Automatic Bitcoin Address Clustering

Dmitry Ermilov[*,†], Maxim Panov[†,‡], Yury Yanovich[*,‡]
[*]Bitfury
[†]Skoltech
[‡]Institute for Information Transmission Problems

***Abstract*—Bitcoin is digital assets infrastructure powering the first worldwide decentralized cryptocurrency of the same name. All history of Bitcoins owning and transferring (addresses and transactions) is available as a public ledger called blockchain. But real-world owners of addresses are not known in general. That's why Bitcoin is called pseudo-anonymous. However, some addresses can be grouped by their ownership using behavior patterns and publicly available information from off-chain sources.**

Blockchain-based common behavior pattern analysis (common spending and one-time change heuristics) is widely used for Bitcoin clustering as votes for addresses association, while off-chain information (tags) is mostly used to verify results. In this paper, we propose to use off-chain information as votes for address separation and to consider it together with blockchain information during the clustering model construction step. Both blockchain and off-chain information are not reliable, and our approach aims to filter out errors in input data.

The results of the study show the feasibility of a proposed approached for Bitcoin address clustering. It can be useful for the users to avoid insecure Bitcoin usage patterns and for the investigators to conduct a more advanced de-anonymizing analysis.

***Index Terms*—Bitcoin, blockchain, clustering, privacy, anonymity.**

I. Introduction

Bitcoin is the widest spread and the most secure blockchain, it is a distributed ledger, digital assets infrastructure, crypto blockchain messaging system, and as well it may work as a cryptocurrency. Exactly cryptocurrency aspect is considered in this paper. Bitcoin is supported by the eponymous decentralized network [1]. The medium of exchange for it is a payment unit with the same name. Cryptocurrancy is transferred using transactions that utilize common cryptographic primitives (such as digital signatures and hash functions) to provide authentication capabilities. All transactions are included in a publicly available distributed ledger (the blockchain) after being verified by the nodes in the Bitcoin network. Currently, Bitcoin is the most widespread and the most secure blockchain serving as digital assets infrastructure and secure blockchain messaging system.

One of the important properties of Bitcoin is its pseudonymity [2]. Participants in the Bitcoin network are not obliged to disclose ownership of bitcoins. Thus, data about the owners of bitcoins is generally not available. However, given the transaction history and data that Bitcoin users have disclosed about themselves, it is possible in some cases to recover information about particular bitcoins. The goal of this paper is to provide a Bitcoin clustering algorithm such that all the addresses in each cluster are controlled by the same user.

The research in both anonymization and de-anonymization in the Bitcoin system is actively developing in recent years and both blockchain (transactions analysis) and off-chain information (analyzing other public information from the Internet) are used. The overviews of anonymization and de-anonymization techniques are provided in [3], [2], [4], [5]. Possible anonymization approaches are: shared coin and send mixers [6], [7], [8], [9], transaction remote release [10] to hide the IP address of the transaction's author, ZeroCash [11] for advanced coin mixing. Different approaches for de-anonymization and analysis are: Sybil Attack [12] and Fake Node Attack [13] to get users' IP, addresses clustering [14], [15], [16] mostly based on two heuristics (common spending and one-time change). In all these approaches tags are used for results verification. There are also papers with behavior-based Bitcoin analysis [17], [18] and the ones where tags are used for address classification [19] or risk scoring [20], [21], [22], [23].

We consider de-anonymization of Bitcoin addresses as a clustering problem. Clustering is an important class of unsupervised learning problems [24], which focuses on splitting data into groups and has a variety of approaches to its solution [25], [26]. However, Bitcoin address clustering is fairly different from classical clustering problems as there is no direct information about the objects' (addresses) such as coordinates or distances. The other peculiarity of the problem is the vastness of the Bitcoin blockchain, which requires designing computationally efficient algorithms for its' clustering. In this paper, we introduce the efficient automatic clustering approach based on both the blockchain and the off-chain information.

II. Bitcoin blockchain information

The Bitcoin blockchain provides information about all bitcoins in use. This information, which presents the global state of the Bitcoin network, is a set of records (unspent transaction outputs, or UTXOs in Bitcoin terminology). Each UTXO specifies the associated value in bitcoins and conditions under which these bitcoins can be spent. In this work, we do not consider such modern (actual since 1^{st} August 2017) advanced features of Bitcoin transactions as BIP141 [27] and BIP091 [28], for simplicity.

Transfer of bitcoins is manifested through transactions. Each transaction contains:

- One or more inputs, each referring to a valid UTXO and containing authentication information, allowing one to spend corresponding bitcoins.
- One or more outputs, each containing a specification of a newly created UTXO.

After a completion of a transaction, UTXOs referenced by its inputs are removed from the UTXO set maintained by every Bitcoin node, and its outputs are added into the UTXO set. To incentivize inclusion of transactions into the Bitcoin blockchain, the cumulative value of transaction outputs is usually slightly less than the cumulative value of its inputs. The difference (transaction fee) is paid to the Bitcoin node that included the transaction into the blockchain, see [1] and [29], Chapter 8.

In order to spend bitcoins, the owner generally needs to present to the network publicly verifiable authentication information proving that the assets belong to him. For convenience, conditions of spending a UTXO can be compressed with a collision-resistant hash function, thus forming a (Bitcoin) address. There are two main kinds of addresses used in Bitcoin today, corresponding to two kinds of conditions under which a UTXO may be spent:

- Knowledge of a single private key in the secp256k1 elliptic curve cryptosystem [30] may be required to spend a UTXO. In this case, the address is the hash of the corresponding public key
- Otherwise, spending a UTXO may require providing proofs of knowledge of m out of n private keys ($1 \leq m \leq n$) in the same cryptosystem [31]. In this case, the address is a hash digest depending on m, n, and n public keys.

There may be multiple UTXOs associated with the same address. We will assume that:

- Each Bitcoin address is controlled by a single real-world entity. Thus, we will ignore those sufficiently rare cases in which a multi-signature address is used for joint ownership of bitcoins, and not for multi-factor authentication [32].
- A single entity may control more than one address.

III. Blockchain-based heuristics

In this section, we present blockchain-based heuristics which are helpful for linking groups of Bitcoin addresses into one cluster. These heuristics, *common spending* and *one time change*, are based on certain patterns which are common for many transactions in the Bitcoin network. However, these patterns are not necessarily satisfied for all the transactions and thus, our heuristics are prone to error. The errors mean that some addresses can be falsely linked together. This may result in a creation of very large clusters, which in fact are not controlled by a single user.

The following definitions will be useful for our analysis.

Definition 1: For the purpose of transaction analysis, a (Bitcoin) transaction is viewed as an ordered triplet $t = (\mathcal{A}, \mathcal{B}, c)$, consisting of:

- The finite multiset of transaction inputs $\mathcal{A}$, where each input $(a_i, A_i) \in \mathcal{A}$ is an ordered pair of the address A_i and the value of the input $a_i > 0$.
- The finite multiset of transaction outputs $\mathcal{B}$, where each output $(b_j, B_j) \in \mathcal{B}$ is an ordered pair of the address B_j and the value of the output $b_j \geq 0$.
- The transaction fee $c = \sum\limits_{(a_i,\cdot)\in\mathcal{A}} a_i - \sum\limits_{(b_j,\cdot)\in\mathcal{B}} b_j \geq 0$.

Definition 2: For an arbitrary multiset of transaction inputs or outputs $\mathcal{A}$ we denote the multiset of addresses in $\mathcal{A}$ as $Addr(\mathcal{A})$.

We start by describing the exact variants of heuristics we used and their differences from ones introduced in [17], [33].

A. Common spending (CS)

The most obvious idea for clustering Bitcoin addresses is linking together all the input addresses of one transaction, which was explored in many papers, see for example [17], [33].

Heuristic 1: If two or more addresses are inputs of the same transaction with one output, then all these addresses are controlled by the same user.

This heuristic is expected to be very accurate as common spending by different users should be based on high degree of trust between them. However, as opposed to the situation in 2013 [33], nowadays there exists multiple services for multi input transactions.

Importantly, we attended to transactions with one output, and likewise [33] do not consider multi-output transactions. The reason is that many multi-output transactions are so-called 'shared sends' [6], [9], which are the transactions induced by several users to obfuscate transaction history.

B. One-time change (OTC)

Our second heuristic is more involved. It is based on the standard Bitcoin mechanism where the change from the transaction is returned to a new address.

Definition 3: We say that the transaction $t = (\mathcal{A}, \mathcal{B}, c)$ satisfies the condition of a one-time change if the following conditions hold.

1) $\#Addr(\mathcal{B}) = 2$, i.e. the transaction t has exactly two outputs.
2) $\#Addr(\mathcal{A}) \neq 2$, i.e. the number of t inputs is not equal to two. If $\#Addr(\mathcal{A}) = \#Addr(\mathcal{B}) = 2$ the transaction is most likely shared send mixer.
3) Both outputs of transaction t, B_1 and B_2, are not self-change addresses, i.e. $B_1, B_2 \notin Addr(\mathcal{A})$.
4) One output of the transaction B_1 did not exist before transaction t and decimal representation of the value b_1 has more than 4 digits after the dot.
5) The other output of the transaction B_2 was previously part of the Bitcoin network and has not been OTC addressed in previous transactions.

Heuristic 2: If the transaction satisfies the conditions of a one-time change transaction, (see Definition 3), then the OTC

output and all the inputs of the transaction are controlled by the same user.

We note that this heuristic is much more prone to errors. Especially vulnerable is the condition on some digits in the decimal representation of change. However, our definition of OTC is more stricter than, for example, definition of [33], which results in a less number of false positive OTC transactions.

IV. Off-Chain information for clustering

The previous section made a deal with intrinsic blockchain information. Despite the fact that address owners are not required to disclose information about their selfs, much public information can be found on the Internet (off-chain information). If the Bitcoin address is mentioned in the same data frame with the tag (key phrase-entity, for example, company name or username), then it is said say that the address has such a tag. This section is devoted to off-chain information collection, types of tags and their relation to the clustering.

A. Tag Collection

Tags could be collected either passively or actively. The passive approach means web crawling of public forums and user profiles (for example, *Bitcointalk.com*, *Twitter* and *Reddit*) and Darknet markets (for example, *Silkroad*, *The Hub Marketplace* and *Alphabay*). The active approach means manual analysis of Bitcoin companies and data actualization procedures. The most common Bitcoin businesses companies are exchanges, marketplaces, mining pools and mixers. Some companies mostly use addresses with specific prefixes. As an address is a public key, for an unknown private key then to generate a specific address, one has to try many private keys, i.e., make some extra computational work. For example, *Satoshi Bones* casino uses *1change* and *1bones* prefixes and *BTC-E* exchange uses *1eEUR* and *1eUSD* prefixes. Addresses starting from *1MartinHafernikorn* and *1Ninja* are also computationally demanding and can help to identify users.

We call the collected tags *dirty* as they are not standardized: they are mostly informationless suffixes (for example, *.com, .co, @gmail*), upper and lower letter cases are mostly unnecessary, misprints are included. The *dirty* tags are processed to eliminate described drawbacks and in the process becoming *clean*.

B. Negative pairs

We distinguish six categories of Bitcoin organizations: mining pools (pools), exchanges, Darknet markets (dnm), mixers, gambling and other services (services). The dictionary of *clean* tag types is prepared: each of *clean* tags may correspond to only one type (if any). An address can have tags from different categories. It is assumed unlikely that any cluster has different tags of the same type (it means, for example, that different people control exchanges *Bitfinex* and *HitBTC*). Some pairs of categories are also unlikely to be present in one cluster (for example, exchange and dark market).

Let us call $L = \{\{a_i, a_j\}\}$ the set of negative pairs, where addresses a_i and a_j in each pair have either different tags from the same category or tags from a forbidden pair of categories, see details in Section VI.

V. The Algorithm for automatic Bitcoin network clustering

In this section we are going to describe the algorithm for Bitcoin address clustering which regulates to balance information coming directly from Bitcoin blockchain (CS and OTC heuristics) and the additional information gathered from the Internet in the form of tags as described in Section IV.

A. Notations

Let us introduce the following notations. Let $T = \{t_j\}$ be the set of all transactions in Bitcoin blockchain and A be the set of all addresses present in transactions from T. The clustering of Bitcoin addresses is a decomposition $A = A_1 \cup A_2 \cup \cdots \cup A_N$ into non-intersecting subsets $A_l \bigcap A_j = \emptyset$ for $l \neq j$. We also denote by $T_H \subset T$ the set of all transactions which satisfy either CS or OTC heuristics. For the transaction $t \in T_H$ we denote by $Addr_H(t)$ the set of all addresses which should be attributed to the single user according to one of the heuristics. We note, that by construction only one of the heuristics can be satisfied for a single transaction t. The information about tags is represented as a set of negative pairs $L = \{(a_i, a_j)\}$. The pair of addresses $(a_i, a_j) \in L$ if we have a piece of information that these addresses are not controlled by the same user (see Section IV-B for details).

B. Probabilistic model

We note that both CS and OTC heuristics and the set of negative pairs L may contain erroneous information. To deal with this situation, we propose a probabilistic framework which allows specifying our confidence to different sources of data. We are going to consider different types of observations (which we *treat as an independent* to make it computationally solvable):

- In the event that all the addresses $Addr_H(t)$ for some $t \in T_H$ indeed belong to the same user is true with probability p.
- In the event that two addresses $\{a_i, a_j\} \in L$ are controlled by the same user is true with probability q. In other words, the information about the negative association between any pair of addresses in L is validated by the probality $1 - q$.

Let the likelihood $\mathbb{P}(A, T_H, L \mid p, q)$ be a function of the clustering A, transactions T_H and negative pairs L:

$$\mathbb{P}(A, T_H, L \mid p, q) =$$
$$= \prod_{t \in T_H} p^{\mathbb{I}\left(Addr_H(t) \subset Cl(A)\right)} \times (1-p)^{\mathbb{I}\left(Addr_H(t) \not\subset Cl(A)\right)}$$
$$\times \prod_{\{a,a'\} \in L} (1-q)^{\mathbb{I}\left(\{a,a'\} \not\subset Cl(A)\right)} \times q^{\mathbb{I}\left(\{a,a'\} \subset Cl(A)\right)},$$

where for some set of Bitcoin addresses S the notation $S \subset Cl(A)$ means, that there exists a cluster A_l such that $S \subseteq A_l$.

Finally, the log-likelihood reads as

$$\begin{aligned}&\ln \mathbb{P}\left(A, T_H, L \mid p, q\right) = \\ &= \sum_{t \in T_H} \mathbb{I}\big(Addr_H(t) \subset Cl(A)\big) \ln(1-p) \\ &+ \sum_{t \in T_H} \mathbb{I}\big(Addr_H(t) \not\subset Cl(A)\big) \ln(p) \\ &+ \sum_{\{a,a'\} \in L} \mathbb{I}\big(\{a,a'\} \not\subset Cl(A)\big) \ln(1-q) \\ &+ \sum_{\{a,a'\} \in L} \mathbb{I}\big(\{a,a'\} \subset Cl(A)\big) \ln(q).\end{aligned} \tag{1}$$

We note that the proposed model is not intended to capture the probabilistic structure of the real world, but more to give an approach for systematical study of confidence trade-offs between different sources of information. Moreover, it allows efficient optimization of parameters as discussed in the next section.

C. Maximization of likelihood

Maximization of log-likelihood (1) is a discrete optimization problem which is in fact NP-hard. We suggest solving it by using a greedy approach. We will go retrospectively through all the transactions in the Bitcoin network, which satisfy one of the heuristics. On each step, we decide whether to join the clusters corresponding to the addresses $Addr_H(t_j)$ for the considered transaction t_j based on the values of the log-likelihood functional. Let $\hat{A}_j = A_{k_1} \cup \cdots \cup A_{k_{m_j}}$ be a union of all the clusters which representatives belong to $Addr_H(t_j)$. Let us find the change in the number of negative pairs if we join all the clusters corresponding to $Addr_H(t_j)$ into one cluster $\hat{A}_j$:

$$\Delta_{t_j}\Bigg(\sum_{\{a,a'\} \in L} \mathbb{I}\big(\{a,a'\} \subset Cl(A)\big)\Bigg) = \sum_{\{a,a'\} \in \hat{A}_j} \mathbb{I}\left(\{a,a'\} \in A_l\right)$$
$$- \sum_{i=1}^{m_j} \sum_{\{a,a'\} \in A_{k_i}} \mathbb{I}\left(\{a_i,a_j\} \in A_l\right) = \Delta_{\hat{A}_j} - \sum_{i=1}^{m_j} \Delta_{A_i},$$

where Δ_{A_m} is a number of negative pairs in cluster A_m. Then, if we merge all the clusters corresponding to $Addr_H(t_j)$, the change of the log-likelihood (1) is equal to

$$\begin{aligned}&\Delta_{\mathbb{P}}(t_j, A, L \mid p, q) = \\ &= \ln\left(\frac{p}{1-p}\right) + \left(\Delta_{\hat{A}_j} - \sum_{i=1}^{m_j} \Delta_{A_{k_i}}\right) \ln\left(\frac{q}{1-q}\right).\end{aligned}$$

Thus, if $\Delta_{\mathbb{P}}(t_j, A, L \mid p, q)$ is positive, then we merge all the clusters corresponding to $Addr_H(t_j)$, otherwise need to we continue with the next transaction.

D. Refinements of the algorithm

We note that due to the greedy (**historical**) approach the change of parameters p and q can lead to very non-monotone changes in the clustering. For example, we can decrease parameter q, which in principle should lead to smaller clusters, but find out that the largest cluster becomes even larger. To overcome this we propose, on the first step, to perform clustering with $q \to 0$, where the clusters are not allowed to contain any negative pairs. In the second step, we once again go through Bitcoin transactions historically and optimize likelihoods as discussed in the previous section. We call this approach the *greedy additive clustering* (**add**) as opposed to the pure historical approach discussed previously.

VI. Experiments

A. Data

In our experiments we considered the blockchain data that contained transactions from Bitcoin blockchain from 3^d January of 2009 to 9^{th} March of 2017. During this period there were 211,789,876 transactions which cover 244,030,115 unique addresses.

The important question is what part of Bitcoin addresses can be covered by CS and OTC heuristics (see Section III). For the considered data, the CS heuristic condition is satisfied for 8,161,086 transactions with 28,416,034 addresses while the OTC heuristic condition holds for 35,844,487 OTC transactions with 69,520,194 unique addresses. Both conditions give a total of 44,005,573 covered transactions with 95.250.167 unique addresses (the overlap is 2,686,061 addresses). It means that with this information we can cluster only slightly more than $1/6$ of the whole Bitcoin blockchain.

services	57
gambling	80
mixer	3
dnm	16
exchange	98
pool	52

TABLE I: Unique clean tags per category.

Off-chain information was collected from 97 sources (*twitter.com, walletexplorer.com*, etc.). It contains more than 20 million clean tags with 305 unique values (*Bitstamp.net, Eligius mining pool*, etc.). Clean tags are distributed between six categories, see the number of unique tags per category in Table I. There exists only one tag (*BTCChina*) which belongs to two categories simultaneously: exchange and pool. Our preprocessing left 335,000 dirty tags with approximately 105,000 unique values (*mrdeposit, crypto_bot*, etc.). We note that in principle this information can be still useful, although but we do not use it in this study.

Looking throughout off-chain information we discovered out that some addresses have multiple distinct clean tags, see Table II. For example, some addresses may have 2-3 categories with one tag in each one or they may have more than one tag in one category. Such situations can either indicate that different types of resources are owned by the same user, or can be artifacts of data collection. The first situation can be illustrated by the following example: there exist addresses with tags *CoinChimp.com* (exchange) and *BitLaunder.com* (mixer). The search over the Internet reveals that both services are owned by the same person [34]. However, tags from some categories are unlikely to be present for the assets of one person. For example, well-known exchanges are impossible to be affiliated with some Darkmarket addresses. We collected the information about appearances of different clean tags, see Table II. We use this information as guidance for further clustering, i.e. we consider any pair of different clean tags from

	services	gambling	mixer	dnm	exchange	pool
services	165,970	66,387	0	0	441	186
gambling	66,387	11,9857	0	0	0	9
mixer	0	0	0	0	1,703	1
dnm	0	0	0	0	13	18
exchange	441	0	1,703	13	606,357	198,551
pool	186	9	1	18	198,551	1,561

TABLE II: The number of addresses with distinct clean tags in respective pairs of categories.

one category as a negative pair and also consider two addresses with different clean tags as a negative pair if tags from these categories never mark the same address in the Bitcoin network. Moreover, we call a negative pair of any combination of tags where one of the tags corresponds to Darkmarket as well as a combinations service – exchange, mixer – exchange and gambling – pool. The appearance of such pairs in our tags is most likely due to the flaws in tagging procedure. All this information is present in Table III.

	services	gambling	mixer	dnm	exchange	pool
services	F	A	F	F	F	A
gambling	A	F	F	F	F	A
mixer	F	F	F	F	F	F
dnm	F	F	F	F	F	F
exchange	F	F	F	F	F	A
pool	A	A	F	F	A	F

TABLE III: Table shows if the appearance of addresses having two distinct clean tags in the same cluster should be forbidden. "F" stays for forbidden and "A" is for allowed.

B. Clustering

We started our experiments by considering clustering based solely on CS and UTC heuristics. As a result, we obtained clustering that covers 95,250,167 addresses and contains 14,117,435 clusters. It appears that the biggest cluster in this clustering has a size of 26,694,671 addresses containing addresses with clean tags from all six categories, see Table IV. Moreover, this clustering has 249 clusters with negative pairs of addresses totaling more than $2.3 \cdot 10^{13}$ negative pairs with the majority being in the largest cluster. It means that we can use the off-chain information to get more (fine-grained) clustering information.

Category	Number of tags	Number of common tags (size)	Examples of common tags
services	33	5 (> 100K)	*Bitpay.com, Xapo.com*
gambling	34	6 (> 50K)	*999Dice.com, primedice.com*
mixer	3	1 (> 100K)	*BitcoinFog*
dnm	14	5 (> 100K)	*SilkRoad Marketplace*
exchange	64	12 (> 100K)	*BTC-e.com, Bittrex.com*
pool	15	2 (> 50K)	*BTCChina, Hashnest.com*

TABLE IV: Tags of the biggest cluster in case of clustering without constraints.

The opposite case is to completely forbid the appearance of a negative pair of addresses in one cluster. It means that we skip all CS and OTC transactions that can lead to the formation of a cluster with a negative pair inside it. We note that this leads to a certain number of addresses to be omitted from clustering. This is due to the fact that some addresses from $Addr_H(t)$ may form a negative pair for some OTC/CS transactions t. Such clustering covers 94,851,585 addresses and contains 14,133,381 clusters. We note, that 50,666 CS and 27,877 OTC transactions are skipped. The biggest cluster has a size of 2,475,769 addresses. Importantly, large clusters have fewer distinct categories of tags than before:

1) The largest cluster has a size of 2.48M addresses and contains
   - 8,809 addresses with tag *BitReserve.com* (services);
   - 83,732 addresses with tag *NitrogenSports.eu* (gambling);
   - 8 addresses with tag *Eligius mining pool* (pool).
2) Second largest cluster has a size of 2.26M addresses and contains
   - 23,202 addresses with tag *Circle pay app* (services);
   - Six addresses with tag *999Dice.com* (gambling);
   - Ten addresses with tag *Eligius mining pool* (pool).

We note that we can create more fine grained and possibly more accurate clustering if we have more off-chain information or use more of available one (i.e. "dirty" tags).

However, it is interesting to explore the intermediate cases via probabilistic algorithm introduced in Section V. We use a historical (chronological) order for treating transactions (see Section V-C) and greedy additive clustering (add), see Section V-D. We study, how the parameters p and q influence the resultant clustering. As only relative values of these parameters matter, we fix $p = \frac{3}{4}$ and vary q in such a way, that a certain number of negative pairs (denoted by Δ_{step}) is allowed to appear in clustering on one iteration. In other words, Δ_{step} is the maximum allowed increase in the number of negative pairs on one step of the algorithm. We obtain the following dependencies of the largest cluster size (Figure 1) and the number of negative pairs (Figure 2) depending on Δ_{step}. These graphs show that the greedy additive approach yields monotonic dependence and results in smaller clusters with less negative pairs.

VII. Conclusions

In this work, a new Bitcoin address clustering algorithm is proposed. Its difference from the existing ones is two-fold. Firstly, it uses for clustering not only blockchain information but also off-chain information from the Internet. Secondly, we treat certain off-chain data types as votes against address union in clustering process. Such approach allows to avoid significant part of erroneous cluster merges suggested by blockchain based heuristics. Numerical experiments show that the proposed approach provides reasonable clustering results outperforming approaches based solely on blockchain data in terms of cluster homogeneity.

Acknowledgments.

D. Ermilov and Yu. Yanovich were supported by Bitfury Group. M. Panov was supported by the Russian Science Foundation grant (project 14-50-00150).

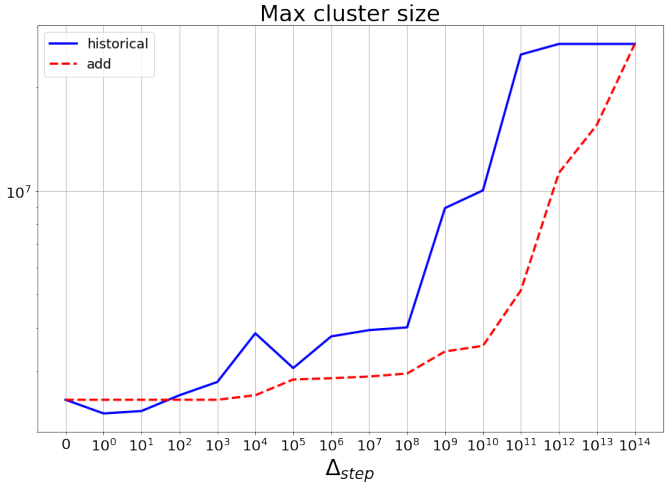

Fig. 1: The size of the biggest cluster, where Δ_{step} is allowed to increase the number of negative pairs per transaction.

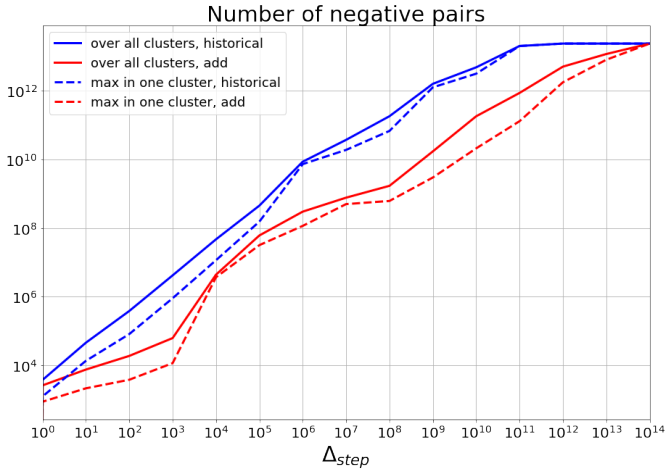

Fig. 2: Number of negative pairs in whole clustering and the maximum in one cluster, where Δ_{step} is allowed to increase the number of negative pairs per transaction.

References

[1] S. Nakamoto. (2008). Bitcoin: A peer-to-peer electronic cash system. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] Q.C. ShenTu and J.P. Yu. (2015). Research on anonymization and de-anonymization in the Bitcoin system. [Online]. Available: https://arxiv.org/abs/1510.07782

[3] F. Reid and M. Harrigan "An analysis of anonymity in the Bitcoin system". In: Proc. 3rd IEEE International Conference on Privacy, Security, Risk and Trust and on Social Computing, SocialCom/PASSAT'11. Pp. 1318–1326. – 2011. doi:10.1007/978-1-4614-4139-7_10

[4] D. Ron and A. Shamir (2012). Quantitative analysis of the full bitcoin transaction graph. Cryptology ePrint Archive, Report 2012/584. [Online]. Available: http://eprint.iacr.org/2012/584

[5] Conti, M., Lal, C. and Ruj, S. (2017). A Survey on Security and Privacy Issues of Bitcoin. arXiv preprint arXiv:1706.00916.

[6] G. Maxwell. (2013). CoinJoin: Bitcoin privacy for the real world. [Online]. Available: https://bitcointalk.org/index.php?topic=279249.0

[7] G. Maxwell (2013). CoinSwap: transaction graph disjoint trustless trading. [Online]. Available: https://bitcointalk.org/index.php?topic=321228.0

[8] J. Bonneau et al. "Mixcoin: Anonymity for Bitcoin with accountable mixes". In: Proc. 18th International Conference on Financial Cryptography and Data Security, FC'14, pp. 486504, 2014. doi:10.1007/978-3-662-45472-5_31

[9] Y. Yanovich et. al. "Shared Send Untangling in Bitcoin". 2016. [Online]. Available: http://bitfury.com/content/5-white-papers-research/bitfury_whitepaper_shared_send_untangling_in_bitcoin_8_24_2016.pdf

[10] S.T. QingChun and Y. JianPing. Transaction Remote Release (TRR): A New Anonymization Technology for Bitcoin, [Online]. Available: http://arxiv.org/pdf/1509.06160v1, 2015.

[11] E. Ben-Sasson et al. (2014). Zerocash: decentralized anonymous payments from Bitcoin (extended version). Cryptology ePrint Archive, Report 2014/349. https://eprint.iacr.org/2014/349

[12] D. Kaminsky. "Black Ops of TCP/IP". Black Hat USA. 2011. P. 44.

[13] A. Biryukov and I. Pustogarov. "Bitcoin over Tor isn't a good idea". Security and Privacy (SP), 2015 IEEE Symposium on. IEEE, 2015. pp. 122-134.

[14] M. Harrigan and C. Fretter. "The Unreasonable Effectiveness of Address Clustering." Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), 2016 Intl IEEE Conferences, pp. 368-373. IEEE, 2016.

[15] S. Meiklejohn et al. "A fistful of Bitcoins: characterizing payments among men with no names". In: Proc. 2013 Internet Measurement Conference, IMC'13. Pp. 127–140, 2013, doi:10.1145/2504730.2504747

[16] Fleder, Michael, Michael S. Kester, and Sudeep Pillai. "Bitcoin transaction graph analysis." arXiv preprint arXiv:1502.01657 (2015).

[17] Androulaki, Elli, et al. "Evaluating user privacy in bitcoin." International Conference on Financial Cryptography and Data Security, pp. 34-51. Springer, Berlin, Heidelberg, 2013.

[18] J. Monaco."Identifying Bitcoin Users by Transaction Behavior", Proc. SPIE 9457, Biometric and Surveillance Technology for Human and Activity Identification XI, pp. 1-15, 2015.

[19] M. Spagnuolo et. al. "Bitiodine: Extracting intelligence from the bitcoin network". In International Conference on Financial Cryptography and Data Security. Springer, Berlin, Heidelberg. Pp. 457–468. – 2014.

[20] M. Mser et. al. "Towards risk scoring of Bitcoin transactions". International Conference on Financial Cryptography and Data Security, pp. 16-32. Springer, Berlin, Heidelberg, 2014.

[21] N. Jonas. "Data-Driven De-Anonymization in Bitcoin". Master Thesis, pp. 1-41. ETH Zurich. – 2015.

[22] B. Huang et al. "Behavior pattern clustering in blockchain networks." Multimedia Tools and Applications. Pp. 1–12. – 2017.

[23] J. Osterrieder and J. Lorenz. "A Statistical Risk Assessment of Bitcoin and its Extreme Tail Behavior." Annals of Financial Economics, vol. 12, number 01, pp. 1-19. – 2017.

[24] Ghahramani, Zoubin. "Unsupervised learning." Advanced lectures on machine learning. Springer Berlin Heidelberg, 2004. 72-112.

[25] Jain, Anil K., M. Narasimha Murty, and Patrick J. Flynn. "Data clustering: a review." ACM computing surveys (CSUR) 31.3 (1999): 264-323.

[26] S. Fortunato. "Community detection in graphs." Physics reports 486.3. Pp. 75–174. – 2010.

[27] BIP 141 on Github. (2015) [Online]. Available: [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki

[28] BIP 91 on Github. (2017) [Online]. Available: [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0091.mediawiki

[29] A.M. Antonopoulos. Mastering Bitcoin: unlocking digital cryptocurrencies. OReilly Media, 2014.

[30] Secp256k1. In: Bitcoin Wiki (2015) [Online]. Available: [Online]. Available: https://en.bitcoin.it/wiki/Secp256k1

[31] Multisignature. In: Bitcoin Wiki (2015) [Online]. Available: https://en.bitcoin.it/wiki/Multisignature

[32] V. Buterin (2014). Bitcoin multisig wallet: the future of Bitcoin [Online]. Available: https://bitcoinmagazine.com/articles/multisig-future-bitcoin-1394686504

[33] Meiklejohn, Sarah, et al. "A fistful of bitcoins: characterizing payments among men with no names." Proceedings of the 2013 conference on Internet measurement conference, pp. 127-140. ACM, 2013.

[34] Bitplastic SCAM https://www.reddit.com/r/Bitcoin/comments/1v9pzr/bitplastic_scam/cgqdpjv/?sh=258b3315&st=iud9x1zg