

Article

Not peer-reviewed version

---

# A Similarity Measure for Linking CoinJoin Output Spenders

---

[Michael Herbert Ziegler](#)\*, [Mariusz Nowostawski](#), [Basel Katt](#)

Posted Date: 18 September 2025

doi: 10.20944/preprints202509.1550.v1

Keywords: blockchain; coinjoin; privacy; bitcoin; similarity






Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

## Article

# A Similarity Measure for Linking CoinJoin Output Spenders

Michael Herbert Ziegler \* , Mariusz Nowostawski  and Basel Katt 

Faculty of Information Technology and Electrical Engineering, Norwegian University of Science and Technology, Teknologivegen 22, 2815 Gjøvik, Norway

\* Correspondence: michael.h.ziegler@ntnu.no

## Abstract

This paper introduces a novel similarity measure to link transactions which spend outputs of CoinJoin transactions, CoinJoin Spending Transactions (CSTs), by analyzing their on-chain properties, addressing the challenge of preserving user privacy in blockchain systems. Despite the adoption of privacy-enhancing techniques like CoinJoin, users remain vulnerable to transaction linkage through shared output patterns. The proposed method leverages timestamp analysis of mixed outputs and employs a one-sided Chamfer distance to quantify similarities between CSTs, enabling the identification of transactions associated with the same user. The approach is evaluated across three major CoinJoin implementations (Dash, Whirlpool, and Wasabi 2.0) demonstrating its effectiveness in detecting linked CSTs. Additionally, the work improves transaction classification rules for Wasabi 2.0 by introducing criteria for uncommon denomination outputs, reducing false positives. Results show that multiple CSTs spending shared CoinJoin outputs are prevalent, highlighting the practical significance of the similarity measure. The findings underscore the ongoing privacy risks posed by transaction linkage, even within privacy-focused protocols. This work contributes to the understanding of CoinJoin's limitations and offers insights for developing more robust privacy mechanisms in decentralized systems. To the authors knowledge this is the first work analyzing the linkage between CSTs.

**Keywords:** blockchain; coinjoin; privacy; bitcoin; similarity

## 1. Introduction

Blockchain systems are designed to be transparent to allow public verification of transactions and the asset's traceability. This transparency, while beneficial for ensuring trust and accountability, poses significant challenges to the privacy of its users. In response to these challenges, various techniques have emerged to improve privacy within blockchain systems. In particular, in blockchain systems which use the *Unspent Transaction Output* (UTXO) model, one such technique is CoinJoin, a method that enables users to collaboratively build their transactions, thereby obscuring the flow of funds and achieving a degree of privacy through mixed outputs.

Despite the diversity in CoinJoin implementations, which range from number of mixing rounds to used amount denominations, the fundamental outcome remains consistent: the creation of transaction outputs that are hard to link to their respective unmixed input. Once the coins are sufficiently mixed, users spend these outputs outside the CoinJoin transaction graph, engaging in what we call *CoinJoin Spending Transactions* (CSTs). Notably, users do not typically spend all their mixed outputs in a single transaction; rather, they tend to create multiple transactions over time, often involving several mixing sessions.

Linking transactions of users can significantly diminish users' privacy, as their previously fragmented on-chain activity can be connected. By participating in a CoinJoin process and subsequently creating a CST, a user carefully tries to obscure their on-chain activity. Therefore, it is especially detrimental if multiple CSTs can be linked, as this would connect on-chain activity. For example, [1,2]

present heuristics for identifying the possible senders of CoinJoin outputs. If several CSTs can be linked using our proposed similarity measure, the heuristics can be applied to a larger set of outputs, thereby increasing the likelihood of successfully tracing the source of the mixed outputs.

In this paper, our aim is to answer **how transactions which spend outputs of CoinJoin transactions can be linked together based on their on-chain properties?** To this end, we propose a novel similarity measure for analyzing CSTs. We demonstrate that the user behavior this measure relies upon is prevalent across the evaluated CoinJoin implementations (Dash, Whirlpool and Wasabi 2.0). Furthermore, we improve the existing transaction classification rules for Wasabi 2.0 and evaluate their effectiveness.

In Section 2 we introduce the necessary background to present the improved classification rule in Section 3 and to discuss the similarity measure in Section 4. Section 5 concludes the paper with potential research avenues.

### 1.1. Related Work

Finding similarities between transactions in a blockchain system is a known research topic. It allows one to detect common transaction patterns, spending behaviors, and group addresses as belonging to services or to a single user. The techniques range from examination of peer-to-peer network traffic [3] to network analysis techniques applied to transactions [4] and machine learning approaches [5]. The authors of [6] use taint analysis of outputs to fingerprint on-chain behavior and link transactions to entities. In [7] the authors are able to attribute transactions, by recognizing statistical significant transaction features, to a central mixing service. CoinJoin fee payments were able to be used to track the mixing activity in [8] by detecting and tracing the start of a Dash CoinJoin process.

Although these works examine the linkability of blockchain transactions, we focus specifically on finding similarities between spending transactions of CoinJoin outputs. We achieve this without a limitation to a specific CoinJoin implementation. Additionally, the proposed similarity measure provides a degree of similarity. To the authors' knowledge, this is the first work to propose such a measure.

Goldfeder et al. [9] propose a cluster intersection attack, where starting with a set amount of outputs owned by a single user, they find related address clusters. The authors note the importance of the time distance between mixed outputs. This relates strongly to our proposed CoinJoin transaction timestamp analysis, where we use the timestamps of spent mixed outputs to determine the uniqueness of a spending transaction.

We build on previous work by Stütz et al. [10] and Schnoering and Vazirgiannis [11], for our improvement to Wasabi 2.0 CoinJoin transaction classification methods.

### 1.2. Scope

In this work, we only consider on-chain data, which refers to data stored in the respective blockchain systems. The behavior and data of wallets or CoinJoin software are not considered, and only their resulting transactions on the blockchain are examined. Additionally, the methods we discuss only apply to blockchain systems using the UTXO model. The similarity measure relies on the information leak which occurs when multiple outputs are spent in a single transaction. Therefore, the measure can not be applied to account-based blockchain system. We limit our examination of CoinJoin implementations to Dash's native CoinJoin protocol and Bitcoin's Whirlpool and Wasabi 2.0 protocols.

## 2. Background

### 2.1. Blockchain

A blockchain constitutes a decentralized database system that is replicated on a network of peers. Introduced by Nakamoto [12] in 2008, it was designed to facilitate a decentralized digital currency known as Bitcoin, allowing peer-to-peer interactions without reliance on a central trusted authority.

In blockchain data are not updated directly. Instead, a series of change records, termed transactions, are aggregated and appended to the database in units referred to as blocks. These transactions apply modifications to the state of the blockchain. Each participating peer possesses the capability to validate these state transitions and must achieve consensus regarding the state maintained within the distributed database. Consequently, a blockchain functions as an immutable ledger of state changes instigated by the transactions of its peers, extending to the most recent block in the chain.

The Bitcoin blockchain system is designed primarily to facilitate value transfers between its users. However, its capabilities are not limited to this function. Any type of state can be recorded within the decentralized database, the primary limitations being the memory capacity and processing power of the participating peers. Since blockchain systems utilize cryptographic techniques to ensure security, currencies that operate on these systems are referred to as cryptocurrencies.

Transactions in blockchain systems which use the UTXO account model are composed of inputs and outputs. A transaction spends inputs and creates new outputs. Newly created outputs can become inputs to other transactions. Each output is associated with a script that defines how it can be spent. Typically, each output is linked to a blockchain address (see Section 2.2).

## 2.2. Address Clustering

In blockchain systems, identities are represented by cryptographic addresses generated from the public keys of cryptographic key pairs. Ownership can be verified, since transaction outputs are signed by their creators. Each user in these systems can own multiple addresses, with their corresponding private and public keys securely stored in their wallets. Therefore, an address acts as a pseudonym in the blockchain systems for its user.

Many blockchain systems are transparent by design. They allow to examine all transactions which occurred in their database. This includes Bitcoin and Dash. This transparency allows us to follow the flow of funds of all users via their respective addresses.

Address clustering techniques allow blockchain analysts to identify connections between addresses and their users. One such technique, known as the multi-input heuristic, operates on the assumption that every transaction is created by a single user. This method links all input addresses associated with a transaction (cf. [13]).

## 2.3. CoinJoin

As described in the previous section, Bitcoin does not offer a high degree of privacy. All on-chain data is publicly visible and value movements can be easily observed. Additionally, address clustering further decreases the privacy of users.

In UTXO-based blockchain systems, transactions can be built collaboratively. This was highlighted by Maxwell [14], who proposed that users could create a single transaction by contributing their individual input and output pairs before the transaction was published. This aggregation of outputs or coins is referred to as a *CoinJoin* or *CoinJoin transaction*. When such transactions are executed in succession, they further obfuscate the ownership of the original funds, a process known as mixing. As a result, CoinJoin transactions cannot be clustered using the multi-input address clustering heuristic, as the foundational assumption of this heuristic is no longer met.

There are multiple CoinJoin implementations. This work considers the native implementation of CoinJoin of the Dash blockchain and the Wasabi 2.0 and Whirlpool CoinJoin implementation on the Bitcoin blockchain.

To identify CoinJoin transaction spenders, the actual CoinJoin transactions themselves first have to be identified. This is done via transaction classification, where all transactions of a blockchain system are checked for various conditions which, in combination, should only apply to CoinJoin transactions.

Depending on the CoinJoin implementation, the transaction format follows a common structure. Possible characteristics are as follows:

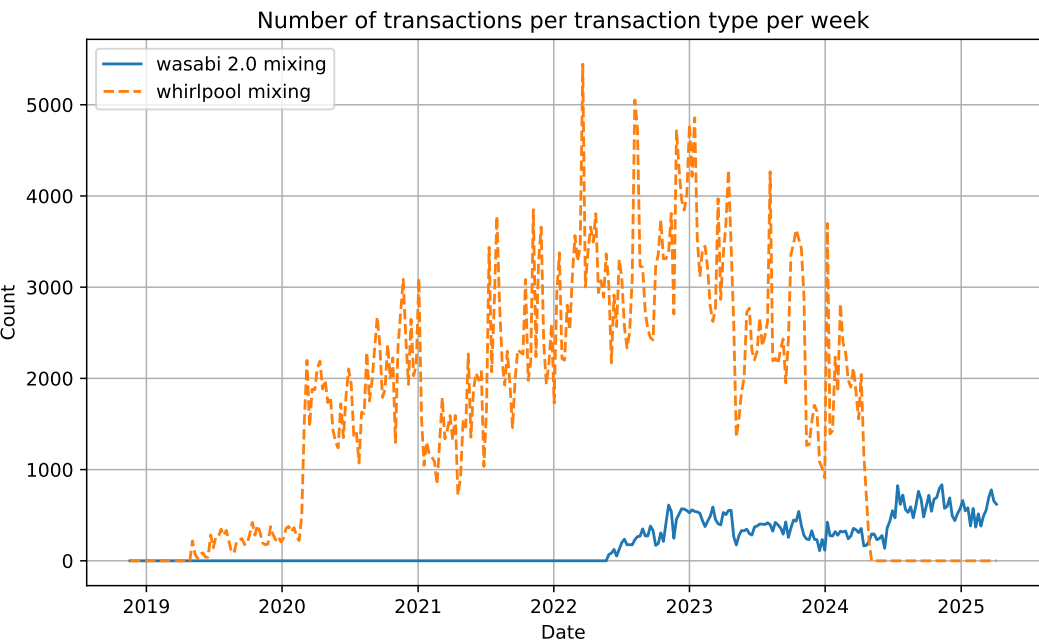
- The transaction uses outputs within a defined range of denominated amounts.

- The transaction creates outputs within a defined range of denominations amounts
- CoinJoin transactions have a set amount of input and outputs
- Payment of the CoinJoin and transaction fee

Transaction classification data presented in the following is the result of our classifications.

2.4. Whirlpool CoinJoin

The Whirlpool CoinJoin service was offered by the Samurai and Sparrow wallets. After the arrest of the Samurai wallet developers<sup>1</sup> in April 2024 and the shutdown of the public coordinator servers (which offered the mixing service), Whirlpool CoinJoin transactions stopped being created on the blockchain. The sudden drop in the count of Whirlpool CoinJoin transactions can be observed in Figure 1 and aligns well with the arrest of developers.



**Figure 1.** Number of Wasabi 2.0 and Whirlpool transactions between 2019 and 2025 per week. Wasabi 2.0 produces more outputs than Whirlpool, indicating higher usage (cf. [15]).

To participate in a Whirlpool CoinJoin, the outputs must be split into Whirlpool denominations. This is done via a Tx0 transaction that creates denominated outputs (including a mixing fee), pays a coordinator fee, and optionally creates a change output. A Whirlpool CoinJoin transaction then consumes the denominated outputs from the Tx0 transactions and other Whirlpool CoinJoin transactions and creates new denominated outputs (cf. [16]).

Appendix B contains the classification rules that we used to detect Whirlpool CoinJoin transactions.

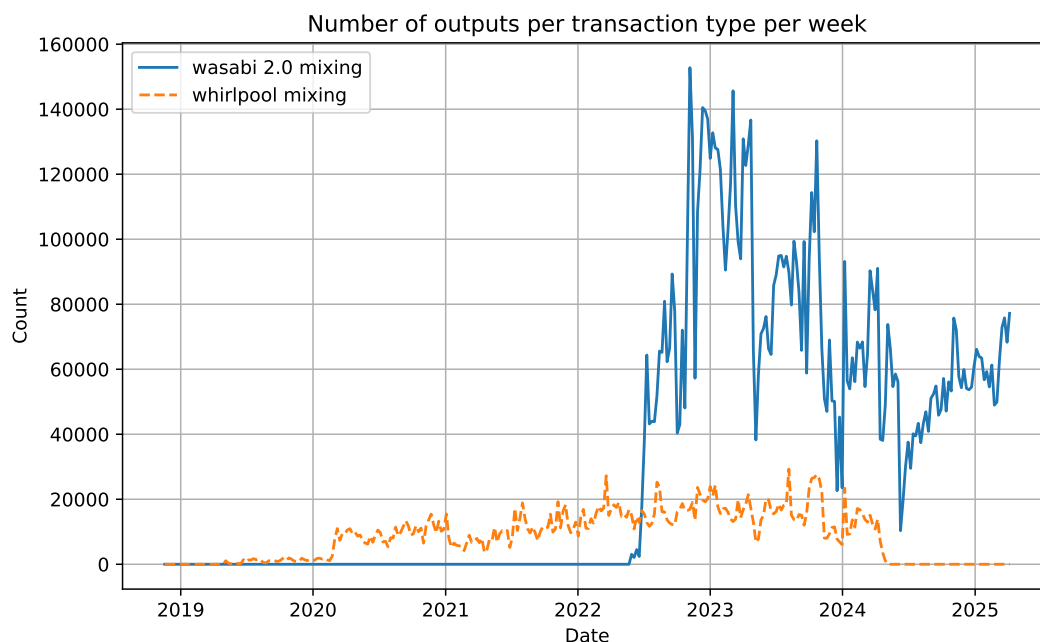
2.5. Wasabi 2.0 CoinJoin

Wasabi 2.0 is an iteration of the Wasabi 1.0 and Wasabi 1.1 CoinJoin implementation, which is offered through Wasabi wallet (cf. [17]). Unlike with the Whirlpool CoinJoin implementation, the outputs can be directly used in Wasabi 2.0 CoinJoin transactions, they do not have to be split into denominations in a separate transaction. A Wasabi 2.0 CoinJoin transaction consumes arbitrary inputs (down to a minimum value) and creates denominated outputs and optionally change outputs. Wasabi 2.0 CoinJoin transactions have characteristically a high number of inputs and outputs. Figure 1

<sup>1</sup> <https://www.justice.gov/usao-sdny/pr/founders-and-ceo-cryptocurrency-mixing-service-arrested-and-charged-money-laundering> (Accessed 11.06.2025)



shows the number of Wasabi 2.0 CoinJoin transactions compared to Whirlpool CoinJoin transactions. The graph indicates a higher activity of Whirlpool usage compared to Wasabi 2.0 usage. However, if the number of outputs is a measure of user activity, Figure 2 shows a much higher amount of outputs being created by Wasabi 2.0 CoinJoin transactions as compared to Whirlpool CoinJoin transactions. This is due to Whirlpool CoinJoin transactions being limited to 8 outputs per transactions, as opposed to Wasabi 2.0 CoinJoin transactions where the number of outputs is much larger.



**Figure 2.** Number of Wasabi 2.0 and Whirlpool CoinJoin outputs between 2019 and 2025 per week.

See Appendix C for the classification rules we used to detect Wasabi 2.0 CoinJoin transactions.

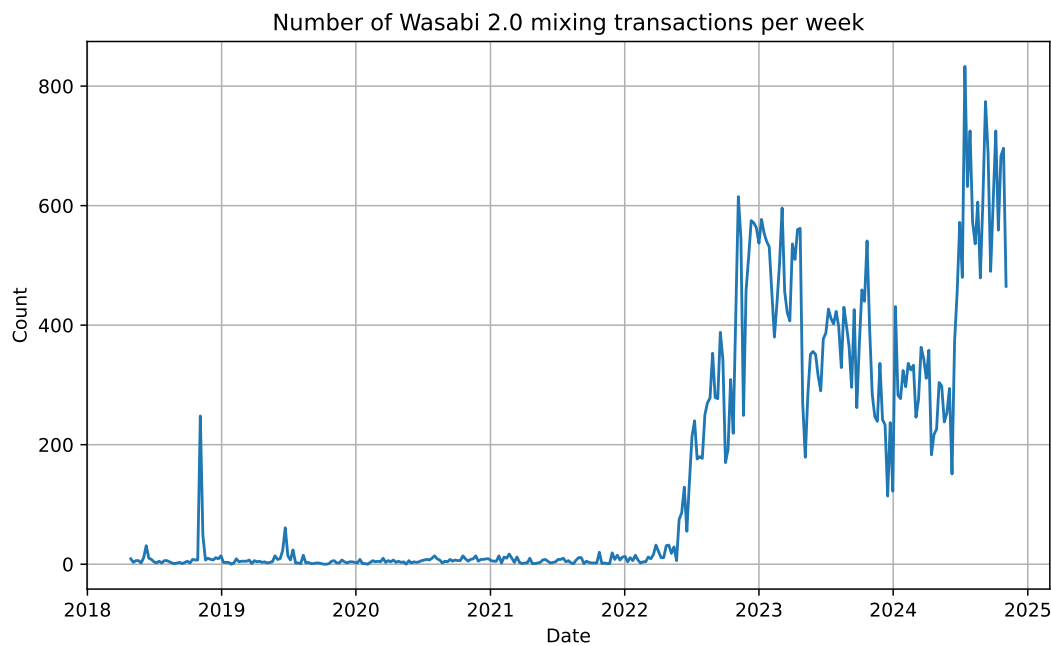
### 2.6. Dash CoinJoin

Similarly to the Bitcoin blockchain system, the Dash blockchain system also uses the UTXO account model. Furthermore, it features a master node system, consisting of privileged nodes controlled by users who stake their coins. The master nodes offer various services to users, one of which is a CoinJoin service. The CoinJoin implementation is therefore native to the Dash protocol. As it is native, it is able to have CoinJoin transactions with a transaction fee of zero. The transaction fee is paid through a separate collateral system (cf. [18]).

Analogously to the Whirlpool CoinJoin implementation, outputs have to be split into denominations before they can be spent in a Dash CoinJoin transaction. This is achieved in Dash via a *Create Denominations* transaction. Appendix A contains the classification rules that we used to detect Dash CoinJoin transactions.

## 3. Improving Wasabi 2.0 Classification

While implementing the Wasabi 2.0 classifier, as described in [11] we found that several transactions are misclassified as Wasabi 2.0 transactions: Wasabi 2.0 was released in June 2022, but transactions classified as Wasabi 2.0 CoinJoin transactions appeared well before that (see Figure 3). Although some Wasabi 2.0 transactions are expected to appear before the official release date, e.g. due to developers' testing, misclassified transactions appear long before Wasabi 2.0 development was even started. Naive classifiers could simply exclude Wasabi 2.0 transactions before the release date, but the number of misclassified transactions before the release date also indicates misclassified transactions after the release date.



**Figure 3.** Number of Wasabi 2.0 CoinJoin transactions per week, without applying the proposed rule demonstrates that a number of transactions (1750) are misclassified as they appear before the Wasabi 2.0 release in June 2022.

The examination of false positives indicated that some transactions meet the denomination criteria because Wasabi 2.0 defines some denominations that are common to be used by users not using the Wasabi wallet (multiples of 5000 Satoshi).

The new rule we are proposing is that a Wasabi 2.0 transaction must have at least one output which has an uncommon denomination (not a multiple of 5000 Satoshi). This is done for the following reasons:

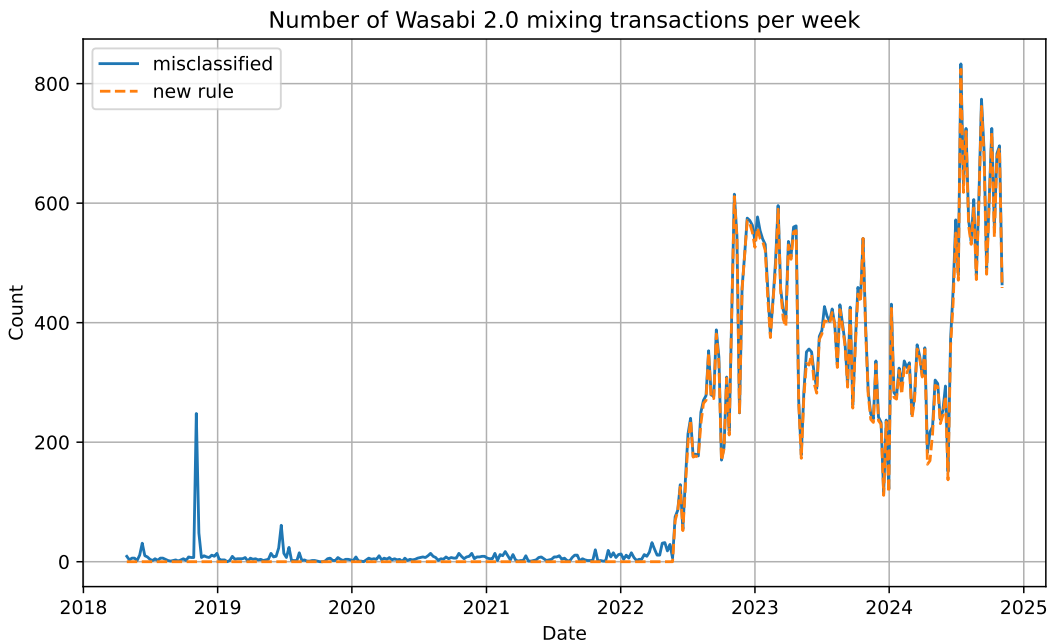
- The output denomination selection in Wasabi 2.0 is based on the provided input amounts. Given that the input amounts have to be split in several denominations, the likelihood is low that only common denominations will be selected is low.
- Given a random selection of the output denomination, with a 33% chance for each output to be of a common denomination, a transaction with 20 denominated outputs has a chance of 0.000000029% to only contain common denomination. Wasabi 2.0 CoinJoin transactions have a high number of outputs (236 on average based on our findings). Therefore, the likelihood that a Wasabi 2.0 transaction only has outputs of common denominations is low.

Data for which transactions are truly Wasabi 2.0 CoinJoin transactions are not available. However, transactions created before the release date of Wasabi 2.0 are likely misclassifications, thus we can evaluate the proposed rule via the following requirements:

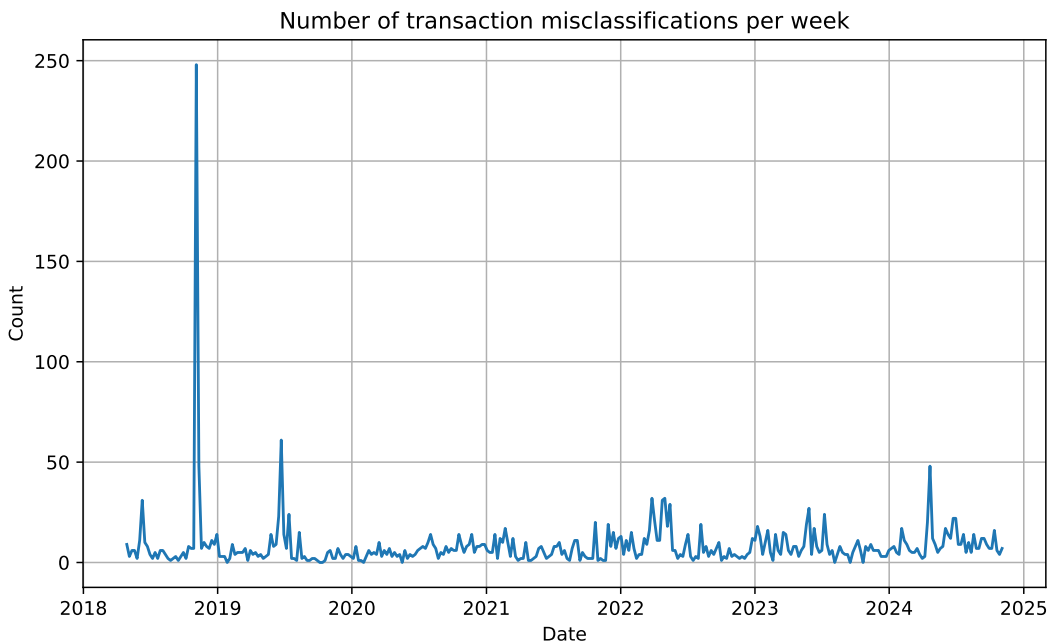
1. The proposed rule should significantly reduce the number of transactions before the Wasabi 2.0 release date (reducing false positives).
2. The proposed rule should not reduce the number of transactions after the Wasabi 2.0 release date by more transactions over time than before the release date. This is because the entities responsible for creating transactions, which are misclassified without the proposed rule, are likely to continue to generate such transactions on an ongoing basis. In other words, the number of transactions detected to be misclassified should be steady over time (low number of false positives).

Figure 4 shows that with the proposed rule applied, no transactions before the release date of Wasabi 2.0 are classified as Wasabi 2.0 CoinJoin transactions, therefore fulfilling requirement 1. The

graph Figure 5 intuitively shows that the number of misclassifications detected is steady even after the release date of Wasabi 2.0. We test the time series for stationarity with the Augmented Dickey-Fuller (ADF) test (cf. [19]). Where a test result of -3.449 would mean that the time series is likely stationary, the tested time series has a score of -14.90, which makes it very likely to be stationary. This corresponds to a very low p-value of  $1.50 \times 10^{-27}$ . Therefore, with the ADF test showing that the time series is stationary, requirement 2 is fulfilled.



**Figure 4.** Number of transactions per week, showing the number of Wasabi 2.0 CoinJoin transactions with (orange line) and without (blue line) the proposed rule. Without the new rule we count 50,830 transactions, while the new rule reduces it by 5.45% to 48,059 transactions. The majority of the reduction occurs before the release date of Wasabi 2.0.



**Figure 5.** Number of transactions per week that have been identified by the proposed classification rule as misclassified Wasabi 2.0 transactions.



With both requirements met, we conclude that the proposed rule is effective in excluding false positive Wasabi 2.0 transactions, and are also confident that the resulting number of false negatives is low.

#### 4. Finding Similar CoinJoin Spenders

In this section, we outline the methodology used to model CoinJoin Spending Transactions (CSTs) for comparative analysis. We present the rationale for our selection of the similarity measure utilized in this study. Furthermore, we detail the procedures performed to compile data sets containing CSTs organized by shared receiver address clusters, for Dash, Whirlpool, and Wasabi 2.0. This data set was instrumental in assessing the prevalence of common spending outputs, thus evaluating the practical significance of our proposed similarity measure.

##### 4.1. Similarity Measure

We define a CoinJoin spending transaction as a transaction that spends at least one output from a CoinJoin transaction, while not being a CoinJoin transaction itself. A CST does not depend on a particular CoinJoin implementation, rather it is the result of a common user action. Users create CSTs when they want to use their mixed outputs. This can be to send money to other addresses or to start another mixing session. It is common for CSTs to spend mixed outputs from multiple distinct CoinJoin transactions. Often these CoinJoin transactions have also been created in distinct time frames. This can be attributed to a user participating in multiple mixing processes in different points of time. For example, a user might mix all their coins in time frame  $tf_1$ , receive new coins afterwards which then get mixed by the same user in time frame  $tf_2$ . If the user then creates a CST, which spends from both time frames, the distinct time frames are visible via the directly connected CoinJoin transactions.

Users spending mixed funds via a CST not necessarily spend all of their mixed funds. Therefore, if a user creates a CST they might spend outputs of the same mixing process as a prior CST did. Keeping with the previous example, a user can create multiple CSTs which use outputs from both  $tf_1$  and  $tf_2$ . This behavior paired with the possibility that CSTs spend outputs from multiple mixing processes enables us to create a CST similarity measure which allows identifying CSTs associated with the same user based on an initial CST. The greater the variety of inputs from different time frames that a CST utilizes, the more distinctive it becomes.

The proposed similarity measure examines when the CoinJoin outputs, which a CST spends, have been created and compares these timestamps with the input timestamps of all other CSTs in the blockchain. Let

$$T = \{t_1, t_2, \dots, t_n\}$$

denote the set of all transactions part of a blockchain system, where  $n$  is the number of transactions.

$$P_k = \{p_{k1}, p_{k2}, \dots, p_{km}\}$$

is the set of input timestamps of a transaction  $t_k$ , where  $k \in \{1, 2, \dots, n\}$  and  $m$  is the number of inputs of transaction  $t_k$ . The input timestamps for each input are the timestamp of its originating transaction. Furthermore, certain inputs originate from CoinJoin transactions, we denote this subset of input timestamps as follows

$$C_k \subseteq P_k$$

and its elements as

$$C_k = \{c_{k1}, c_{k2}, \dots, c_{kl}\}$$

Where  $l$  is the number of CoinJoin inputs of the transaction  $t_k$ .

We model the input timestamps of each CST as 1-dimensional point clouds. A point cloud is a set of discrete points in a given space. Each input timestamp of a CST is treated as one point of the transaction's point cloud. In the following  $t_a$  and  $t_b$  are two CSTs which are compared, while  $t_a$  is the

CST of interest for which we want to find similar CSTs.  $C_a$  and  $C_b$  are their respective sets of CoinJoin input timestamps. The distance measure, in the Euclidean 1-dimensional space, should fulfill the following requirements.

- Not sensitive to aberrations: timestamps of  $C_b$  that have a large Euclidean distance to timestamps of  $C_a$  should not influence the distance measure, if points with a low Euclidean distance to timestamps of  $C_a$  are present. Input timestamps are in some cases years apart; therefore, a similarity measure sensitive to such aberrations would skew the result.
- Set size: the distance measure should be able to compare sets with unequal lengths<sup>2</sup>. Input counts of transactions in blockchain systems can range from a single input to several hundred inputs.

We evaluated the following distance measures for our use case:

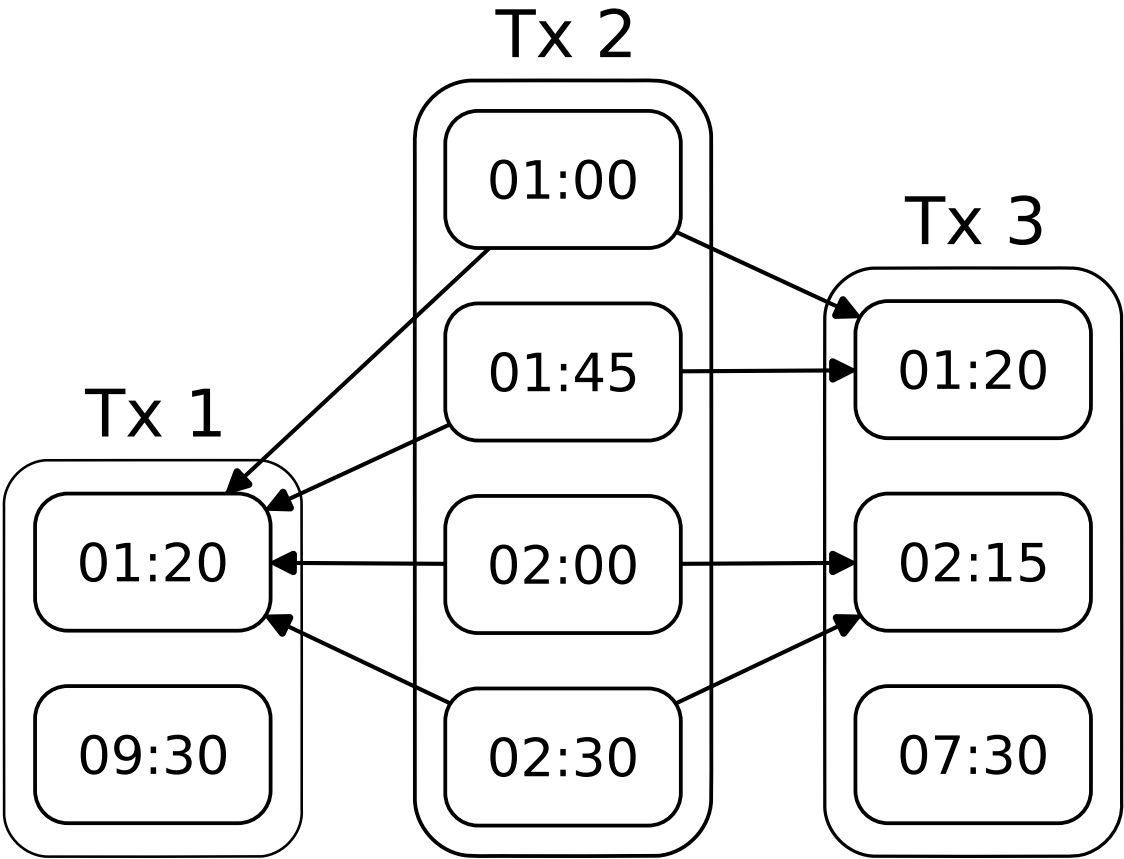
- Hausdorff distance [20]: This measure compares the dissimilarity of two point clouds by finding the maximum distance of any two points. This makes it sensitive to aberrations. It can handle sets of unequal length.
- Earth Mover distance [21]: This measure determines how similar two point clouds are by measuring how much one of the point clouds has to be changed to become equal to the other. Although it can handle sets of unequal length, it is not robust to aberrations.
- Chamfer distance [22]: The measure finds the mean minimum distance between all points in the compared point clouds in both directions. It is sensitive to aberrations and can handle sets of unequal length.
- One-sided Chamfer distance: The one-sided variant of the Chamfer distance only finds the mean minimum distance from one point cloud to the other. It is not sensitive to aberrations and can handle sets of unequal length.

Only the one-sided Chamfer distance fulfills all of our requirements. Therefore, we use the one-sided Chamfer distance with the 1-dimensional Euclidean distance measure between two timestamps, to measure the similarity between two sets of  $C$  (see Equation (1)). Here, the timestamps of other transactions are compared to the timestamps of the transaction  $t_a$ .

$$D(C_a, C_b) = \frac{1}{|C_a|} \sum_{c_a \in C_a} \min_{c_b \in C_b} \|c_a - c_b\| \quad (1)$$

Figure 6 shows how the proposed similarity matches the timestamps between the transactions. In the example, transaction 2 is the transaction of interest, with its CoinJoin input timestamps  $C_a$ . These are compared with the CoinJoin input timestamps of transaction 1  $C_b$  and transaction 3  $C_{b'}$ . The figure illustrates that only close timestamps are matched, while aberrations (from the perspective of transaction 2) are ignored.

<sup>2</sup> Although sampling could overcome this requirement, it would still distort the input of the measure and the result.



**Figure 6.** This transaction graph shows the one-sided Chamfer distance in action. With transaction 2 being the transaction of interest, it shows which input timestamps are used by the distance measure for transaction 1 and transaction 3. The two aberrations (09:30 in transaction 1 and 07:30 in transaction 3) are ignored.

4.2. Evaluation of the Proposed Similarity Measure

The similarity measure described in the previous section depends on the user behavior of creating multiple CSTs which spend outputs from shared CoinJoin processes. To determine if this behavior is prevalent and thus how useful the proposed similarity measure is, a dataset of CSTs which are created by the same entity is required. Subsequently, the linkability of CSTs belonging to the same entity, via the proposed similarity measure, can be evaluated. This type of data set can be created by finding address clusters that receive funds from multiple CSTs. This link may indicate that the senders of each CST are a single entity or that there exists another form of relationship among the senders. For the purpose of evaluating the usefulness of the proposed similarity measure, this correlation is sufficient.

We use Bitcoin’s<sup>3</sup> and Dash’s<sup>4</sup> core clients to collect all transactions of the respective blockchain until July 1, 2025. After classifying (see Appendices A–C) the CoinJoin transactions of each CoinJoin implementation (Whirlpool, Wasabi 2.0 and Dash) and their CSTs transactions, we apply the multi-input address clustering heuristic (see Section 2.2) to all non-CoinJoin transactions. Dash’s CoinJoin implementation changed over time, to avoid subtle differences of its early development, we exclude Dash CoinJoin transactions before the year 2018 from our analysis.

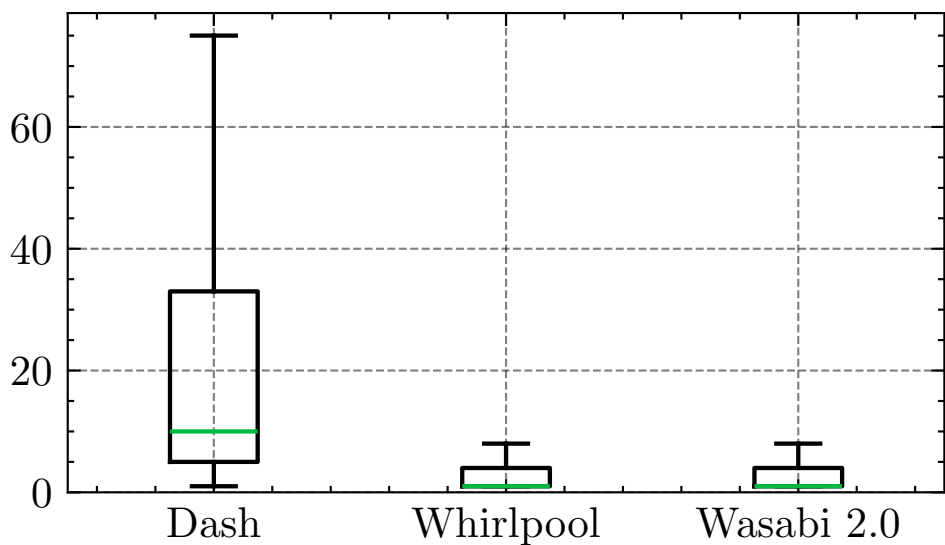
As described, we find all address clusters that receive funds from at least 2 CSTs. We want to exclude address clusters belonging to services, as their high transaction volume might falsify the evaluation results. Therefore, we exclude address clusters with more than 1000 addresses ("super-clusters" cf. [23]) and less than 2 CSTs. Note that CSTs can create multiple outputs and thus send funds to multiple address clusters. The resulting data set contains for each address cluster a set of CSTs. This data set is collected for the CoinJoin implementations of Dash, Whirlpool and Wasabi 2.0.

<sup>3</sup> <https://github.com/bitcoin/bitcoin> (Accessed 17. September 2025)

<sup>4</sup> <https://github.com/dashpay/dash> (Accessed 17. September 2025)

During the collection of the evaluation dataset, we also examined the input count of CSTs across Dash, Whirlpool and Wasabi 2.0. The results are presented in Figure 7. We note that for both Whirlpool and Wasabi 2.0 a large portion of CSTs only have a single input. This makes the transactions very generic in terms of time frames, and therefore the results of the similarity measure are not significant.

### Input Count of CoinJoin Spending Transactions



**Figure 7.** Box plots illustrate the number of inputs consumed by the CSTs originating from CoinJoin transactions. Outliers are not plotted. Both Whirlpool and Wasabi 2.0 CSTs utilize a very low number of inputs, 50% using only 1 input and 75% using up to 5 inputs. In contrast, Dash CSTs show that 50% use up to 9 inputs and 75% use up to 35 inputs.

We evaluate the similarity measure on the collected data set by calculating the  $K$  closest CSTs per assessed CST. If one of the address cluster’s other CSTs is in the top  $K$  similarity measure results, the address cluster is counted as linked. See Table 1 for the evaluation results. The last three lines of the table shows the ratio of linked address clusters for  $K \in \{10, 20, 30\}$ . With all three CoinJoin implementations showing that transactions can be linked via the proposed similarity measure, we conclude that the behavior of creating multiple CSTs which spend outputs from shared CoinJoin processes is prevalent, and thus our proposed similarity measure is useful.

**Table 1.** Shows for each CoinJoin implementation: the number of CSTs, the number of CSTs which are included in the evaluation, the number of unique address clusters the CSTs send funds to and the number of address clusters thereof which are included in the evaluation. Finally, the ratio of address clusters which contain at least one transaction pair which can be linked via the proposed similarity measure.

	Dash	Whirlpool	Wasabi 2.0
CSTs	284,069	198,984	342,228
Included CSTs	128,528	95,379	102,169
Address Clusters	146,066	247,112	432,919
Included Address Clusters	21,168	41,897	35,356
Linked Address Clusters (top 10)	14.22%	8.42%	10.40%
Linked Address Clusters (top 20)	27.33%	17.64%	20.97%
Linked Address Clusters (top 30)	63.19%	42.50%	40.29%

5. Conclusions

In this work, we present a novel similarity measure for transactions that use multiple outputs from CoinJoin transactions for payments. This technique allows for finding links between transactions and thus establishes a potential grouping of multiple transactions to a single controlling entity. This negatively impacts user privacy, especially because the aim of CoinJoin transactions is to weaken the link between fund ownership. Furthermore, we show that the behavior that enables this similarity measure is widespread and common in many CoinJoin implementations. This work is the first to propose such a measure, addressing a critical gap in the analysis of CoinJoin transaction linkage.

Furthermore, we proposed and evaluated a Wasabi 2.0 CoinJoin classification rule, which requires uncommon amount denominations to be present in each Wasabi 2.0 CoinJoin transaction.

The proposed similarity measure focuses only on the CoinJoin input timestamp of CoinJoin spending transactions. Expanding the analysis to other attributes (such as used amounts, input count, and connected CoinJoin graph) could improve the similarity measure.

**Author Contributions:** Conceptualization, M.Z., M.N. and B.K.; methodology, M.Z., M.N. and B.K.; validation, M.Z.; data curation, M.Z.; writing—original draft preparation, M.Z.; writing—review and editing, M.Z., M.N. and B.K.; visualization, M.Z.; supervision, M.N. and B.K.; project administration, M.N. and B.K.; All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The raw data supporting the conclusions of this article will be made available by the authors on request.

**Conflicts of Interest:** The authors declare no conflicts of interest.

Appendix A. Dash CoinJoin Transaction Classification Algorithm

Dash defines the following denominations: 10.0001, 1.00001, 0.100001, 0.0100001, 0.00100001 [18]. Classification criteria for Dash CoinJoin transactions are (cf. [8]):

- The number of inputs and outputs must be equal
- The transaction fee must be zero
- All input and output amounts must be part of the defined denominations
- All inputs and outputs must be of the same denomination

Appendix B. Whirlpool CoinJoin Transaction Classification

Whirlpool defines the following denominations: 0.001, 0.01 0.05, 0.5. Classification criteria for Whirlpool CoinJoin transactions are:

- Must have at least 5 outputs
- Must have at maximum 8 outputs
- Must have same amount of inputs as outputs
- All input and outputs must have the same denomination
- Must spend at least one input from a whirlpool Tx0 transaction

## Appendix C. Wasabi 2.0 CoinJoin Transaction Classification Algorithm

Wasabi 2.0 defines 79 denominations which are multiples of 5000, 6561, 8192 Satoshi. We define denominations which are not multiples of 5000 Satoshi as *uncommon* denominations [17].

Full list of Wasabi 2.0 denominations (in Satoshi):

5000, 6561, 8192, 10000, 13122, 16384, 19683, 20000, 32768, 39366, 50000, 59049, 65536, 100000, 118098, 131072, 177147, 200000, 262144, 354294, 500000, 524288, 531441, 1000000, 1048576, 1062882, 1594323, 2000000, 2097152, 3188646, 4194304, 4782969, 5000000, 8388608, 9565938, 10000000, 14348907, 16777216, 20000000, 28697814, 33554432, 43046721, 50000000, 67108864, 86093442, 100000000, 129140163, 134217728, 200000000, 258280326, 268435456, 387420489, 500000000, 536870912, 774840978, 1000000000, 1073741824, 1162261467, 2000000000, 2147483648, 2324522934, 3486784401, 4294967296, 5000000000, 6973568802, 8589934592, 10000000000, 10460353203, 17179869184, 20000000000, 20920706406, 31381059609, 34359738368, 50000000000, 62762119218, 68719476736, 94143178827, 100000000000, 137438953472.

Classification criteria for Wasabi 2.0 CoinJoin transactions are:

- All output scripts must be unique
- Input amounts must be at least 5000 Satoshi
- At least half of the output amounts must be part of the defined denominations
- The number of outputs must be at least the number of minimum participants
- Transactions must contain at least one uncommon denomination (new rule)

## References

1. Deuber, D.; Schröder, D., CoinJoin in the Wild: An Empirical Analysis in Dash. In *Computer Security – ESORICS 2021*; Springer International Publishing, 2021; pp. 461–480. [https://doi.org/10.1007/978-3-030-88428-4\\_23](https://doi.org/10.1007/978-3-030-88428-4_23).
2. Baldimtsi, F.; Brandao, J.; Chatzigiannis, P.; Karantaidou, I. Dash cryptocurrency deanonymization. <https://cina.gmu.edu/wp-content/uploads/2023/04/Dash-Cryptocurrency-Deanonymization.pdf>, 2023.
3. Biryukov, A.; Tikhomirov, S. Transaction Clustering Using Network Traffic Analysis for Bitcoin and Derived Blockchains. In *Proceedings of the IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2019, pp. 204–209. <https://doi.org/10.1109/infcomw.2019.8845213>.
4. Awan, M.K.; Cortesi, A., Blockchain Transaction Analysis Using Dominant Sets. In *Computer Information Systems and Industrial Management*; Springer International Publishing, 2017; pp. 229–239. [https://doi.org/10.1007/978-3-319-59105-6\\_20](https://doi.org/10.1007/978-3-319-59105-6_20).
5. Lu, Y.; Wang, H. Similarity Matching, Classification, and Recognition Mechanism for Transaction Analysis in Blockchain Environment. *IEEE Transactions on Consumer Electronics* **2024**, 70, 7018–7027. <https://doi.org/10.1109/tce.2024.3473691>.
6. Tovanich, N.; Cazabet, R. Fingerprinting Bitcoin entities using money flow representation learning. *Applied Network Science* **2023**, 8. <https://doi.org/10.1007/s41109-023-00591-2>.
7. Zavřel, J.; Koutenský, M.; Dolejška, D.; Veselý, V. Tumbling down the stairs: Exploiting a tumbler's attempt to hide with ordinary-looking transactions using wallet fingerprinting. *Forensic Science International: Digital Investigation* **2025**, 52, 301869. <https://doi.org/10.1016/j.fsidi.2025.301869>.
8. Ziegler, M.H.; Nowostawski, M.; Katt, B., The Privacy Impact of Dash Mixing Fee Payments. In *Data and Applications Security and Privacy XXXIX*; Springer Nature Switzerland, 2025; pp. 427–438. [https://doi.org/10.1007/978-3-031-96590-6\\_23](https://doi.org/10.1007/978-3-031-96590-6_23).
9. Goldfeder, S.; Kalodner, H.; Reisman, D.; Narayanan, A. When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies. *Proceedings on Privacy Enhancing Technologies* **2018**, 2018, 179–199. <https://doi.org/10.1515/popets-2018-0038>.



10. Stütz, R.; Stockinger, J.; Haslhofer, B.; Moreno-Sanchez, P.; Maffei, M. Adoption and Actual Privacy of Decentralized CoinJoin Implementations in Bitcoin, 2021. <https://doi.org/10.48550/ARXIV.2109.10229>.
11. Schnoering, H.; Vazirgiannis, M. Heuristics for Detecting CoinJoin Transactions on the Bitcoin Blockchain, 2023. <https://doi.org/10.48550/ARXIV.2311.12491>.
12. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. <https://bitcoin.org/bitcoin.pdf>, 2008.
13. Meiklejohn, S.; Pomarole, M.; Jordan, G.; Levchenko, K.; McCoy, D.; Voelker, G.M.; Savage, S. A fistful of bitcoins: characterizing payments among men with no names. In Proceedings of the Proceedings of the 2013 conference on Internet measurement conference. ACM, 2013, IMC'13. <https://doi.org/10.1145/2504730.2504747>.
14. Maxwell, G. CoinJoin: Bitcoin privacy for the real world. <https://bitcointalk.org/index.php?topic=279249>, 2013.
15. Ziegler, M.H.; Nowostawski, M.; Katt, B. A Systematic Literature Review of Information Privacy in Blockchain Systems. *Journal of Cybersecurity and Privacy* **2025**, *5*, 65. <https://doi.org/10.3390/jcp5030065>.
16. Samurai Wallet Authors. Samurai Wallet - Whirlpool Repository. <https://github.com/Samurai-Wallet/Whirlpool/commits/whirlpool/>, 2019.
17. Wasabi Wallet Authors. Wasabi Wallet Documentation. <https://docs.wasabiwallet.io/FAQ/FAQ-UseWasabi.html>, 2025.
18. Dash Core Group, I. Dash Core CoinJoin Documentation. <https://docs.dash.org/projects/core/en/stable/docs/guide/dash-features-coinjoin.html>, 2025.
19. Dickey, D.A.; Fuller, W.A. Distribution of the Estimators for Autoregressive Time Series With a Unit Root. *Journal of the American Statistical Association* **1979**, *74*, 427. <https://doi.org/10.2307/2286348>.
20. Hausdorff, F. *Grundzüge der Mengenlehre*; Chelsea: New York, 1978.
21. Rubner, Y.; Tomasi, C.; Guibas, L. A metric for distributions with applications to image databases. In Proceedings of the Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271). Narosa Publishing House, 1998, ICCV-98, pp. 59–66. <https://doi.org/10.1109/iccv.1998.710701>.
22. Barrow, H.G.; Tenenbaum, J.M.; Bolles, R.C.; Wolf, H.C. Parametric correspondence and chamfer matching: Two new techniques for image matching. In Proceedings of the Proceedings: Image Understanding Workshop, 1977, pp. 21–27.
23. Harrigan, M.; Fretter, C. The Unreasonable Effectiveness of Address Clustering. In Proceedings of the 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld). IEEE, 2016, pp. 368–373. <https://doi.org/10.1109/uic-atc-scalcom-cbdcom-iop-smartworld.2016.0071>.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.