Reducing Privacy of CoinJoin Transactions: Quantitative Bitcoin Network Analysis

Anton Wahrstätter, *Member, IEEE,* Alfred Taudes, and Davor Svetinovic, *Senior Member, IEEE*

Abstract—Privacy within the Bitcoin ecosystem has been critical for the operation and propagation of the system since its very first release. While various entities have sought to deanonymize and reveal user identities, the default semi-anonymous approach to privacy was judged as insufficient and the community developed a number of advanced privacy-preservation mechanisms. In this study, we propose an improved variant of the multiple-input clustering approach that incorporates advanced privacy-enhancing techniques. We examine the CoinJoin-adjusted user graph of Bitcoin through quantitative network analysis and draw conclusions on the effectiveness of our proposed clustering method compared to naive multiple-input clustering. Our findings indicate that CoinJoin transactions can significantly distort commonly applied address clustering approaches. Moreover, we demonstrate that Bitcoin's user graph has become less dense in recent years, concurrent with the collapse of several independent user clusters. Our results contribute to a more comprehensive understanding of privacy aspects in the Bitcoin transaction network and lay the groundwork for developing enhanced measures to prevent money laundering and terrorism financing.

Index Terms—Bitcoin, Blockchain, Anonymity, Privacy.

1 Introduction

Complex network analysis has already been applied to various different domains such as social networks [1], [2], malware propagation [3] or digital payment protocols [4], [5].

Complex network analysis has been accelerated by an increase in publicly available data and computational power in recent years. This has enabled researchers to concentrate on network analysis and leverage graph analysis tools to describe (social) network properties. As data sets grow larger and more complex, a variety of metrics have demonstrated their reliability in conducting studies related to highly rational data.

Data science has evolved alongside blockchain-based digital currencies, beginning with Bitcoin. These digital, decentralized currencies enable users to transfer monetary value peer-to-peer without the need for centralized trust. By viewing Bitcoin transactions as connections between nodes (users) in a financial network, it becomes feasible to analyze the digital currency, its ecosystem, participants, and privacy aspects from a network theory perspective.

Monetary system networks are crucial for economic markets, not only for combating money laundering and financing of terrorism but also due to their impact on real-world markets. Understanding these systems to track digital money flows has become essential, especially with the rise of cryptocurrencies.

- *A. Wahrstätter and A. Taudes are with the Department of Information Systems and Operations Management, Vienna University of Economics and Business, Vienna, Austria. E-mail: {anton.wahrstaetter, alfred.taudes}@wu.ac.at*
- *D. Svetinovic is with the Department of Information Systems and Operations Management, Vienna University of Economics and Business, Vienna, Austria, and the Center for Secure Cyber-Physical Systems, Department of Computer Science, Khalifa University, Abu Dhabi, UAE. E-mail: dsve@acm.org*

Manuscript received April 19, 2005; revised August 26, 2015.

The Bitcoin blockchain, the most notable distributed ledger application today, can be modeled as a complex network to examine its intricate topological features. Public blockchain systems, in general, offer ideal conditions for researchers to conduct quantitative analyses. Unlike traditional data sets maintained by a central entity responsible for their integrity, authenticity, and availability, data recorded in public blockchains like Bitcoin is openly accessible and consistently maintained by a decentralized network participants. This accessibility not only eliminates entry barriers but also allows researchers to generate fully verifiable and reproducible results.

Bitcoin went through turbulent times in recent years, with privacy emerging as a paramount and contentious issue among its developers, users and adversaries. Recently, researchers have honed in on enhancing privacy-preserving techniques, for example, [5]–[8]. One prevalent approach involves the use of mixing transactions (CoinJoins), which are designed to impede traceability. At the same time, new analysis techniques are continually emerging to develop a more precise user graph [9], [10], potentially undermining or breaking privacy. The most recent complete-record network analysis of Bitcoin's user graph has been carried out in 2021 with blockchain data reaching until 2017 [11].

The Bitcoin blockchain went through times of broad adoption since the last completed study. Therefore, it is crucial to provide an up-to-date snapshot of the cryptocurrency's transaction and user network, as well as an analysis of the privacy-increasing techniques.

The growth of Bitcoin can be seen in Figure 1 (a), depicting the number of constant-size raw block files (blk*****.dat files) that were created each month. These files contain binary data representing blocks of the blockchain. It shows that the size of the Bitcoin blockchain grew significantly since 2017, having months in which more than 40 constant-size blk files were produced by the Bitcoin core client. In

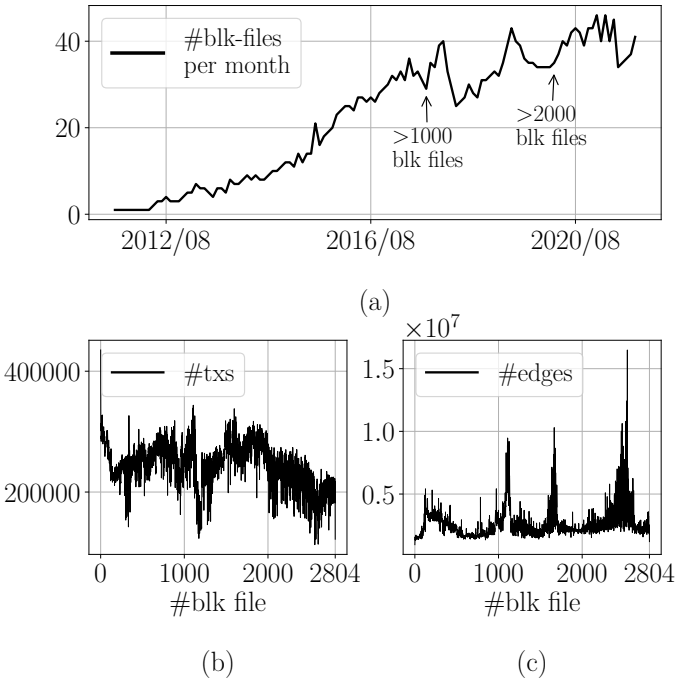

Fig. 1: Visualization of the raw blockchain data files over time.

parallel, the average number of transactions per blk file has steadily been decreasing in recent times (b). On the other hand, Figure 1 (c) highlights spikes in the number of edges per block file. Given that the size of blk files is capped at 128 MiB, we can infer that the number of edges per transaction may have increased recently, indicating an upsurge in complexity of Bitcoin's transaction graph.

Furthermore, we observe a present trend towards deanonymization studies, many of which address the inclusion or enhancement of CoinJoin transactions in an appropriate manner [6], [7], [12], [13].

Our study builds upon the work of Ermilov *et al.* [9] and Möser and Narayanan [10]. We expand their research by analyzing new data related to the significant growth of the Bitcoin transaction network and introduce methods to cluster privacy-focused users beyond obscuring transactions. Consequently, we develop a more cautious address clustering approach that considers on-chain obfuscation techniques. By applying the heuristics developed by researchers from the field of CoinJoin transaction detection, such as Ficsór [14] and Stockinger *et al.* [13], we conduct the first study that definitively considers trustless mixing of transactions when constructing the user graph. This approach refines the final user graph, maintaining its characteristics comparable to other complex social networks.

This research is significant given the substantial increase in CoinJoin transactions in recent years, driven by heightened privacy concerns. As privacy-enhancing transactions continue to expand, certain methodologies, like the widely used multiple-input clustering technique, which presumes that multiple input addresses within a transaction belong to the same user, must be reevaluated to prevent the merging of numerous privacy-focused users into a single large supercluster. A single CoinJoin transaction executed by addresses from different clusters could result in their unification, potentially causing significant distortion in an analysis of the ecosystem and its participants.

Particularly in the fight against money laundering and terrorist financing, it is crucial to maintain a reliable understanding of ownership structures and track payments throughout the protocol. Money flow analysis is a key forensic method for law enforcement agencies [15]. Given that users are able to create an unlimited number of addresses, the primary challenge for blockchain analysis is to cluster multiple addresses that belong to a single user.

In response to the growing number of obfuscating transactions, we have modified the multiple-input heuristic. Although CoinJoin transactions have played a negligible role over years, their recent integration with widely adopted and user-friendly wallets such as Wasabi Wallet and Samourai Wallet has significantly increased the number of untraceable and private transactions within the Bitcoin network.

In the main part of this work, we examine the Bitcoin user graph constructed using our improved variant of the multiple-input heuristic, termed the "CoinJoin-adjusted multiple-input-heuristics." This approach considers obfuscating transactions by first detecting them and then excluding them when executing the clustering algorithm and constructing the user graph. Furthermore, we introduce stringent cluster-collapse constraints to prevent the collapse of publicly known and independent clusters. This allows us to present a comparison between the naive multiple-input clustering and our CoinJoin-adjusted variant.

This is the first study to deterministically account for on-chain obfuscation techniques when constructing the Bitcoin user graph, adding another layer of detail to our quantitative network analysis This also enables us to provide a snapshot of the ongoing increase in trustless CoinJoin transactions within the ecosystem, assess their impact, and evaluate the privacy level of Bitcoin transactions after years of continuous improvement in clustering heuristics. With particular attention to privacy-related aspects, we provide the analysis of the selected privacy-enhancing practices that have been applied within the transaction network of Bitcoin. Our findings have potential to improve methods to combat money laundering and terrorist financing.

We also made our `python-bitcoin-scraper` available in an open-source Github repository[1], translate the CoinJoin detection heuristics , originally developed in C#, into SQL and ensure full transparency of the research process by documenting it in publicly accessible Jupyter Notebooks.

2 Background

2.1 Bitcoin

Introduced in 2008 [16], Bitcoin is a payment system that utilizes blockchain technology, eliminating the need for trusted intermediaries to prevent double-spending of assets. Instead, the system's integrity is maintained through resource-intensive mining. The blockchain serves as a decentralized public ledger, stored on numerous computers worldwide, and records every transaction since the inception of the first block, or the "genesis block," in January 2009. The Bitcoin

1. https://github.com/Nerolation/python-bitcoin-scraper

blockchain comprises blocks, each containing transactions, which are cryptographically connected through hash functions, forming an unalterable chain. Bitcoin operates on public key cryptography. Each user of this digital currency possesses one or more key pairs, each consisting of a public and a private key. Users can exchange Bitcoins within the network by broadcasting transactions peer-to-peer, which are signed with their unique private key.

"Miners" in the Bitcoin network run client software to solve proof-of-work puzzles. When a miner successfully solves a puzzle, they can propose the next block to be added to the blockchain. Miners monitor unconfirmed transactions, bundle them into blocks, and append them to the Bitcoin blockchain. They receive Bitcoin (BTC) as a reward for solving the cryptographic puzzle, which includes a block reward (6.25 BTC as of 2023) and transaction fees from the newly created block. Typically, a new block is added every 10 minutes. Miners also validate other miners' blocks and actively participate in Bitcoin's consensus mechanism. Users broadcast their transactions to the network, which are then held in the "mempool," a local temporary storage for unconfirmed transactions until a miner incorporates them into a block.

Aside from the "coinbase" transactions, which are always the first transaction in a block and are created by the miner to claim the block reward, every transaction must have at least one non-empty input and output. In the input section of a transaction, users must specify a transaction output from another transaction that was addressed to them and has not been used as an input in another transaction. Unlike account-based systems where an account's balance is updated after each transaction, Bitcoin uses the Unspent Transaction Output (UTXO) model. Nodes in the Bitcoin network track every UTXO, allowing them to calculate the balance of an account by adding up all its associated UTXOs.

A transaction can include multiple inputs and outputs, allowing users to bundle several UTXOs associated with different public keys and distribute them to various entities in a single transaction. The fee paid to miners for processing the transaction is the difference between the total input values and output values. The Bitcoin protocol requires UTXOs to be spent in full. Therefore, if the input values surpass the desired output values, users must provide a change address in the transaction outputs. Typical examples of Bitcoin transactions are depicted in Figure 2.

Nakamoto [16], in his Bitcoin whitepaper, advised for privacy reasons to use new key pairs for every transaction, implying the use of a new one for change as well. Since multiple inputs within one transaction might have been signed by a single user who is in possession of the respective private keys to unlock each input, multiple input addresses could be grouped in way that they originate from a single user. This concept was later refereed to as the multiple-input heuristics when researchers [11], [17]–[22] deconstructed the transaction graph into a user graph and clustered users based on multiple input addresses within a transaction.

The limitation of multiple-input heuristic clustering is that different users are technically capable to collaborate and spend their assets within a shared transaction. This approach, known as CoinJoin, was first introduced in the *bitcointalk* forum by Maxwell [23] in 2013. CoinJoin transactions represent an important privacy-enhancing technique and are crucial for Bitcoin's fungibility. These transactions have been present since 2013, but notably, the emergence of more privacy-focused wallet applications such as Wasabi or Samurai Wallet in recent years has led to an increasing number of obfuscating transactions.

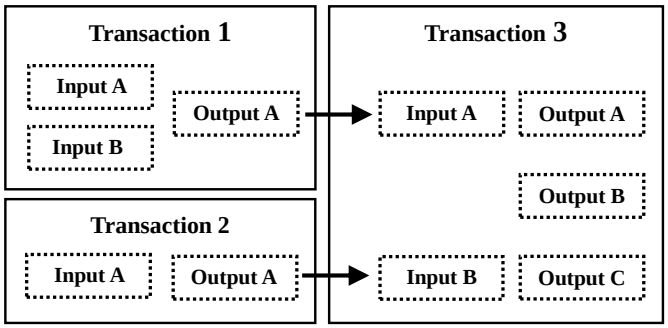

Fig. 2: Bitcoin transaction in which two UTXOs that originate from Transaction 1 and 2 are combined in Transaction 3, which sends BTC to the addresses in Output A and Output B with Output C (probably) representing the change address.

2.2 CoinJoins

CoinJoin transactions were conceived as a response to deficiencies in privacy and fungibility within the Bitcoin ecosystem. Nakamoto, in the original Bitcoin white paper [16], admitted that some level of traceability would be inevitable as multiple inputs within a transaction might have been signed by the same user. This implies that users could be clustered based on common inputs of different transactions. To address this traceability issue and ensure that Bitcoin retains its fungibility *Maxwell* [23] proposed CoinJoin transactions, emphasizing that such transactions would not require any changes to the protocol, as the technical foundation for these transactions has always existed within Bitcoin's architecture.

CoinJoin transactions leverage the ability to combine multiple UTXOs, even when they belong to separate real-world entities. As illustrated in Figure 2, transaction 3 could potentially represent a CoinJoin transaction, indicating distinct inputs that belong to different users collaborating to transfer their funds.

As the multiple-input heuristic clusters addresses based on transactions sharing common inputs, CoinJoins can disrupt this process by leading the method to erroneously categorize different users within the same cluster. Even a single CoinJoin transaction between two disparate entities could result in their amalgamation into a single cluster.

Notably, services that allow users to import their own private keys might inadvertently generate CoinJoin transactions by combining the UTXOs of their users in new transactions. Such transactions might act as a link between different clusters and eventually lead to their merging. For instance, the prominent Mt. Gox exchange, which permitted users to import their private keys, has led to inaccuracies in multiple-input clustering outcomes [20].

In Bitcoin's early days, centralized services such as *Bitcoin Fog* or *Bitcoin Laundry* coordinated CoinJoins for a subset of privacy-seeking users [24]. With increased privacy efforts, CoinJoin transactions have been implemented into

TABLE 1: CUMULATIVE STATISTICS ON COINJOINS

	$\#txs$	$\#edges$	$\overline{edges/tx}$	$\overline{\#outputs}$
wasabi	21,414	222,597,392	10,395	119
samurai	82,763	2,069,075	25	5

trustless applications, making the technique more widely accessible. Notably, Wasabi Wallet and Samourai Wallet significantly contributed to an improved privacy regime within Bitcoin. By pooling together the UTXOs of users, such wallets provide essential privacy features while pushing the limits of traceability and address clustering.

Wasabi Wallet and Samurai Wallet offer distinct implementations of the ZeroLink framework [25]. Both provide users a way to participate in CoinJoin transactions to attain a higher level of privacy. However, users need to adhere to the best practices suggested by both wallets to avoid risking deanonymization. It is of crucial importance for users to avoid co-spending mixed UTXOs with non-mixed ones or consolidating multiple mixed outputs within a single transaction after the mixing procedure.

While both services are founded on the same principles, their transactions appear markedly different. For Wasabi Wallet, CoinJoin transactions may contain more than 100 outputs on average. Conversely, Samurai Wallet transactions are strictly composed of 5 inputs and outputs, with equal output denominations that correspond to one of Samurai's offered pool sizes. These pools sizes are available for 0.001, 0.01, 0.05 and 0.5 BTC.

As per our observations,, Wasabi Wallet generates more complex transactions compared to Sarmurai Wallet. Wasabi CoinJoin transactions typically consist of around 10.000 (100x100) edges per transaction contrasting Samurai CoinJoins, which have a constant number of edges of 25 (5x5). However, the total count of Samurai CoinJoin transactions is nearly four times higher than that of Wasabi CoinJoins. We present these statistics in Table 1.

For our analysis it is crucial to consider CoinJoin transactions appropriately, given their potential to significantly distort network analyses [15]. Building a user graph using naive multiple-input clustering results in the merging of clusters that belong to various different entities that are part of the same CoinJoin transaction. As Wasabi CoinJoins often count more than 100 different input addresses, each single input address would be categorized into the same cluster. For Samurai CoinJoins, we would inaccurately cluster 5 inputs per transaction.

Since it is impossible to determine the actual number of entities involved in CoinJoins, our approach that strictly separates them can be considered more conservative than naive multiple-input clustering. Nonetheless, we do not attempt to cluster addresses within CoinJoin transactions, which may lead to us counting users more than once if they participate multiple times in a single CoinJoin transaction.

3 Related Work

The exploration of the Bitcoin network extends into a wide range of specialized areas. This includes, but is not limited to, the study of network statistics and visualization, identifying anomalies and money laundering, and undertaking deanonymization and clustering studies. Frequently, researchers apply multiple of the mentioned approaches and combine them to analyze the blockchain from distinct angles.

Studies examining the Bitcoin network holistically often utilize various heuristics to dissect the transaction graph into a user graph, aiming to gain a more precise understanding of the network and its intrinsic properties. In the following section, we will elaborate on select approaches applied to the Bitcoin network. We categorize the literature into *User Graph Clustering* and *CoinJoins*, given that these categories form the bedrock of our work and the foundation upon which we build. Our primary emphasis is on recent entity-mining attempts that employ clustering heuristics or deanonymization techniques as alternatives to methods that use the complete transaction graph as a launchpad for further statistical analysis.

3.1 User Graph Clustering

Harrigan and Fretter [20] outlined the efficacy of the multiple-input clustering approach along with an explanation of the factors contributing to its effectiveness. Upon analysis of the Bitcoin graph up to 2016, they noted high levels of address reuse, avoidable merging, large clusters with high centrality measures, and the progressive expansion of clusters as the primary catalysts for successful input-based clustering methodologies. While they did acknowledge the limitations of the multiple-input heuristic concerning CoinJoin transactions, they did not introduce any countermeasures to account for such transactions.

Maesa *et al.* [26] conducted a formal description and application of the multiple-input clustering approach to the Bitcoin transaction graph up to 2015. They observed a rising densification over time and corroborated the network's scale-free attribute. Moreover, they examined the concentration of monetary value within the network and endorsed the "rich-get-richer" hypothesis.

In a subsequent study, Maesa *et al.* [21] explored Bitcoin's connectivity components. Using sequential, temporal snapshots, they presented the graph in a bow-tie structure and clustered nodes based on their connectivity to other nodes and economic data.

Ermilov *et al.* [9] developed a more sophisticated user graph that paid heed to cluster collapses. Based on multiple-input and change-address heuristics, they deconstructed the Bitcoin transaction graph up to March 2017 into a user graph, utilizing off-chain information such as address tags/labels to prevent the clustering of different entities within a single cluster. Their study, however, focused more on improving the address clustering process with a probabilistic tool without explicitly considering CoinJoins. On the contrary, our study delves deeper into selected network properties of our resulting graph.

Alqassem *et al.* [27] analyzed the Bitcoin blockchain up to 2014, quantifying the evolution of the network's key structural properties. They transformed the transaction graph into a user graph by applying the multiple-input heuristic. They then compared the transaction graph with the approximated user graph, measuring large diameters,

confirming a densification power law, and revealing community structures. They also detected anonymity-seeking behavior among users of the Bitcoin protocol.

Möser and Narayanan [10] developed a method for clustering the Bitcoin transaction graph, utilizing a unique version of the change-address heuristic applied to blockchain data up to June 2020. They identified CoinJoins as a potential error source in the address clustering process. To counteract the formation of massive, incorrectly clustered groups, they deployed a random forest classifier to predict if an output address was being used as a change address. This approach enabled them to decrease the likelihood of falsely clustered addresses using probabilistic methods. Unlike their user graph, our approach involves the detection of CoinJoin transactions using a selected set of deterministic rules. This detection is taken into account during the user graph creation process. Additionally, we strictly avoid cluster collapses of known clusters to prevent the misattribution of multiple entities to distinct addresses.

3.2 CoinJoins

The detection of CoinJoin transactions represents a relatively young research discipline, even though the underlying concept has been introduced already in 2013 by Maxwell [23]. The rise in the prevalence of CoinJoin transactions on the Bitcoin blockchain can largely be attributed to more user-friendly wallet applications such as Wasabi and Samurai Wallet. Prior to the development of trustless CoinJoins, trust-required Bitcoin mixers and laundry services such as *Bitcoin Laundry*, *Bitcoin Fog* or *Blender.io* offered services for users to anonymize their funds by routing them through shared addresses [24].

Since the launch of trustless CoinJoin services that were implemented through CoinShuffle, JoinMarket or the wallet applications of Wasabi and Samurai, the focus has more and more shifted towards CoinJoin detection and deanonymization attempts.

Maurer *et al.* [28] conducted an examination of CoinJoin transactions and revealed that the prevalence of common input amounts can potentially allow an adversary to track money flows. Their argument centers around the inadequacy of conventional CoinJoin transactions in providing sufficient anonymity for users looking to transfer arbitrary Bitcoin amounts. In order to address this, they introduce a novel output splitting algorithm designed to enhance the level of anonymity within CoinJoin transactions, without limiting users from transferring arbitrary amounts. Although their algorithm generates satisfactory results, they discovered that multiple transactions could still be susceptible to deanonymization.

Fröwis *et al.* [15] questioned the credibility of commonly adopted heuristics, such as multiple-input heuristics, particularly when applied without considering CoinJoin transactions. They highlighted that CoinJoin transactions could result in inaccurate assumptions stemming from excessively clustered addresses. Their study leans more towards the legal implications that might arise due to incorrect interpretations of user graphs, thereby emphasizing the need to modify the multiple-input heuristic to accommodate the growing number of obfuscating transactions.

Wu *et al.* [29] conducted an analysis of both mixing services and trustless CoinJoin providers, specifically focusing on the patterns of services like Chipmixer, Wasabi Wallet, ShapeShift, and Bitmix.biz. They first identified the obfuscation mechanisms of these services through an experimental analysis of the transaction graph, then applied an algorithm capable of detecting mixing transactions with a 92% accuracy rate. In addition, they provided profitability estimates and demonstrated their approach's effectiveness in tracing stolen Bitcoins. However, their work has limitations, especially with respect to Wasabi CoinJoin transactions. The authors used identified CoinJoins obtained from the Wasabi coordinator API as a seed to identify related transactions containing outputs with the same denomination as the input seed. As a result, their method lacks the ability to detect historical CoinJoins.

Stocking *et al.* [13] presented a comprehensive analysis focusing on Wasabi and Samurai CoinJoins, building upon the CoinJoin detection attempts previously made by Ficsór [14]. They succeeded in enhancing the precision of identifying CoinJoin transactions originating from Wasabi and Samurai Wallet applications, both of which represent the most utilized obfuscation technologies as of 2023. Their refined detection rules form a crucial foundation for our study as we have adopted their methodology, implemented the rules using SQL, and applied the suggested heuristics in the preprocessing phase of our research.

Our methodology incorporates and builds upon the findings of previously cited authors to enhance existing Bitcoin user graph analysis methods, thereby establishing an additional layer of precision within the clustering process. We utilize the well-established multiple-input clustering heuristic in conjunction with a CoinJoin-adjusted transaction graph to ultimately derive our CoinJoin-adjusted user graph. Moreover, we strictly prevent cluster collapses by using publicly known address-labels. By doing so, our aim is to achieve a more comprehensive picture of the topology of the Bitcoin network.

4 Data acquisition

First, a Bitcoin Core client version 0.21.1 was set up to create a local copy of the Bitcoin blockchain as of 14.11.2021. In total, we downloaded 2804 blk files, which together represent the actual blockchain and have a total of about 350 GiB in size. The last block included in our analysis is block number 709,631 with a timestamp of 2021-11-14 04:57:14 UTC. This block has been incorporated in the blk file number 2804. Each individual blk-file contains blocks in binary format and can have a maximum capacity of 128 MiB.

Two approaches were considered for proceeding with the data: One approach involved using the client's integrated JSON RPC and commands like *getrawtransaction* and *getblock* to parse each transaction, as done by Zhao et al. [30]. The second approach involved using open-source software like `python-blockchain-parser`, adjusted to fetch transaction data directly from the locally run Bitcoin Core client.

In this study, the second approach was applied for performance reasons and therefore the `python-blockchain-parser` was adjusted to our

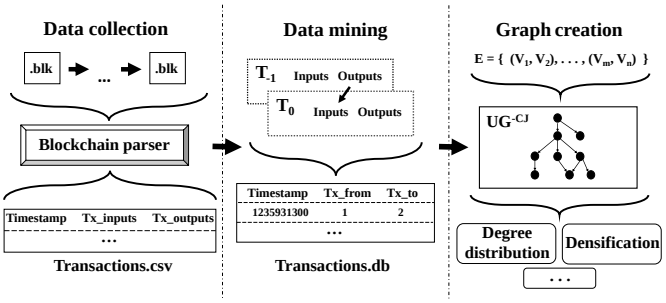

Fig. 3: Description of the data acquisition and analysis workflow

TABLE 2: CUMULATIVE STATISTICS ON BITCOIN

snapshot	#txs	#edges	#nodes
31.12.2009	32,709	36,118	32,611
31.12.2010	218,012	357,980	176,326
31.12.2011	2,119,777	8,699,936	2,768,497
31.12.2012	10,572,827	50,619,652	8,707,387
31.12.2013	30,216,068	165,406,409	24,806,749
31.12.2014	55,479,788	536,722,857	58,880,264
31.12.2015	101,153,811	1,104,704,739	115,225,325
31.12.2016	183,780,434	1,707,653,884	209,678,116
31.12.2017	287,843,663	2,769,993,445	353,414,610
31.12.2018	369,239,299	3,615,499,650	464,829,692
31.12.2019	489,022,946	4,747,563,790	595,594,675
31.12.2020	601,576,444	5,931,290,239	759,260,121
31.12.2021	686,642,867	7,427,788,006	901,545,413

requirements to parse the raw byte files into parquet-format files that were uploaded directly to the Google BigQuery data warehouse, containing the following fields: $Timestamp$, $TransactionID$, $IN_{TransactionID}$, IN_{Vout}, $OUT_{Address}$, OUT_{Index} and $blkfile\text{-}nr$. The inputs of a Bitcoin transaction consist of a transaction ID,$IN_{TransactionID}$, and the index of the respective UTXO within that transaction, IN_{Vout}, whereas the outputs of a Bitcoin transaction are represented by the output address, $OUT_{Address}$, and its index, OUT_{Index}, in the transaction. The $blkfile\text{-}nr$ indicates the location of the respective transaction within the raw binary files gathered from the Bitcoin core client software.

We used SQL to map every $IN_{TransactionID}$ and IN_{Vout} to the respective UTXO, such that the $OUT_{Address}$ of another transaction at the particular OUT_{Index} turns to the $IN_{Address}$ of the transaction that spends its UTXOs. This was described in more detail in Algorithm I. Furthermore, we used SQL to pre-process the data for the final user graph generation.

For performance reasons we mapped every address and transaction ID to an integer that represents its unique ID using SQL. Some measures were directly calculated within Google BigQuery using SQL, while Python was used for more complex calculations. For the final graph, we created a two-columns edge-list of integers that were mapped to timestamps. This list represents the actual graph and contains the edges between the users of our network. We accomplished to do so by implementing an *union-find*-like clustering algorithm that is capable of linking transactions with common input addresses.

Our final data collection contained 7,427,788,006 edges and 686,642,867 transactions. Among these transactions, 99.97 % were valid, with a total of 9,162 transactions that contained invalid output scripts. Another 221,177 transactions incorporated output scripts that could not be decoded to a valid address and 49,485,398 transactions incorporated the script opcode OP_RETURN, which is used to manually designate an output as invalid. Following, coinbase transactions could be identified because their $IN_{TransactionID}$ consists 32 bytes of nulls and 4 bytes containing 0xFF. Until the block with the height 709,630, a total of 709,628 unique coinbase transaction IDs were recorded. The discrepancy of two IDs originates from the fact that the Bitcoin blockchain saw duplicated coinbase transaction IDs twice, constituting an undesirable bug that was fixed with BIP-30 [31]. To maintain consistency, we treated coinbase transactions as originating from the dummy address "0", which served as the $IN_{Address}$ of such transactions. Subsequently, coinbase transactions were excluded to prevent the formation of a graph consisting of a single connected component solely linked by our dummy address.

By filtering out invalid transactions, OP_RETURN and coinbase transactions, we reduced the number of edges within our transaction graph to 7,360,515,878. Furthermore, by excluding CoinJoin transactions, the number of edges was reduced to 7,143,915,409. This was accomplished by first detecting CoinJoin transactions, followed by filtering out their respective edges. The process to detect CoinJoin transactions is further described in the Method section. Doing so, we excluded another 104,177 transactions containing 21,414 Wasabi and 82,763 Samurai CoinJoins. Notably, the CoinJoin adapted data set was only used for the user clustering process. For the final analysis on the CoinJoin-adjusted user graph, each transaction from the cleaned data set (no invalid scripts and coinbase transactions) was considered.

As described formally in the Method section, to build the CoinJoin-adjusted User Graph, we first created a mapping of transaction IDs to input addresses. Every transaction ID within the mapping can be linked to another transaction ID through its input address. This is formally described in Algorithm I, in which we go through each edge e of the transaction graph E_{raw} (line 2) and determine the transaction ID $TxID(e)$ that is specified in the input of the respective edge, $IN_{TxID}(e)$ (line 3). Following, we select the output, $utxo$, at the index $IN_{VOUT}(e)$ (line 4), map its address $OUT_{Address}(utxo)$ to the output of the edge $OUT_{Address}(e)$ (line 5) and add the resulting edge to our matched graph, $E_{matched}$.

Grouping the transaction IDs by input addresses enabled us to eventually create links between the input addresses used in their respective transactions. In total, we encounter 78,601,972 "linking" transactions in our data set. Following, we merge user clusters through assigning shared cluster IDs to the input addresses of the linked transactions.

In the final step of our data acquisition process, we broke the CoinJoin-adjusted transaction graph down into a user graph. The result was a network composed of 415,067,228 clusters connected via 1,820,539,209 edges. This data set forms the basis for the network analysis in the following sections.

To avoid excessive consolidation of clusters, we assem-

Input: Non-mapped Transaction Graph
E_{RAW} = {($TxID$, IN_{TxID}, IN_{Vout}, $OUT_{Address}$ and OUT_{Index}) , . . . }
Output: $E_{matched}$ = { ($IN_{Address}$, $OUT_{Address}$) , . . . }

1: $edges\ E_{matched} \leftarrow empty\ set\ \emptyset$
2: **for all** $e \in E_{RAW}$ **do**
3: $e_{prev} \leftarrow \{e | IN_{TxID}(e) = TxID(e)\}$
4: $utxo \leftarrow e | IN_{Vout}(e) = OUT_{Index}(e_{prev})$
5: $E_{matched}$ += $\{(OUT_{Address}(utxo), OUT_{Address}(e))\}$
6: **end for**
7: **return** $E_{matched}$

Algorithm I: Transaction Input/Output Matching

bled small samples of fewer than 100 addresses affiliated with specific known entities. These entities include exchanges, mining pools, gambling services, darknet market wallets, and other services. For this purpose, we sourced addresses from WalletExplorer.com and gathered information on 389 known wallets. Furthermore, we incorporated the publicly available GraphSense Tag Packs [32], which provided another 325 entities and related descriptions. In total, we collected 126,810 addresses associated with 714 different entities.

To seed our clustering algorithm, we mapped the collected labels to their corresponding addresses, then performed a naive clustering of addresses. The algorithm was designed to ensure no clusters contained two associated entities. This measure was implemented to prevent "cluster collapses" [10] within our graph.

Besides our analysis, we contribute an easy-to-use and scalable Python command-line tool that is capable of parsing raw transaction data (inputs represented by a transaction ID and a vout) while offering multiprocessing integration and maintaining low RAM and disk requirements. Nevertheless, the resulting data set requires the execution of the Input/Output mapping, as described in Algorithm I, at a separate stage.

In addition to our analysis, we have published a Github repository[2] containing Jupyter Notebook files detailing the complete workflow - from chain parsing to the creation of the final tables and graphs. The code has been diligently documented and released to ensure full reproducibility and transparency. Similarly, the SQL queries used have been published. By doing this, we have ensured that our study can be readily reproduced without requiring significant technical expertise.

5 METHOD

In the following study, we leverage quantitative data science techniques to examine the relational Bitcoin network, with particular emphasis on privacy concerns. We describe the evolution of Bitcoin's complex network structure over time and add further granularity to our results by applying a derived input clustering heuristic approach to the transaction graph. Unlike previous studies on Bitcoin's user graph, we omit obfuscating CoinJoin transactions stemming from Wasabi or Samurai wallets, which tend to obscure the data. Additionally, we take measures to prevent the collapse of known clusters. The advantage of our method lies in our treatment of CoinJoin users. Instead of lumping them together, we treat each as a separate entity. This allows us to potentially link distinct CoinJoin users to other clusters in future analyses. Thus, we obtain a more restrained picture of the actual ecosystem. However, our approach also imposes certain limitations on our analysis as we might miss certain CoinJoin money flows that were not merged at a later stage.

Our practical approach can be split into several steps and begins with the preprocessing of the blockchain data which is followed by the creation of the user graph and a quantitative analysis from a network-perspective.

5.1 Data Preprocessing

After removing coinbase transactions from our data set, we employ heuristics from previous studies [13], [14] to identify CoinJoin transactions. Initially, we detect Wasabi Wallet transactions starting from block-height 530,500, which is the first block featuring Wasabi Wallet CoinJoin transactions. This relies on the presence of certain addresses (coordinator addresses) in each transaction. However, Wasabi Wallet altered their procedure by introducing changing coordinator addresses at block 609,999, necessitating an adjustment in our detection method. For Wasabi Wallet, we search for Bitcoin transactions that meet specific conditions:

From block-height 530,500 to 609,999:

1) The transaction output addresses contain one of the following coordinator addresses:
   - bc1qa24tsgchvuxsaccp8vrnkfd85hrcpafg20kmjw
   - bc1qs604c7jv6amk4cxqlnvuxv26hv3e48cds4m0ew
2) The transaction output values contain at least three distinct values

Since block-height 610,000:

1) The transaction output values contain at least 10 equal values
2) The transaction's most frequent equal output value equals 0.1 ± 0.02 BTC
3) The transaction has more inputs than outputs of the most frequent equal value
4) The transaction has at least one unique output value
5) The transaction has at least three distinct output values
6) The transaction uses Bech32 addresses (SegWit)

The conditions (1) - (3) have been carved out by Ficsór [14] while (4) - (6) were proposed by Stockinger *et al.* [13]. Furthermore we apply the following filters that were proposed by Stockinger *et al.* [13] to the resulted data set in order to further reduce the false-positives rate and achieve a precision of roughly 98%:

1) Transactions in which at least one input address is reused as an output address
2) Transactions with output values between 0.08-0.085 and 0.115 - 0.12 BTC or equaling exactly 0.08, 0.09, 0.1, 0.11, or 0.12 BTC

2. https://github.com/Nerolation/wu-bitcoin-clustering-analysis

Next, to identify Samurai Wallet CoinJoins, we examine transactions included in blocks starting from block-height 570,000, which is the first block featuring Samurai CoinJoins. We look for transactions that meet certain conditions:

1) The transaction counts exactly five inputs and outputs
2) The transaction counts exactly one distinct input value
3) The transaction counts at least one and at most three equal input values that exactly match one of Samurai's pool sizes while the remaining input values are between the respective pool size $\pm$ 0.0011 BTC

Additionally, with block-height 658,738, Samurai launched version *v0.99.96e*, which supports pool sizes of 0.001 BTC. As a result, we only applied the above mentioned filters for the new pool size to transactions after the specified block-height. Ultimately, no further processing was performed on the resulting Samurai Wallet data set.

After implementing the aforementioned procedures and identifying the transaction IDs of suspected CoinJoins of Wasabi Wallet and Samurai Wallet, we can finally exclude them from our data set. The resulting data set serves as the starting point for the final user graph.

5.2 Graph Construction

In the following, we will initially provide a formal depiction of Bitcoin's transaction graph, followed by its deconstruction into our user graph that has been modified according to CoinJoin.

5.2.1 Transaction Graph

We represent Bitcoin as a directed, unweighted multigraph designated as $G^{TX} = (V, E)$ where V and E correspond to the sets of vertices and edges that the graph is composed of, respectively. Each edge $e \in E$ stands for an ordered pair of vertices, symbolizing an input and an output node. Consequently, (V_i, V_j) constitutes an edge originating from V_i and ending at V_j such that $(V_i, V_j) \in E$ and $V_i, V_j \subseteq V$. It is crucial to consider the position of a node within an edge since it designates the direction of flow. This implies that $(V_i, V_j) \neq (V_j, V_i)$ for $V_i \neq V_j$. It is worth noting that G^{TX} can incorporate self-loops such that $\exists\ e$ where $V_i = V_j$. Occasionally, these self-loops can be witnessed when the input amount of a transaction surpasses its intended output amount plus the transaction fees, and it is thus utilized as a change address. However, in most instances, a fresh address, which has not previously been involved in the network, is employed for these return transfers. For the upcoming analysis, coinbase transactions were omitted, thus resulting in $|e| = 2$.

To expand upon the provided context, each edge, denoted as e, symbolizes a subset of a transaction, indicated as $tx = e_i, \ldots, e_{i+n}$. The condition that every transaction, tx, has at least one edge is established, i.e., $\forall tx, |E(tx)| \geq 1$, under the assumption that a single edge is necessary to construct a transaction. Moreover, a transaction can be represented as an aggregation of paired and sequentially ordered vertices, formulated as $tx = (v_i,\ v_j), \ldots, (v_{i+k},\ v_{j+l}) \mid k, l \geq 0)$. In this notation, each pair is an element of $E(tx)$, hence $(v_i, \ldots, v_{i+k}) \times (v_j, \ldots, v_{j+l}) \subseteq tx$. Furthermore, the product of k and l equals the cardinality of $E(tx)$, indicated as $k \times l = |E(tx)|$.

Input: $TransactionGraph\ G^{TX},\ updated\_rows \neq 0$
Output: $ClusterIDs\ cid(V)$

```
while updated_rows ≠ 0 do
  updated_rows = 0
  for all tx ∈ TX(G) where |V^IN(tx)| ≥ 2 do
    tx_IN ← V^IN(tx)
    cid_min ← min cid(tx_IN)
    for all v ∈ tx_IN do
      if cid(v) > cid_min then
        cid(v) ← cid_min
        updated_rows += 1
      end if
    end for
  end for
end while
return cid(V)
```

Algorithm II: Address Cluster Mapping

Consequently, the subsets $V^{IN}(tx) = (v_i \ldots v_{i+k})$ and $V^{OUT}(tx) = (v_i \ldots v_{i+k})$ represent the inputs and outputs of a transaction.

Furthermore, every transaction can be identified by its unique transaction ID $id \mid id \in ID$ and since $e \subseteq tx$, edges inherit the transaction IDs of the transactions they are part of.

Additionally, for every edge a UNIX timestamp ts is collected $f : e \mapsto ts$, indicating the exact date when the block in which the respective edges has eventually been included.

As a recall, we already excluded coinbase transactions and edges with OP_RETURN or invalid output scripts in the pre-processing phase of this study to avoid creating a graph that consists of a single giant connected component.

In addition, for the user graph clustering, we filtered out CoinJoin transactions CJs from our data set. This can be formally described with the notation $tx(G^{TX}) \cup CJs = \emptyset$.

5.2.2 User Graph

Breaking down the transaction graph into our CoinJoin-adjusted user graph UG^{-CJ} is an essential step to ensure that our measured network properties are comparable with those of other networks such as social networks [33] or similar digital payment protocols [4].

This means that we treat multiple inputs within a transaction as they originate from the same node. Moreover, by connecting different transactions based on shared input addresses we significantly reduce the number of nodes in our network and add another level of granularity to our graph.

Formally, a transaction $tx_0 \mid tx_0 \in TX$ contains of at least two different transaction inputs $V^{IN}(tx_0) = \{v_i, \ldots v_{i+k}\} \mid k \geq 1$. If a second transaction tx_1 shares a common input address $V^{IN}(tx_0) \cap V^{IN}(tx_1) \neq \emptyset$, then the

tx	*v*	*entity*	*cid*	cid_{min}	*min* cid_{min}
1	11	-	11	11	11
1	12	-	12	11	11
2	12	-	12	12	11
2	13	-	13	12	12
2	14	-	14	12	12
3	15	Mt.Gox	15	15	15
3	16	-	16	15	15
4	16	-	16	16	16
4	17	F2Pool	17	16	16

Fig. 4: Example Workflow of the Clustering Algorithm: Transaction 1 links the nodes 11 and 12. Both then share the cluster ID 11. Transaction 2 combines the nodes 12, 13 and 14. Since node 12 has already been assigned the smaller cid 11, it is not updated. After a second iteration the nodes 11-14 share the cid 11. The vertices at 15 and 17 are associated with known entities. Strict cluster-collapse prevention hinders these clusters from collapsing.

addresses of both transactions are grouped together. Consequently, a third transaction $tx_2 \mid V^{IN}(tx_2) \cap \{V^{IN}(tx_0) \cup V^{IN}(tx_1)\} \neq \emptyset$ would become part of the same cluster, therefore tx_0, tx_1 and tx_2 could be linked.

In order to build the user graph we first assign unique IDs to every (input) address $v \in V$ such that $V \mapsto id$ with $id \in ID$ and create a duplicated mappings of cluster IDs $V \mapsto cid$ with $cid \in ID$. The latter mappings will be updated as described in Algorithm II, such that the ID of the respective address is overwritten by its cluster ID. This way, multiple inputs can be clustered by reconstructing the transaction graph with the updated cluster IDs at a later stage. Therefore, we start by looking for input addresses that might potentially connect different transactions.

As visible in Algorithm II, to create edges between transactions, we initially exclude those that have less than two inputs, $|V^{IN}(tx)| < 2$, and those that do not share input addresses with other transactions such that $\forall(tx_i,\ tx_j) \in TX \times TX \mid V^{IN}(tx_i) \cap V^{IN}(tx_j) = \emptyset$ with $tx_i \neq tx_j$. This way we retrieve a table with two columns that contains transaction IDs and input addresses. Notably this table exclusively includes transaction IDs that can be linked with other transactions through shared input addresses.

In order to cluster transaction IDs, we introduce a novel SQL-based approach to cluster transactions transitively by applying the multiple-input heuristic.

Therefore, we bundle every input address of a transaction that could be connected through shared input addresses with another transaction by assigning the minimum value of all address IDs to every input address in the transactions.

Furthermore, we update the input IDs of transactions that could not be linked through shared input addresses with the minimum ID of all addresses in the respective transactions.

In doing so, we overwrite the cid of each address that was assigned to a cluster by our algorithm. Any address that was not clustered because it does not share transaction inputs with any other addresses kept its ID. After this procedure, the input addresses of multiple transactions that could be linked via a shared input address received a common ID. We describe this process more formally in Algorithm II. This algorithm updates updates the Cluster IDs of an address by sequentially taking the minimum cluster ID of the inputs within a single transaction and overwriting the current cluster ID with this minimum. We stop iterating as soon as no more updates to the underlying mapping are made anymore. This process was further described in Figure 4.

Notably, our proposed algorithm is structured based on the basic idea of the prominent *Union-Find* algorithm and implemented using SQL.

By using the cluster ID of every node $cid(v) \mid v \in V$ to reconstruct the transaction graph and replacing id with cid, we finally retrieve our CoinJoin-adjusted user graph UG^{-CJ}.

For the sake of an objective comparison between the commonly applied multiple-input clustering and our CoinJoin-adjusted variant, we also construct a Non-CoinJoin-adjusted user graph UG. This allows us to draw conclusion on our clustering approach based on the differences we observe between UG and UG^{-CJ}.

5.2.3 Cluster-collapse prevention

In order to prevent our algorithm from merging clusters of different publicly-known entities, we introduce strict cluster-collapse constraints. Cluster collapses occur when the clusters of two independent and publicly known entities get connected through overlapping input addresses.

Using the obtained labels of WalletExplorer.com and GraphSense, we seeded our initial transaction graph and map entities to known addresses. By strictly preventing mergers in our clustering process that could unify two independent clusters of known entities, we can guarantee that UG^{-CJ} does not contain addresses assigned to multiple entities.

5.3 Network Analysis

We apply quantitative network analysis to our CoinJoin-adjusted user graph to contribute up-to-date metrics on the Bitcoin network.

5.3.1 Activity Period

We measure the activity period of the Bitcoin user base to gain information about the period of address-reuse. We determine the activity of a specific address or user by counting the days between an entity's first and last interaction with the network. This can be formally described in the following way:

$$\delta(v) = \max_{v \in V} ts(v) - \min_{v \in V} ts(v)$$

As a reminder, Bitcoin addresses can be used multiple times as long as a user prefers to. Privacy-aware users create new addresses for every transaction to increase privacy. We do not account for activity during the time span between the first and last interaction, however, we strive to obtain results on user longevity and activity distribution.

Furthermore, we define users as a collection of clustered addresses. Thus, we expect users to have greater activity periods than those of individual addresses. The activity period of individual addresses must represent the minimum activity period of its respective cluster.

$$\delta(v_i) \geq \delta(v_j) \mid v_i = v_j,\ v_i \in UG^{-CJ},\ v_j \in G^{TX}$$

Comparing clustered users to individual addresses, we expect the average activity period of entities within UG^{-CJ} to be longer than compared to G^{TX}.

5.3.2 Densification Power Law

The densification power law coefficient is commonly used to empirically determine the evolution of graphs over time [27], [34]. The principle states that the number of edges grows superlinearly with the number of nodes which consequently causes the graph to become denser over time. A network with a high densification power law coefficient grows faster with respect to the number of edges for every newly added node than one with a power law coefficient close to one. Notably, graphs with a coefficient a, where $a > 1$, in general show an increasing average degree over time [34]. Formally, we define this relations as follows:

$$e(t) \propto v(t)^{\alpha} \quad \text{where } 1 < \alpha < 2$$

For this purpose, we determine the number of users $|v(t)|$ and edges $|e(t)|$ at a certain time snapshot t, with t representing the 31th of December of each year from 2009 to 2021 and draw it onto a log-log plot. A putative linear relationship in the resulting chart indicates a super linear relationship between the number of the number of nodes and edges of the graph. As a reference, Leskovec *et al.* [34] studied the popular ArXiv network and determined a coefficient α of 1.68.

In addition we present the densification exponent for smaller timestamps of 3 and 5 years to identify short-term trends and present dynamics. For example, for α_{t5} in 2021, the years 2017-2022 are considered.

5.3.3 Degree Distribution

The degree distribution of the networks captures information about the connectivity of the nodes. We define the in-degree as the number of incoming edges. The out-degree of a node is defined by the number of outgoing edges. Power law distributed degrees have often been observed in literature [26], [35]. Power law distributions indicate scale-free networks in which newly joined nodes preferably connect to high-degree nodes. The preferential attachment process contributes to a strong right-skewed distribution. Scale-free networks can be defined as a fraction of nodes $P(k)$ which decreases exponentially with increasing degrees k, such that

$$P(k) \sim k^{-\gamma}$$

with γ usually in the range $2 < \gamma < 3$. Plotting such distributions onto logarithmic axes unveils the scale-free property, however, the ascertained value of the exponent γ is required to confirm the scale-free hypothesis.

Furthermore, nodes with a degree that far exceeds the average degree of the network often serve a specific purpose and form "hubs". Thus, we it is presumed that the highest-degree entities in the network represent known services such as exchanges or mining pools.

6 Evaluation

In the forthcoming section, we present an evaluation of our proposed clustering method. We compare our CoinJoin-adjusted user graph to a user graph that we created with naive multiple-input clustering and without cluster-collapse prevention. This comparison serves to shed light on the distinctions between both graphs, thereby allowing us to make a definitive assessment of the effectiveness of our proposed approach. Our ultimate objective is to present a quantitative exposition of our rationale for adapting the widely recognized multiple-input heuristic.

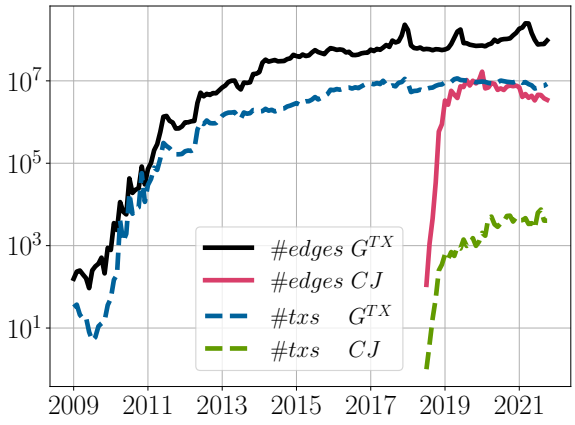

Fig. 5: Number of (CoinJoin) transactions and edges per month

As mentioned earlier, CoinJoin transactions have been around for eight years now, but their integration into user-friendly wallet applications has accelerated their adoption.

6.1 CoinJoins in total

As illustrated in Figure 5, the number of transactions in the Bitcoin network has seen a substantial growth of over 100% in the past six years. Alongside this, the number of trustless CoinJoin transactions experienced a significant surge in 2018, maintaining a consistent upward trajectory through 2021. Given that CoinJoin transactions typically contain a large number of input and output addresses, they represent a considerable portion of edges within the Bitcoin network.

The prominence of CoinJoin transactions, with their extensive number of edges, presents a substantial challenge for network studies focusing on Bitcoin. To illustrate, in January 2020, the edges derived from CoinJoin transactions made up 24.02% of the total number of edges observed, emphasizing the need to consider them appropriately. We observed certain binary block files (namely, blk01919.dat, blk01945.dat, blk01947.dat) in which more than 30% of the edges stemmed from CoinJoin transactions.

The common multiple-input clustering approach would assign the users of CoinJoin transactions, who have participated in the same transaction, to the same cluster. For instance, Wasabi CoinJoin transactions, according to the Wasabi Wallet documentation[3], often involve more than 100 distinct input addresses. Consequently, these addresses would be clustered together.

3. https://docs.wasabiwallet.io/using-wasabi/CoinJoin.html

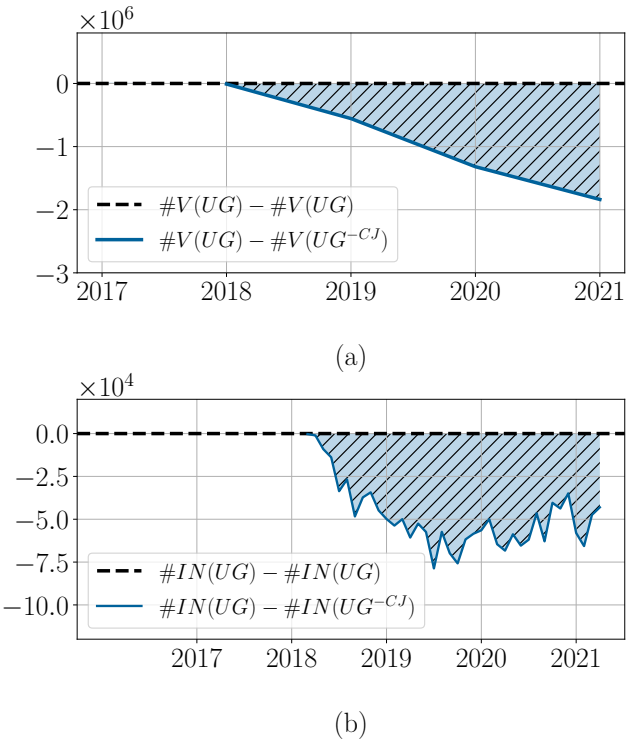

Fig. 6: Active public keys and clustered users per month (a) and the throughput of CoinJoin services per month in BTC (b)

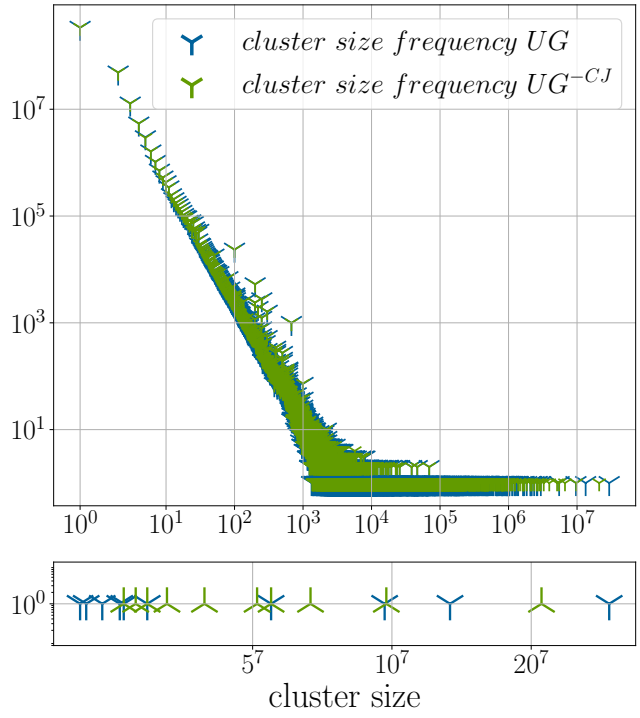

Fig. 7: Cluster sizes plotted onto logarithmic axes.

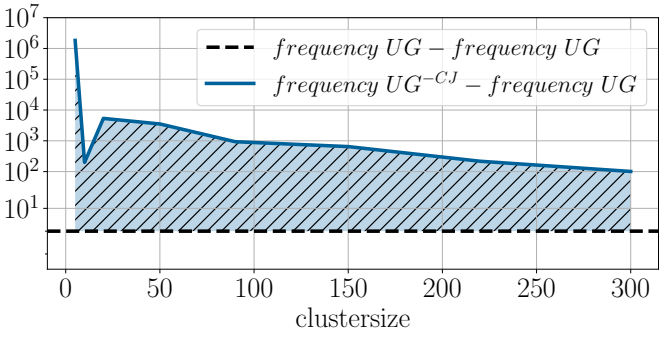

Fig. 8: Comparison of the frequency of clusters with a size of < 300 between UG and UG^{-CJ}

As a consequence, the commonly applied multiple-input heuristic leads to a significantly lower number of users. The difference results from improperly clustered CoinJoin users.

The discrepancy between UG and UG^{-CJ} is visualized in Figure 6 (a). We plot the difference between the cumulative number of users over time. By November 2021, we observe a difference of 1,837,417 users, which results from false-positive clustered addresses. Focusing on the difference in the number of active users per month between UG and UG^{-CJ}, we observe a similar discrepancy. In 2019 there have been months with a spread of more than 75.000 active users.

6.2 Cluster Size

The number of clusters and their size distribution provide an indication of how effectively we've managed to deconstruct the graph from a transactional representation into a user-centric view. We define the size of a cluster by the number of addresses that are associated with it. Furthermore, every address can only be grouped into a single cluster. In total, we observe over 6,000 different cluster sizes. In both graphs, it is common to see clusters made up of a single address, yet, the application of the multiple-input heuristic has allowed many addresses to be grouped together. As demonstrated in Figure 7, the distribution of cluster sizes adheres to a power law.

Focusing on the positive outliers, shown in Figure 7, we find variations in the cluster sizes. The most substantial cluster within UG is composed of 29,485,883 addresses. In comparison, the second-largest cluster has a size of around 13 million addresses. Interestingly, this "supercluster" has already been the subject of investigations in several recent studies [9], [10] that proposed methods to split it into separate chunks of independent entities that were inaccurately consolidated through naive multiple-input clustering. Incorporating our off-chain data , specifically publicly known addresses linked to a known entity, we find that the largest cluster within UG includes 3 of our 714 recognized services. We identify the entities Doubleinvestment.cf, Doublecrypto.ml and an unnamed CoinJoin service provider, all of which are part of this cluster.

The second-largest cluster within UG contains multiple entities. Among these, we identify exchanges such as *Mt. Gox*, *Bitcoin-24.com* or *Laxotrade*, payment providers such as *ePay.info* and mining pools such as *F2pool.io*, *Polmine*, *Eligius.st*. Furthermore, it includes addresses associated with ransomware and the Mt. Gox hack. In total, we find that 15 known entities within UG account for 9,644,446 addresses and include entities such as *CoinPayments.net* and several addresses associated with the hack of the exchange *Binance*.

The largest cluster within UG^{-CJ}, with a size of 21,052,287 addresses, could not be assigned to any known service. Following, the largest cluster within UG counts 8,433,596 addresses more than the largest one within UG^{-CJ}.

The second-largest cluster in UG^{-CJ}, with a size of 9,725,473 addresses, can be associated with *Laxotrade*. Similarly to UG, the third-largest cluster is associated with *CoinPayments.net*. It has a size of 6,668,897 addresses. The differences between the two graphs are visualized in Figure 7.

Focusing on Figure 8, we observe that UG^{-CJ} has more clusters with cluster sizes up to 300 addresses compared to UG. This discrepancy is a direct outcome of the CoinJoin-adjusted clustering method, which divides CoinJoin transactions and considers their inputs as separate entities. As a consequence, Figure 8 depicts a global maximum at a cluster size of 1. Comparing the frequency of cluster sizes up to 100 addresses, we observe a declining difference of a almost 5,000 addresses. When we examine the prevalence of cluster sizes up to 100 addresses, a diminishing difference is observed, with a discrepancy of nearly 5,000 addresses. These users were not clustered with other CoinJoin users but formed their own small-size clusters.

In summary, we conclude that our adapted clustering approach has contributed to the decomposition of certain superclusters. We have eliminated overlaps in cluster affiliation by strictly preventing mergers of known clusters. We treated CoinJoin users as separate entities, which led to significant variances in presented metrics. Our approach allowed us to cluster CoinJoin users beyond CoinJoin transactions. As a consequence, those users who commingled their mixed assets in later transactions formed independent clusters and became vulnerable to deanonymization attacks.

7 Results

In the following section, we present findings retrieved from analyzing UG^{-CJ}. This allows us to leverage our adapted user graph and contribute up-to-date measures. We begin with examining selected network properties. This provides a basis for comparing Bitcoin with other social (non-monetary) networks. We particularly focus on privacy-related measures and their impact on address clustering techniques.

First, we focus on the user activity period. Second, we concentrate on the network's densification process to demonstrate how the accumulated number of nodes and edges within UG^{-CJ} have evolved over time. Finally, we analyze the degree distribution of UG^{-CJ} to determine if the Bitcoin network is scale-free.

7.1 Activity Period

In our study, one key aspect we have measured is the "activity period" of users in the Bitcoin network, a crucial parameter to comprehend the dynamics of user participation. Illustrated in Figure 9, we present the distribution of entities and their associated activity periods, expressed in days. We observe that a significant proportion of addresses have been utilized for merely a single day. In numerical

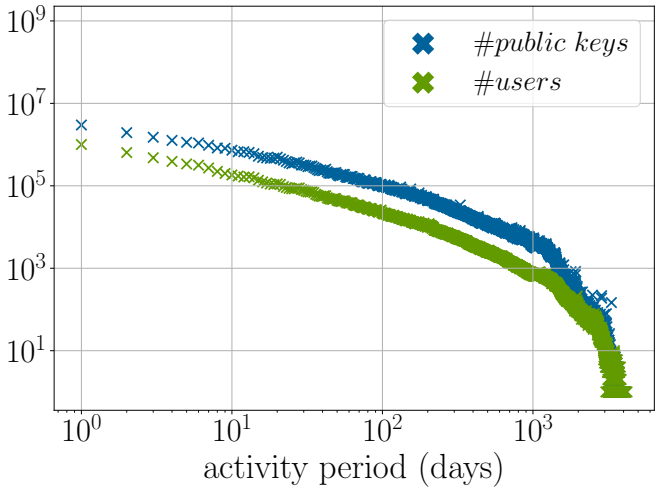

Fig. 9: Activity Distribution

terms, 816,382,701 addresses, which equates to 93.88% of the total addresses in our data set fall into this category. Furthermore, 828,925,894 addresses (95.32%) exhibit usage for a duration of fewer than 10 days. For users within UG^{-CJ}, we discern that 96.57% of these users have had Bitcoin interactions over a single day, while 97.58% show an activity period of less than 10 days.

In contrast, we have discovered that two addresses and eight users have exhibited activity spanning over 11 years. The address/user with the lengthiest activity period can be located at `17THx8KXVDnHM7arcoTxdvmYhS9rP141c2`. This address started using Bitcoin in August 2010 and last interacted with the blockchain in September 2021. Throughout this period, it was involved in a total of 31 transactions. Specifically, it sent or received seven transactions in 2010, an additional seven transactions from February 2011 to 2013, and subsequently remained inactive until 2021, when it recorded eight outgoing transactions. By conducting web searches, we have ascertained that this address was mentioned in July 2010 on a Russian tech forum called *openlife.org* by a user who advocated for Bitcoin and requested test transactions to this address [36].

Among the users with the most extensive lifespan, we have identified several mining pools, namely *Slush Pool* (4,026 days), *PolMine* (3,862 days), *F2Pool* (3,822 days) and *Eligius* (3,586 days). Additionally, we have recognized notable exchanges such as *VirWoX* (3,836 days) *MercadoBitcoin.com.br* (3,760 days), *Bitcoin.de* (3,731) and *Bitstamp* (3,729 days), along with the gambling services *Just-Dice.com* (3,743 days) and *BtcDice.com* (3,695 days).

The average activity period for addresses is 10.2 days, with a standard deviation of 81.9 days. Similarly, users exhibit an observed lifespan of 5.3 days, with a standard deviation of 63.7 days.

Notably, a significant number of addresses with activity periods that far exceed the average could be clustered with similar addresses. Consequently, the average activity period in UG is longer compared to UG^{-CJ}. This discrepancy underscores the impact of clustering on the overall average activity period within the Bitcoin network.

7.2 Densification

As detailed in the methodology section, we have examined the densification procedure of Bitcoin by employing two strategies: First, we have calculated the densification power law coefficient, represented by $\alpha(t)$, for annual snapshots denoted by t. Second, we have constructed a temporal graph by mapping the average out-degree of UG^{-CJ} onto a time axis. Our aim in undertaking these steps is twofold. Primarily, we aim to illuminate the growth dynamics inherent in the Bitcoin network. Furthermore, we aim to formulate a set of standard comparison metrics that will facilitate meaningful and quantitative comparisons between Bitcoin and other diverse networks, including non-monetary social networks or those associated with different cryptocurrencies.

As shown in Figure 10 (a), the Bitcoin network grows with respect to the number of nodes and edges simultaneously. We measure a densification exponent α of 1.0883 (1). This indicates that the number of edges grows superlinearly in relation to the number of nodes. Thus, Bitcoin's densification is not arbitrary but follows a power law pattern. This is further depicted in 10 (b), indicating a super linear relation between the number of nodes and edges. For the Bitcoin network we observe an α that is only slightly greater than 1, indicating a small deviation from linear growth.

Additionally, as visible in Figure 10 (c), the densification exponent α stabilized below 1.1 after 2018, but steadily decreased since 2015. Our observation is consistent with the analysis of Alqassam *et al.* who found a densification exponent of 1.2 with data until 2014.

For the densification exponents in shorter time spans, α_{t3} and α_{t5}, we observe values below 1 after 2015. The smallest value, 0.77, was observed for α_{t3} in 2017. We consider exponents α with $\alpha < 1$ as an indicator for a decreasingly dense network between the years 2013 and 2018.

Focusing on the average out-degree of the Bitcoin graph, as depicted in Figure 10 (d), we observe decreasing trend over the recent years. Decreasing average degrees are indicators of a decreasingly dense network. The average out-degrees went from 7.17 to 4.75 within almost 9 years. This finding is consistent with our observation in Figure 10 (c).

Both measures - the evolution of α and the average out-degree of the Bitcoin network - point to a decreasingly dense network in recent years.

Contrary to Leskovec *et al.* [34], we do not observe an increasing average out-degree, despite having a densification exponent α that is greater than 1. We find that shorter time intervals of 3 and 5 years provide us with more valuable information into short-term densification dynamics.

7.3 Degree Distribution

We analyze the degree distributions of UG^{-CJ} to demonstrate the scale-freeness of the Bitcoin network. In this setting, the in-degree distribution denotes the number of incoming connections a node possesses, which is to say, the count of ingoing edges that originate from incoming transactions to a particular Bitcoin address. Conversely, the out-degree distribution indicates the number of outgoing connections, represented by the number of outgoing edges established by the said address. These measures offer valuable insights into the behavior patterns of users and the

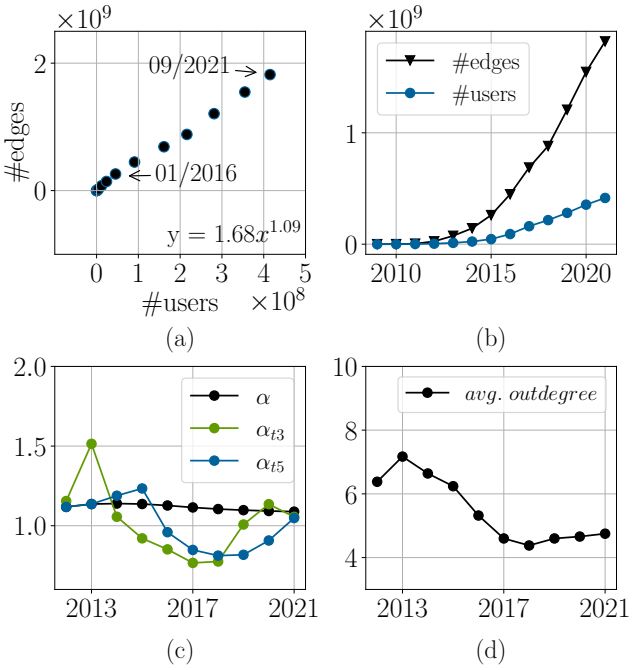

Fig. 10: Network densification over time follows a power law with a densification exponent of 1.09. (left) Absolute number of nodes rose to almost 1 billion with more than 7 billion edges connecting them. (right)

overarching structure of the network. The degree distributions of UG^{-CJ} are depicted in Figure 11. We place particular focus on the analysis of nodes whose degree is substantially higher than the network's average degree. Moreover, for a more nuanced understanding, we juxtapose UG^{-CJ} with UG. This comparative approach underscores the unique features inherent to each and allows us to ascertain key differentiators that set them apart.

As seen in Figure 11, the distributions follow power law behavior and exhibit heavy-tailed patterns. To determine the power law coefficients, we apply a method proposed by Clauser *et al.* [35] that combines a maximum-likelihood method with goodness-of-fit tests based on the Kolmogorov-Smirnov statistic and likelihood ratios. The results reveal that the total degree distribution yields a power law exponent of 1.828. For the in- and out degree distributions, the power law exponents are 2.176 and 1.932, respectively. Thus, we can confidently confirm our observation that the degree distributions follow a power law and the scale-free property of our Bitcoin user graph.

Comparing UG^{-CJ} with UG, we identify unusual curvilinear deviations at degrees close to 100. Intriguingly, there are significantly more users in UG^{-CJ} compared to UG at this juncture. This observed anomaly is a direct consequence of our tailored clustering method. We disassemble Wasabi CoinJoin transactions, which typically have more than 100 outputs. As a result, each individual input is correspondingly linked to every singular output, thereby inflating the count of nodes with degrees approximating 100. Notably, we observe that the deviation is decidedly more marked in the case of the out-degrees, thereby demonstrating the impact of our clustering methodology on the degree distribution characteristics within the Bitcoin network.

In Table 3, we show the degree of those entities with the

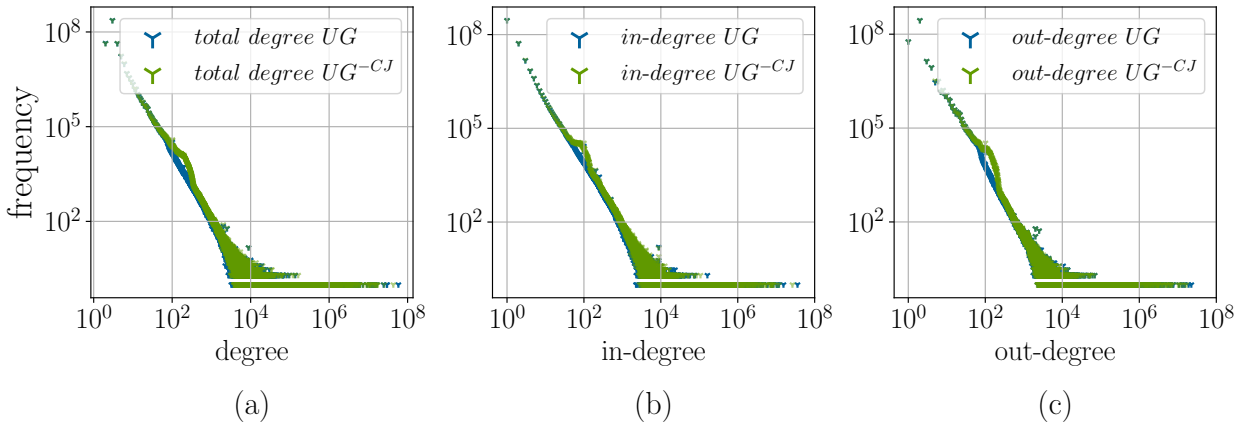

Fig. 11: Degree distribution (a), in-degree distribution (b) and out-degree distribution (c) of UG and UG^{-CJ}.

TABLE 3: USER GRAPH ENTITY COMPARISON

entity	$degree_{UG}$	$degree_{UG^{-CJ}}$	Δ
Doublecrypto.ml	58,249,388	7,524,361	50,725,027
Laxotrade	30,057,625	16,834,191	13,223,434
Sextortion Spam	12,232,121	3,838,573	8,393,548
CoinPayments.net	14,208,576	9,918,948	4,289,628
Bittrex.com	14,102,204	14,242,350	-140,146
Bitcoin.de	2,139,042	4,198,860	-2,059,818
Bitstamp.net	921,016	4,101,274	-3,180,258
Kraken.com	5,965,974	12,002,222	-6,036,248
Huobi.com	220,012	8,964,760	-8,744,748
Binance hack	12,156,537	63,180,075	-51,023,538

greatest discrepancies between UG and UG^{-CJ}. Notably, we observe positive and negative differences. While the applied entity-seeding has led to larger-degree clusters for certain exchanges, we discern a negative difference where wallets associated with fraudulent activities have been segmented due to our cluster-collapse prevention strategy. Specifically, entities such as Doublecrypto.ml, Laxotrade, and Sextortion Spam, which are linked to scams as per the GraphSense database, were subject to this split-up mechanism.

8 Discussion and Limitations

We analyzed Bitcoin's user graph with particular focus on the effects of privacy enhancing techniques on the user graph creation. We conclude that a deterministic exclusion of CoinJoin transactions can prevent exceedingly large clusters that originate from privacy seeking entities.

Our study demonstrates that the commonly used multiple-input heuristic remains largely valid, but requires adaptions to handle privacy-enhancing transactions. We outlined that around 1.8 million nodes would have been falsely clustered without adapting for trustless CoinJoin transactions. This finding confirms our expectation that CoinJoin transactions have a significant impact on address clustering.

To address the issue of inaccurately clustered users of Wasabi and Samurai Wallet, we have excluded the relevant transactions during the user graph creation process. However, this adjustment introduced the possibility of false negatives in our clustering method, as a single user could participate in a CoinJoin transaction multiple times. As a result, we may assign several clusters to such users, which underscores a distinct limitation of our study.

To prevent the formation of superclusters during the clustering process, we strictly avoided collapses of already known clusters. The limitation of this approach is that the actual boundary of a known cluster could be distorted towards another known cluster. Nodes within a transaction that connects two independent clusters might have been incorrectly assigned. This factor influences the size of the known clusters within the network.

In contrast to the naive multi-input clustering, our method may produce a greater number of clusters than those that truly exist in the ecosystem. Therefore, our clustering technique can be viewed as more cautious compared to standard clustering methods. Particularly for law enforcement, these strategies that merge on-chain and off-chain data, like service-name labels, are crucial to preserve a precise understanding of the actual ownership structures within a monetary ecosystem, rather than solely depending on established clustering heuristics. However, this cautious approach may potentially be ineffective in tracking specific individuals who intentionally obfuscate their activities behind CoinJoin transactions, which is a limitation of our study. Such individuals may be inadvertently omitted rather than explicitly targeted, indicating a potential area for further refinement in our methodology.

Utilizing our CoinJoin-adjusted user graph, we conducted an analysis of selected network properties, allowing us to compare the state of the Bitcoin network as of November 2021 to its previous snapshots. While the Bitcoin network fundamentally differs from other social networks across various domains, it shares key properties with them, such as scale-freeness, power law distributed cluster sizes, and super-linear densification [34].

We observed an average user lifespan of 5.3 days in the network, a duration that is remarkably low for a payment network. We conjecture that this could be due to a significant number of users prioritizing privacy and consequently avoiding connections between their wallets.

Moreover, we discovered that the densification process

within the network follows a weak power law pattern, suggesting that the process is not arbitrary. This inference is further supported by the decreasing average out-degree of the network observed since 2013. We deduce that the Bitcoin network was significantly denser in the past compared to its current state. The network's densification process has seen a consistent slowdown over recent years. Low density networks may indicate an increased awareness of privacy within the network. We observed the emergence of large hubs consisting mainly of exchanges and mining pools. Conversely, we observed a marked rise in the number of small-sized clusters, an indication of increased privacy within the network.

Our findings suggest that naive clustering approaches tend to amalgamate entities associated with fraudulent activities into a single cluster. This is anticipated, as fraudulent users employ similar techniques to conceal their activities, ultimately landing in the same network cluster in an attempt to blend into the crowd. Yet, our proposed algorithm capably differentiates distinct entities by overlooking transactions that could otherwise result in cluster collapses. As we look forward, the evolution of Bitcoin network analysis will necessitate a continuous refinement of methodologies to keep pace with the ongoing innovations in the field of blockchain privacy. Developments geared towards reducing the detectability of CoinJoin transactions present future challenges for forensic analysis. These advancements make it increasingly difficult to identify such transactions and effectively exclude them from clustering attempts.

9 Conclusion

In this study, we examined the effect of CoinJoin transactions on blockchain forensic techniques. We demonstrated the potential of CoinJoins to distort the network graph by leading to the collapse of independent clusters. Based on this analysis, we proposed an improvement to the multiple input clustering heuristic. We accounted for CoinJoin transactions in the clustering process and strictly prevent cluster collapses of known clusters. We showed that CoinJoin transactions can be detected and incorporated into the address clustering process by applying specific rules to avoid oversized CoinJoin clusters.

For blockchain analytics, it is essential to obtain an accurate picture of the ecosystem. We contend that on-chain address clustering has become more intricate over time. Traditional clustering techniques, which rely solely on on-chain information and lack external data sources such as known entity labels, yield inaccurate results. We found that a combination of improved clustering techniques and off-chain information yields more effective outcomes. Our findings hold potential for advancing efforts in preventing money laundering and terrorism financing.

Acknowledgment

This research has been supported in part by ASPIRE under the ASPIRE Virtual Research Institute Program, Award Number VRI20-07. ASPIRE is part of the Advanced Technology Research Council located in Abu Dhabi, UAE.

References

[1] H. Li, Q. Chen, H. Zhu, D. Ma, H. Wen, and X. S. Shen, "Privacy leakage via de-anonymization and aggregation in heterogeneous social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 2, pp. 350–362, 2020.

[2] G. Liu, Q. Yang, H. Wang, and A. X. Liu, "Trust assessment in online social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 2, pp. 994–1007, 2021.

[3] V. Karyotis, "A markov random field framework for modeling malware propagation in complex communications networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 4, pp. 551–564, 2019.

[4] Y. Li, G. Yang, W. Susilo, Y. Yu, M. H. Au, and D. Liu, "Traceable monero: Anonymous cryptocurrency with enhanced accountability," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 2, pp. 679–691, 2021.

[5] Y. Jia, S.-F. Sun, Y. Zhang, Q. Zhang, N. Ding, Z. Liu, J. K. Liu, and D. Gu, "pbtpbt: A new privacy-preserving payment protocol for blockchain transactions," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 1, pp. 647–662, 2022.

[6] N. Lu, Y. Chang, W. Shi, and K.-K. Raymond Choo, "Coinlayering: An efficient coin mixing scheme for large scale bitcoin transactions," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2020.

[7] R. Xiao, W. Ren, T. Zhu, and K.-K. R. Choo, "A mixing scheme using a decentralized signature protocol for privacy protection in bitcoin blockchain," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 4, pp. 1793–1803, 2021.

[8] B. Lian, G. Chen, J. Cui, and M. Ma, "Compact e-cash with efficient coin-tracing," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 1, pp. 220–234, 2021.

[9] D. Ermilov, M. Panov, and Y. Yanovich, "Automatic bitcoin address clustering," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2017, pp. 461–466.

[10] M. Möser and A. Narayanan, "Resurrecting Address Clustering in Bitcoin," *arXiv:2107.05749 [cs]*, Jul. 2021, arXiv: 2107.05749. [Online]. Available: http://arxiv.org/abs/2107.05749

[11] J. Crowcroft, D. Di Francesco Maesa, A. Magrini, A. Marino, and L. Ricci, "Leveraging the users graph and trustful transactions for the analysis of bitcoin price," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1338–1352, 2021.

[12] Y. Wu, F. Tao, L. Liu, J. Gu, J. Panneerselvam, R. Zhu, and M. N. Shahzad, "A bitcoin transaction network analytic method for future blockchain forensic investigation," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1230–1241, 2021.

[13] J. Stockinger, B. Haslhofer, P. Moreno-Sanchez, and M. Maffei, "Pinpointing and measuring wasabi and samourai coinjoins in the bitcoin ecosystem," 2021.

[14] Ádám Ficsór, "Dumplings," 2020. [Online]. Available: https://github.com/nopara73/Dumplings

[15] M. Fröwis, T. Gottschalk, B. Haslhofer, C. Rückert, and P. Pesch, "Safeguarding the evidential value of forensic cryptocurrency investigations," *Forensic Science International: Digital Investigation*, vol. 33, p. 200902, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1742287619302567

[16] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System — Satoshi Nakamoto Institute," Tech. Rep., 2008, https://bitcoin.org/bitcoin.pdf [Last accessed 1 Okt 2020].

[17] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, "A fistful of bitcoins: Characterizing payments among men with no names," in *Proceedings of the 2013 Conference on Internet Measurement Conference*, ser. IMC '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 127–140. [Online]. Available: https://doi.org/10.1145/2504730.2504747

[18] M. Ober, S. Katzenbeisser, and K. Hamacher, "Structure and anonymity of the bitcoin transaction graph," *Future Internet*, vol. 5, no. 2, pp. 237–250, 2013. [Online]. Available: https://www.mdpi.com/1999-5903/5/2/237

[19] E. Androulaki, G. O. Karame, M. Roeschlin, T. Scherer, and S. Capkun, "Evaluating User Privacy in Bitcoin," in *Financial Cryptography and Data Security*, A.-R. Sadeghi, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 34–51.

[20] M. Harrigan and C. Fretter, "The unreasonable effectiveness of address clustering," *CoRR*, vol. abs/1605.06369, 2016. [Online]. Available: http://arxiv.org/abs/1605.06369

[21] D. D. F. Maesa, A. Marino, and L. Ricci, "The bow tie structure of the bitcoin users graph," *Applied Network Science*, vol. 4, no. 1, pp. 1–22, 2019.
[22] P. Nerurkar, D. Patel, Y. Busnel, R. Ludinard, S. Kumari, and K. Khan, "Dissecting bitcoin blockchain: Empirical analysis of bitcoin network (2009-2020)," *Journal of Network and Computer Applications*, vol. 177, 11 2020.
[23] G. Maxwell, "CoinJoin: Bitcoin privacy for the real world," Aug. 2013. [Online]. Available: https://bitcointalk.org/index.php?topic=279249.0
[24] M. Möser, R. Böhme, and D. Breuker, "An inquiry into money laundering tools in the bitcoin ecosystem," in *2013 APWG eCrime Researchers Summit*, 2013, pp. 1–14.
[25] Ádám Ficsór, "Zerolink the bitcoin fungibility framework," 2017. [Online]. Available: https://github.com/nopara73/ZeroLink
[26] D. Maesa, A. Marino, and L. Ricci, "Uncovering the Bitcoin Blockchain: An Analysis of the Full Users Graph," 2016, pp. 537–546.
[27] I. Alqassem, I. Rahwan, and D. Svetinovic, "The anti-social system properties: Bitcoin network data analysis," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 1, pp. 21–31, 2020.
[28] F. K. Maurer, T. Neudecker, and M. Florian, "Anonymous coinjoin transactions with arbitrary values," in *2017 IEEE Trustcom/BigDataSE/ICESS*. IEEE, 2017, pp. 522–529.
[29] L. Wu, Y. Hu, Y. Zhou, H. Wang, X. Luo, Z. Wang, F. Zhang, and K. Ren, "Towards understanding and demystifying bitcoin mixing services," in *Proceedings of the Web Conference 2021*, ser. WWW '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 33–44. [Online]. Available: https://doi.org/10.1145/3442381.3449880
[30] C. Zhao and Y. Guan, "A GRAPH-BASED INVESTIGATION OF BITCOIN TRANSACTIONS," in *Advances in Digital Forensics XI*, G. Peterson and S. Shenoi, Eds. Cham: Springer International Publishing, 2015, pp. 79–95.
[31] P. W. , "Bip-30," Feb. 2012. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0030.mediawiki
[32] B. Haslhofer, "Graphsense tagpacks." [Online]. Available: https://github.com/graphsense/graphsense-tagpacks
[33] S. Peng, G. Wang, Y. Zhou, C. Wan, C. Wang, S. Yu, and J. Niu, "An immunization framework for social networks through big data based influence modeling," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 6, pp. 984–995, 2019.
[34] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, pp. 2–es, 2007.
[35] A. Clauset, C. R. Shalizi, and M. E. J. Newman, "Power-law distributions in empirical data," *SIAM Review*, vol. 51, no. 4, pp. 661–703, Nov. 2009, arXiv: 0706.1062. [Online]. Available: http://arxiv.org/abs/0706.1062
[36] Jedi-to-be, "Openlife/open source/bitcoin." [Online]. Available: http://open-life.org/blog/opensource/1181.html

Anton Wahrstätter received his B.Sc. in Economics and LL.B. in Business Law from the University of Innsbruck, Austria, in 2018 and 2019, respectively. He completed his M.Sc. in Digital Business from the University of Innsbruck, Austria, in 2020 and an M.Sc. in Blockchain and Digital Currency from the University of Nicosia, Cyprus, in 2022. He has contributed to the Bitcoin and Ethereum community as an open-source developer since 2016, primarily focusing on privacy and quantitative data analysis. He is affiliated with the Institute for Distributed Ledgers and Token Economy at the Vienna University of Economics and Business, and the Research Institute for Cryptoeconomics in Vienna, Austria. His research focuses on blockchain privacy and trust and the application of data science techniques to blockchain data.

Alfred Taudes Alfred Taudes is a Professor at the Department of Information Systems and Operations Management, Institute for Production Management at Vienna University of Economics and Business (WU Vienna). He holds a doctoral degree in Business Administration from WU Vienna, a Magister Degree in Management Information Systems from Vienna University and a PhD (Habilitation) from WU Vienna in both fields. After assistant professorships at WU in the areas Operations Research and Applied Informatics he was visiting professor at Universities Augsburg, Münster and Essen and joined WU in 1993. There he coordinated large scale research projects such as the WWTF-project "Integrated Demand and Supply Chain Management" and the special research area Adaptive Models in Economics and Management Science. From 2010 to 2016 he served as head of Department of Information Systems and Operations. He has also been teaching at Kobe and Tsukuba University, Japan. He is coordinating WU's research group on Cryptoeconomics and is head of the scientific board of the Austrian Internet Offensive. Alfred has undertaken research in the interface of Operations and Supply Chain Management, Marketing Engineering and Knowledge Management. He has published more than 150 articles, among others, in international top journals like Management Science, European Journal of Operational Research, International Journal of Production Economics, Marketing Science, and MIS Quarterly. His current research deals with the impact of Big Data and Blockchains on the design of production systems. Using Complexity Science and Cryptoeconomics he is studying integrated value chains and market designs for digital production.

Davor Svetinovic (SM'16) is a professor of computer science at the Department of Computer Science, Khalifa University, Abu Dhabi, and the Department of Information Systems and Operations Management, Vienna University of Economics and Business, Austria (on leave), where he is the head of the Institute for Distributed Ledgers and Token Economy, and the Research Institute for Cryptoeconomics. He received his doctorate in computer science from the University of Waterloo, Waterloo, ON, Canada, in 2006. Previously, he worked at TU Wien, Austria, and Lero – the Irish Software Engineering Center, Ireland. He was a visiting professor and a research affiliate at MIT and MIT Media Lab, MIT, USA. Davor has extensive experience working on complex multidisciplinary research projects. He has published over a hundred papers in leading journals and conferences and is a highly cited researcher in blockchain technology. His research interests include cybersecurity, blockchain technology, cryptoeconomics, trust, and software engineering. His career has furthered his interest and expertise in developing advanced research capabilities and institutions in emerging economies. He is a Senior Member of IEEE and ACM (Lifetime) and an affiliate member of the Mohammed Bin Rashid Academy of Scientists.