11	4	2.0	3.161
1JBygewRQuHuh4qJnpQtb1qCfwNJ2zNrg5	2	**1JujBBkRGAEm7JvdnCDfGw939cEtbuuWa2**	9	4	0.8	3.203
1G52wBtL51GwkUdyJNYvMpiXtqaGkTLrMv	2	15WJMyfHeiD3rT8nvvrnyQs2THA3J3wxnF	11	4	1.333	3.161
1G52wBtL51GwkUdyJNYvMpiXtqaGkTLrMv	2	**1JujBBkRGAEm7JvdnCDfGw939cEtbuuWa2**	9	4	0.8	3.203
18euqRRpC2Zp9i9dwrT7Qp3M8jfbu9TUn6	2	15WJMyfHeiD3rT8nvvrnyQs2THA3J3wxnF	11	4	1.333	3.161
18euqRRpC2Zp9i9dwrT7Qp3M8jfbu9TUn6	2	**1JujBBkRGAEm7JvdnCDfGw939cEtbuuWa2**	9	4	0.8	3.203
1Q3AvrmZ8cykHGQ7kCkSZaopnWiHPzc1Qw	2	15WJMyfHeiD3rT8nvvrnyQs2THA3J3wxnF	11	4	1.333	3.161
1Q3AvrmZ8cykHGQ7kCkSZaopnWiHPzc1Qw	2	**1JujBBkRGAEm7JvdnCDfGw939cEtbuuWa2**	9	4	0.8	3.203
1Nr2y7XD8c27tMQF5XrQJdfUWjCZ3zY5uy	2	15WJMyfHeiD3rT8nvvrnyQs2THA3J3wxnF	11	4	1.333	3.161
1Nr2y7XD8c27tMQF5XrQJdfUWjCZ3zY5uy	2	**1JujBBkRGAEm7JvdnCDfGw939cEtbuuWa2**	9	4	0.8	3.203
1BkUNiTfJRTTG1iycaWGZavmLnLVHorUUo	2	15WJMyfHeiD3rT8nvvrnyQs2THA3J3wxnF	11	4	1.333	3.161

TABLE IV: Paths of low normalized length $\tau(l)$, defined in (3), described by their in- and out- addresses.

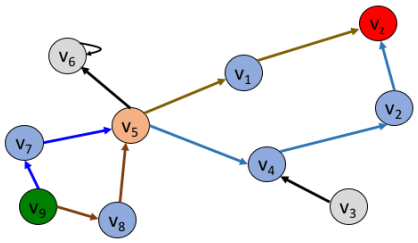

Fig. 6: Example graph: A chain of path confluences between v_9, v_5 and v_z, discovered using v_1 and v_2 as inputs.

demo tool BiVA [21], and the code (and data) is available at: https://github.com/feog/Biva.

We applied the above path confluence algorithm among pairs of nodes from the shortlist of 22 nodes (described in SectionII-B), to determine if they in-fact interact with each other, even if indirectly, but in a manner which suggests coordination among their activities. Based on the confluence algorithm, for this specific set of addresses, we indeed observe that the money is circulated among them, and the addresses are in-fact close knit (more precisely, 102 paths of confluence were found). The confluence algorithm also helps us identify 23 further addresses that the original suspect addresses seem to be liaising with. This thus yields a new group of total 45 suspect addresses. We note that the existence of such a tightly connected group is not in itself a definitive proof that this group of addresses was indeed the one used for collecting the extortion amount: it is only indicative of the fact that this group of addresses - the originally suspected ones, as well as the other addresses they are liaising with, are controlled by the same entity (it could be a single person or a group).

In Fig. 7 we show a superimposition of several chains of path confluences (in blue) involving the wallet address *1Nr2y7XD8c27tMQF5XrQJdfUWjCZ3zY5uy* indicated by the last letters *uy*. This is one of the 2 addresses singled out from Tables III and IV. This address is one of the 20 in-addresses from Table III, and 9 other in-addresses from this table appear as part of these confluences (*cD*, *s3*, *Xd*, *Dy*, *e9*, *4G*, *hH*, *NW* and *oN*). Also 3 out-addresses from the same table appear (*N4*, *nF* and *s3*). Also *s3* appears twice, and while *uy* was an original suspect address, the path confluence around it contains 8 of the 23 new suspicious addresses (not encircled in green).

The superimposition of path confluences involving the other outstanding address *1JujBBkRGAEm7JvdnCDfGw939cEtbuuWa2* labelled by the last letters *a2* is shown in Fig. 8. We observe that this new path confluence also includes *cH*, *dV*, but further adds *mN*, *tu* and *Sw* to the list of new addresses to look at, thus between

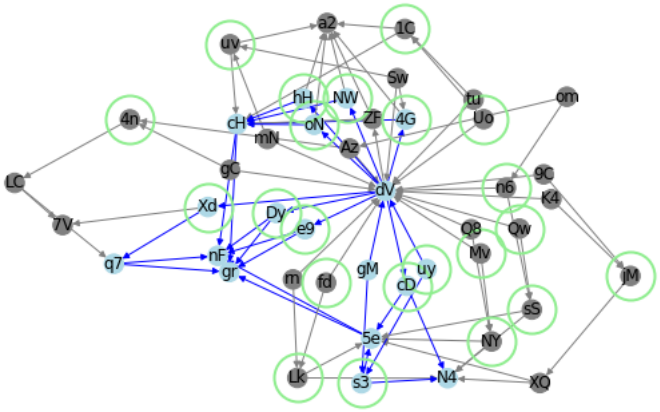

Fig. 7: A superimposition of path confluences involving the original 22 suspect addresses (encircled in green), and the 23 new addresses identified in the process: we also highlight (in blue) the confluences involving the wallet address *1Nr2y7XD8c27tMQF5XrQJdfUWjCZ3zY5uy* indicated as *uy*.

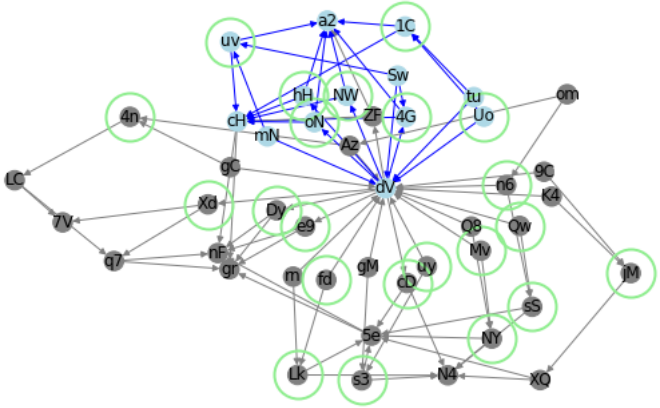

Fig. 8: Same as Fig. 7: we highlight the confluences involving the wallet address *1JujBBkRGAEm7JvdnCDfGw939cEtbuuWa2* indicated as *a2*.

the two of them, *1Nr2y7XD8c27tMQF5XrQJdfUWjCZ3zY5uy* and *1JujBBkRGAEm7JvdnCDfGw939cEtbuuWa2* implicate about half of the 23 new addresses.

Given that none of the other originally suspected addresses seem to have such behavior patterns and appear to be operating in isolation, while the group of 22 addresses and other associated 23 addresses as identified using the path confluence algorithm are operating in a manner which resemble laundering activity, we assume (with the above caveat) these 45 (22+23) addresses to be involved in collecting and laundering the extortion money. When we study the transactions in which the 45 addresses are involved in, and eliminate 1.05Ƀ transactions among this group of addresses, we identify 18 payments that are made by other unrelated wallet addresses. This as an indicator of the amount (18.9Ƀ) of money collected by the specific extortion campaign (under the above assumption).

C. Estimation of the Total Money Controlled

Having estimated the amount of money the extortionists received based on the particular amount of 1.05Ƀ per blackmail victim, we wanted to determine how much money they might be handling during the same time period - be it money they may have received under the same blackmail campaign but by receiving other amounts, or money they received and handled for other reasons, but using the same set of 45 addresses we have previously identified. To that end, we need to study the transaction network, which we describe next, before continuing our investigation. We note that the same extortionists may be handling other groups of addresses at the same time, and the estimate here is limited to only the amount handled by the specific cluster of 45 addresses we identified, and for the specific period of time we studied. However, following the money flow over a longer period of time, using the methodology discussed in this paper, one can discover more addresses, and repeat the exercise.

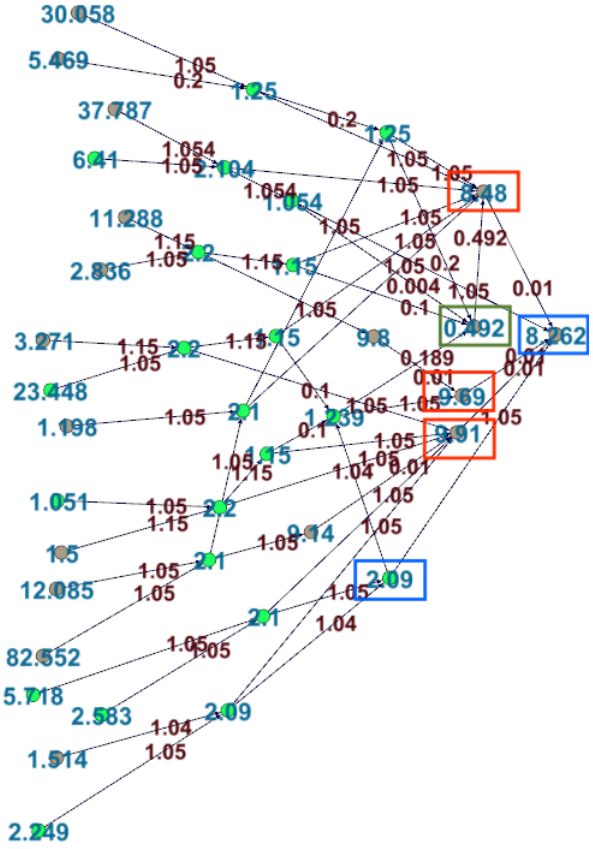

Fig. 9: Transaction network: Only edges involving the 45 suspected Bitcoin wallet addresses are shown.

A transaction network, as used e.g. in [9], represents the flow of Bitcoins between transactions over time. Each vertex represents a transaction, and each directed edge between a source vertex to a target vertex connects an output transaction of the source, to the target where it is used as input, as illustrated on Fig. 10: we have 6 transactions thus 6 nodes. The first transaction has no input, but one output, transaction 2 has one input and two outputs, each of them are inputs to respectively transactions 3 and 4.

In Fig. 9, we show a subset of the transaction graph for the period of 2015-08-24 08:07:13 to 2015-08-27 00:18:11 GMT.

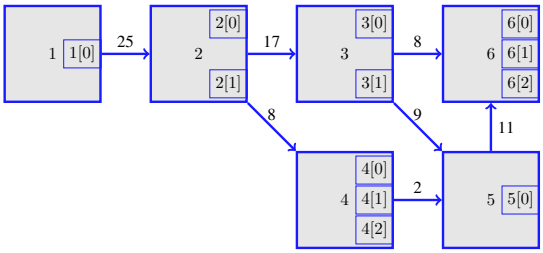

Fig. 10: The transaction network corresponding to Table I.

We only show the transaction edges that involve at least one of the 45 suspected addresses. The transactions which contain at least one of the original 22 suspected addresses are shown in green, and the others are shown in gray. The total amounts of the transactions, and the specific amount of Bitcoins which are flowing from one transaction are indicated over the nodes and edges respectively (rounded to three digits after decimal). The transactions highlighted by the red rectangles are sinks where the suspected addresses are aggregating their funds. Summing them, we obtain an initial estimate of 36.56฿ as the sum of money the suspect addresses are handling. Note that, we overlook the amount in the transactions encompassed in green rectangle to avoid double counting, because the money flows further to the node on top. However, we also see that our initial estimate missed 2.09฿ (including which, the volume of money the 45 suspect nodes were handling during the time period of interest can be estimated as 38.65฿) from the bottom transaction highlighted by a blue rectangle, because we did not consider it as a sink. Yet, the rightmost transaction also encompassed in blue includes this amount but also partial amounts from other nodes we considered as sinks. As mentioned above, the study can be carried out for a larger time-window to get a better estimate of the overall money being handled by the extortionists, by identifying further addresses that can be aggregated with the initial group of 45 addresses, in a manner similar to how we discovered 23 new addresses based on the initial shortlist of 22 suspects.

V. Concluding remarks

Motivated by the escalation of extortions using cryptocurrencies to receive and launder the money, this paper looks at possible ways to extend heuristics for detecting/aggregating perpetrators' wallet addresses for the Bitcoin network. The two main ingredients of the proposed methodology are (1) detecting outliers with respect to two measures, the expected and the normalized expected path lengths between two nodes, and (2) proposing an algorithm to track flow confluences between node addresses of interest. The 2015 Ashley Madison extortion scam has been analyzed as our case study. We further evaluated the amount of money involved both in the scam itself, but also estimated a lower bound on the amount of money the suspicious addresses were controlling altogether during the time window for which we conducted our study.

An obvious line of future work to be investigated is the use of the same methodology but for other scams, such as the WannaCry one. Once it becomes known that simple heuristics such as detecting a particular amount of money (1.05฿in this scam) can leak information, perpetrators are quick to update their scams, and in fact, even for the Ashley Madison scam, subsequent amounts of Bitcoin requested became more random. Also, while mixing services had already appeared long before August 2015, this group of perpetrators did not deem necessary to cover their tracks through mixing services, a scenario that may not hold true anymore for more recent scams. We expect our flow path confluence based approach to be robust even in the presence of mixing. Nevertheless, this is yet to be validated by studying if and upto what extent, accuracy of the path confluence based address aggregation mechanism deteriorates in the presence of mixing, and accordingly refine the technique if necessary. Finally, abstracting out the principles and adapting them to analyze other cryptocurrencies is another interesting direction.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system", 2008.

[2] M. Spagnuolo, M. Federico, Z. Stefano, "Bitiodine: Extracting intelligence from the bitcoin network", in "Intl. Conf. on Financial Cryptography & Data Security" 2014.

[3] M. Lischke and B. Fabian, "Analyzing the Bitcoin Network: The First Four Years", in "Future Internet", 8(7) 2016.

[4] M. Harrigan, C. Fretter, "The Unreasonable Effectiveness of Address Clustering", in "arXiv", 2016.

[5] K. Liao, Z. Zhao, A. Doupe, G-J. Ahn, "Behind closed doors: measurement and analysis of CryptoLocker ransoms in Bitcoin", in "APWG Symposium on Electronic Crime Research (eCrime)", 2016.

[6] H. Kuzuno, C. Karam, "Blockchain Explorer: An Analytical Process and Investigation Environment for Bitcoin", in "APWG Symposium on Electronic Crime Research (eCrime)", 2017.

[7] E. Androulaki, et al. "Evaluating user privacy in bitcoin", in "Intl. Conf. on Financial Cryptography and Data Security" 2013.

[8] J.D. Nick, "Data-Driven De-Anonymization in Bitcoin", in "ETH Master Thesis", 2015.

[9] F. Reid, M. Harrigan, "An Analysis of Anonymity in the Bitcoin System", *Altshuler Y., Elovici Y., Cremers A., Aharony N., Pentland A. (eds), Security and Privacy in Social Networks. Springer*, New York, 2013. https://arxiv.org/pdf/1107.4524.pdf

[10] M. Moser, R. Bohme, "Anonymous Alone? Measuring Bitcoin's Second-Generation Anonymization Techniques", in "IEEE European Symposium on Security and Privacy Workshops", 2017.

[11] D. Ermilov, M. Panov, Y. Yanovich, "Automatic Bitcoin Address Clustering", in "IEEE Intl. Conf. on Machine Learning and Applications", *ICMLA* 2017.

[12] https://blog.wizsec.jp/2017/07/breaking-open-mtgox-1.html

[13] https://blog.cloudmark.com/2015/09/01/does-blackmailing-pay-signs-in-bitcoin-blockchain-of-responses-to-ashley-madison-extortion-emails/, accessed on 14 July 2018.

[14] B. Zheng et al., "Malicious Bitcoin Transaction Tracing Using Incidence Relation Clustering", in "Mobile Networks and Management", (MONAMI) 2017.

[15] A. Fronczak, P. Fronczak, J. A. Holyst, "Average Path Length in Random Works", Physical Review E 70.5 (2004): 056110.

[16] A. Narayanan, J. Bonneau, E. Felten, A. Miller, S. Goldfeder, "Bitcoin and Cryptocurrency Technologies", *Princeton University Press*, 2016.

[17] Kondor D., Pósfai M., Csabai I., Vattay G. (2014), "Do the Rich Get Richer? An Empirical Analysis of the Bitcoin Transaction Network". PLoS ONE 9(2): e86197.

[18] J. Alstott, E. Bullmore, E. Plenz, "powerlaw: A Python Package for Analysis of Heavy-Tailed Distributions", PLoS ONE 9(1): e85777, 2014.

[19] A. P. Guinand, "On Poisson's Summation Formula", *Annals of Math.*, vol. 42, no. 3, July 1941.

[20] I.S. Gradshteyn and I.M. Ryzhik, "Table of Integrals, Series, and Products", 7th edition, Elsevier, Academic Press.

[21] F. Oggier, S. Phetsouvanh, A. Datta, "BiVA: Bitcoin Network Visualization & Analysis", Demo paper, *ICDM* 2018.