Towards Measuring the Traceability of Cryptocurrencies

Domokos M. Kelen
ELKH Institute for Computer Science and Control
(SZTAKI)
Budapest, Hungary

István András Seres
Eötvös Loránd University
Budapest, Hungary

ABSTRACT

Cryptocurrencies aim to replicate physical cash in the digital realm while removing centralized and trusted intermediaries. Decentralization is achieved by the blockchain, a permanent public ledger that contains a record of every transaction. The public ledger ensures transparency, which enables public verifiability but harms untraceability, fungibility, and anonymity. In the last decade, cryptocurrencies attracted millions of users, with their total market cap reaching approximately three trillion USD at its peak. However, their anonymity guarantees are poorly understood and plagued by widespread misbeliefs. Indeed, previous notions of privacy, anonymity, and traceability for cryptocurrencies are either non-quantitative or inapplicable, e.g., computationally hard to measure.

In this work, we put forward a formal framework to measure the (un)traceability and anonymity of cryptocurrencies, allowing us to quantitatively reason about the mixing characteristics of cryptocurrencies and the privacy-enhancing technologies built on top of them. Our methods apply absorbing Markov chains combined with Shannon entropy. To the best of our knowledge, our work provides the first practical, efficient, and probabilistic measure to assess the traceability of cryptocurrencies quantitatively, which also generalizes to entire cryptocurrency transaction graphs. We implement and extensively evaluate our proposed traceability measure on several cryptocurrency transaction graphs. Among other quantitative results, we find that in the studied one-week interval, the Bitcoin blockchain, on average, provided comparable but quantifiably more natural mixing than the Ethereum blockchain.

KEYWORDS

Blockchain privacy, Untraceability, Fungibility, Anonymity.

1 INTRODUCTION

Traditional monetary systems and cryptocurrencies have different centralization, transparency, and openness properties. The former operates in a centralized setting, where the transaction history is not publicly available, hence, the ledger of transactions is not auditable. In stark contrast, cryptocurrencies, most notably Bitcoin [38] and Ethereum [59], operate in the permissionless and decentralized setting, where all transactions are readily available for inspection in a public ledger, typically in a blockchain. The blockchain stores all valid transactions in a timestamped manner, which makes it possible to publicly verify the monetary system's state. The public nature of the ledger comes at the cost of rendering

each coin in the system traceable and non-fungible, as its entire history is transparent. The privacy implications of traceability and its effects on anonymity are not hypothetical: both academic and industry efforts have been made to extract insights and intelligence from these public ledgers [10, 14, 33]. Several cryptocurrency addresses are black-listed at major cryptocurrency exchanges due to their alleged illicit activities [35].

The traceable nature of cryptocurrencies creates friction, inefficiencies, and ample opportunities for censorship [54, 56, 58]. These major deficiencies render cryptocurrencies a subpar form of money compared to cash from a privacy point of view. The traceability of cryptocurrencies, specifically Bitcoin, was first studied by Meiklejohn et al. in 2013 [33]. Since then, significant efforts – both academic [2, 7, 28, 37, 61] and industry [5, 11] – have been committed to studying, understanding, and exploiting the traceability of these novel digital currencies.

There are two sides to cryptocurrency traceability. The first line of work aims at extracting information and intelligence thanks to the traceable nature of cryptocurrencies. Typically, these works try to trace illicit funds with heuristics [36, 53] and ad-hoc methods, sometimes even across multiple blockchains [26]. In contrast, the second line of work aims to enhance anonymity and untraceability by employing cryptographic tools, e.g., zero-knowledge proofs [9, 40, 45] or secure multi-party computation [31]. To tackle these issues, besides standalone privacy-focused cryptocurrencies, such as Monero [52] or Zcash [45], several privacy-enhancing technologies (PETs) have been proposed for cryptocurrencies [8, 25, 31, 39, 44, 51]. Nevertheless, their quantitative benefits in terms of anonymity, (un)traceability or fungibility remain unexplained from a quantitative point of view.

In this work, we are primarily interested in what we call *objective* traceability, where we measure the uncertainty that *any* observer of the public ledger has about the origins of coins residing at a specific address. On the other hand, *subjective* traceability measures the uncertainty of a more potent adversary about the source of some coins. In particular, when measuring subjective traceability, we allow the adversary to have external knowledge or use additional heuristics about the flow of money. This external knowledge may come from the peer-to-peer (P2P) layer [19] or any source other than the public ledger, e.g., application layer [23]. In essence, objective traceability measures the maximum uncertainty given only the public ledger as an information source, while subjective traceability depends on the observer.

We believe that currently, the quantification of cryptocurrency traceability is lacking. To the best of our knowledge, there is no prior satisfactory formal quantitative framework to argue about the traceability of cryptocurrencies, e.g., Bitcoin [38], Ethereum [59], or Zcash [45]. Our main goal is to define a metric which is general

enough to encompass most transaction graphs arising from real-world use cases, see Section 4. Moreover, it should be efficiently computable on real-world blockchain data. Therefore, this work provides a method to quantify cryptocurrencies' objective traceability and evaluate it on several cryptocurrency transaction graphs. Our quantitative framework for traceability allows a clear comparison of cryptocurrencies and privacy-enhancing technologies concerning their achieved untraceability benefits. We also show how one could extend our traceability framework to quantify the anonymity guarantees of cryptocurrency transaction graphs to calculate subjective measures, i.e., contingent on "adversarial" knowledge of traceability.

In many applications, we would like an anonymity score or degree of anonymity [15, 47]. But this is impossible on a pseudonymous blockchain, e.g., Bitcoin or Ethereum, since these blockchains do not provide strong identities. Hence, defining a useful anonymity score is futile in our context. On the other hand, anonymity and traceability are closely related. Traceability is a good proxy for assessing the anonymity of *coins* or *outputs*. Further evidence for this lies in the fact that most privacy-enhancing technologies explicitly focus on making it harder to trace the source of funds on transaction graphs.

Despite this relation, we want to emphasize that traceability is not a privacy or anonymity metric. In a typical (information-theoretic) anonymity metric [15, 47], the adversary observes a communication system and outputs a probability distribution of the possible targets in an anonymity set. The entropy of this distribution captures the adversary's uncertainty about the target. Therefore, the anonymity of a communication system depends on adversarial capabilities and background knowledge. In contrast, we wish to provide an absolute traceability metric that does not depend on external components but solely on the public ledger containing all transactions. Informally, an untraceability metric should measure how "unique" unspent coins are in a cryptocurrency.

The uniqueness of a coin is determined by its transaction history, which we model in Section 3. The two main applications that motivate the introduction of an untraceability metric are as follows.

- **Privacy-focused cryptocurrency wallet.** A user typically owns several coins (or addresses) in a cryptocurrency wallet. However, the coins owned by the user might have differing levels of traceability. A traceability measure could help users select the coin with sufficient traceability/anonymity required to perform a privacy-critical transaction as described in Section 4.
- **Benchmarking cryptocurrency and PET designs.** A traceability and anonymity metric allows us to compare the efficacy of various privacy-preserving cryptocurrency designs quantitatively, e.g., Zcash [45], Monero [52] or Dash [16]. Similarly, one could quantitatively study the traceability and anonymity guarantees of privacy-enhancing overlays, e.g., mixers [8, 50], stealth addresses [13], tumblers [24, 32, 40, 44, 46], etc., built on top of cryptocurrencies.

In this work, we provide the following contributions.

- **Quantitative theoretical framework for cryptocurrency (un)traceability.** We propose a model to quantify the traceability of cryptocurrencies. Our traceability metric can be

considered a generalization of the information-theoretic anonymity metric by Diaz et al. [15] for cryptocurrencies and transaction graphs.
- **Empirical evaluation on major blockchains.** We extensively evaluate our proposed traceability metric on major cryptocurrencies, such as Bitcoin, Ethereum, Zcash, and some ERC-20 tokens. For the first time, we are able to assess the traceability of crypto assets quantitatively.
- **Open-source implementation.** The implementation of our traceability metric is open-sourced and available for reproduction upon publication.

The rest of this paper is organized as follows. In Section 2, we provide our system and transaction graph models and define our proposed untreaceability score. In Section 3, we apply our model to cryptocurrencies. We detail a few applications of our metric in Section 4. In Section 5, we evaluate our proposed traceability measure on the public ledgers of some cryptocurrencies. We discuss limitations and possible extensions in Section 6. In Section 7, we review previous attempts to quantify the privacy provisions of cryptocurrencies. Finally, we conclude our work in Section 8 with open research directions.

2   UNTRACEABILITY

First, we describe our system model. Afterwards, we introduce our transaction graph model, which we subsequently convert into an absorbing Markov chain to define a metric of untraceability.

2.1   System Model

In our model, we distinguish between three system components: *users*, the public *ledger*, and *adversaries*.

- **Users** issue transactions. We assume users can issue transactions in a privacy-preserving manner. In other words, we do not model privacy leakages on the network and application layers and solely focus on the untraceability achieved by the issued and recorded transactions.
- **Ledger** is an immutable, public database recording faithfully every transaction users perform. The ledger is known to every participant of the system. We model the ledger as a weighted, directed graph, see Section 2.2.
- **Adversary** aims to establish the sources of each coin with respect to a certain "source" set when it has a particular "knowledge" set (both of these terms are defined precisely later). We provide an untraceability metric with respect to a given "source" set from the adversary's perspective, i.e., when the adversary relies on a specific knowledge set. The adversary might possess external information sources, e.g., social media, application layer data, etc., and tools, e.g., logging broadcast transaction IP addresses, to increase its tracing capabilities. While the adversary model in this paper does not utilize such information sources, our model can naturally incorporate them (in the applied transition matrix) whenever they are available, see Section 6.

2.2   Transaction Graph Model

We define a general transaction graph model that can be used to calculate untraceability, independent of the exact implementation

details of cryptocurrency transaction networks. We discuss the transformation of real-world cryptocurrency networks into transaction graphs in Section 3.

A transaction graph G is a weighted, directed graph $\overrightarrow{G} = (V, E, a)$, where V consists of addresses, $E \subseteq V \times V$ is a set of ordered pairs that denotes transactions, while $a : V \times V \to \mathbb{N}$ assigns the currency amounts $a((v_1, v_2))$ transferred from v_1 to v_2. Note that $a((v_1, v_2)) = 0$ if $(v_1, v_2) \notin E$. The definition $Im(a) = \mathbb{N}$ is not a restriction since we can consider the smallest denomination of the currency, for example, satoshi for Bitcoin or wei for Ether. We set no restrictions about a node's incoming and outgoing amounts: nodes with incoming surplus become the sinks, while nodes with outgoing surplus become the sources of the transaction graph.

Looking ahead, we think of V as the set of all entities in the ledger: addresses of users or smart contracts, as well as time-snapshots of addresses in some cases, see Section 3.4. Multiple details about converting ledgers into transaction graphs require special attention, e.g., multiple transactions between two addresses or the local graph structures resulting from the use of UTXO-based currencies. We detail these transformations in Section 3.

2.3 Desiderata

We want to provide an untraceability score $U_{S,G,\mathcal{A}}(\cdot)$ defined on the nodes V of the transaction graph G using adversarial knowledge $\mathcal{A}$, i.e., $U_{S,G,\mathcal{A}} : V \to \mathbb{R}_0^+$ with respect to a source set $\emptyset \neq S \subset V$. Note that in the following, we will omit from the subscript the set of source nodes S to which we trace the money back, the transaction graph G, or the adversary $\mathcal{A}$ whenever they are irrelevant or obvious from the context to avoid notational clutter.

Next, we outline the requirements we believe a traceability quantification should possess.

- **Range**. We want to have a positively or negatively oriented range, where one extreme of the spectrum signifies total untraceability (i.e., k-anonymity), while the other extreme is total traceability (i.e., exact identification of source).
- **Fine-grained**. Instead of a binary metric, we want the metric to reflect the *degree* of traceability, also capable of capturing constructs between total traceability and total untraceability.
- **Decay**. We want the traceability value to naturally decrease with time: as a source gets mixed with other sources, its traceability should decrease. Formally, we want the maximum traceability to decrease (or remain constant) whenever edges are added to a subgraph.
- **Tamper-resistant**. We want the measure to be such that without the use of third-party funds (i.e., in isolation), the traceability cannot be decreased. For an illustrative example, see Figure 1.
- **Subjective**. The metric should be able to incorporate background knowledge (i.e., private information) to yield a subjective measure of traceability.
- **Efficient**. It should be computationally feasible to calculate in practice, i.e., for graphs with millions of nodes and edges.

We review and evaluate previous approaches in Section 7 and Table 3, whether they satisfy our desiderata. For instance, the previously proposed Boltzmann score does not satisfy our efficiency
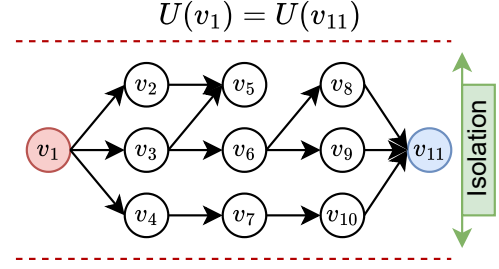

$$U(v_1) = U(v_{11})$$

Figure 1: Illustrating the requirement of "tamper resistance" for our proposed metric, see Section 2.3. Intuitively, the traceability metric for v_1 and v_{11} should be the same since no third-party coins are mixed into v_1.

requirement as it reduces to the NP-hard subset sum problem. Similarly, the system's anonymity metric [18, 22] reduces to computing the permanent of the transaction graph's adjacency matrix, again an NP-hard problem. In Section 2.5, we define our traceability metric and argue why it satisfies our desiderata above.

2.4 Tracing Money Backwards

A transaction graph G has many source nodes. The source nodes correspond to transactions that spend money they did not receive. These can either be coinbase transactions that mint money into existence or nodes with an initial non-zero balance in the case of a time snapshot of a transaction graph. Let us denote the set of source nodes in the transaction graph as S, i.e., S $\subseteq V$. Similarly, many nodes act as sinks in the transaction graph, i.e., they do not spend the money they receive. To trace the origin of each output residing in sink nodes to source nodes, we reverse the direction of the edges and start random walks on the reversed edges.
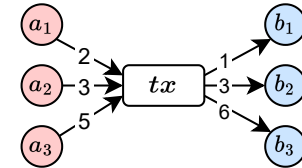
(a) A simple transaction in a UTXO-based cryptocurrency. Each input $(a_i)_{i=1}^3$ and output $(b_i)_{i=1}^3$ are connected through a tx node. The weights on the edges represent the value of the inputs and outputs used in the transaction.
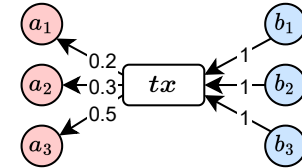
(b) The transition probability graph of the example transaction from Figure 2a. Note that edges are reversed, i.e., the random walk proceeds from outputs towards the transaction's inputs. Numbers on the edges represent transition probabilities according to the transition matrix, cf. Equation (1).

Figure 2: Transition probabilities in the Markov chain.

While tracing can naturally be performed along a backwards topological order if the transaction graph has no loops, for the general case, we will have to define random walks and Markov chains as follows. First, we extend the set of nodes of the graph G by creating an auxiliary node $v' \in V'$ for each source node v in G with a corresponding edge $(v', v) \in E$ and weight $a((v', v))$. This edge represents an imaginary transaction that accounts for the outgoing surplus of v. This way, only the newly created nodes V' act as sources, and all of the original nodes V either act as transient nodes or sinks. More importantly, the source nodes of the modified graph are guaranteed to have no incoming edges. The newly created nodes V' together with the original nodes V of G form the states $U = V \cup V'$ of the Markov chain G^* with edge directions reversed. When reversing the edges of G, the nodes in V' become absorbers of the Markov chain. Intuitively, this is done by adding self-loops to these nodes so that a random walk cannot escape once it reaches an absorber node. Ultimately, we are interested in computing the distribution of absorbing probabilities over the absorbers when starting random walks from sinks of the original graph G. We denote the transition probability matrix as $\mathbf{P}$, where $\mathbf{P}_{i,j}$ denotes the transition probability from state u_i to u_j in the Markov chain. We define

$$\mathbf{P}_{i,j} \overset{def}{=} \begin{cases} 1, & \text{for absorber self-loops, i.e., } u_i = u_j \in V' \\ \frac{a((u_j, u_i))}{\sigma(u_i)}, & \text{for reversed edges, i.e., } u_i \in V, a((u_j, u_i)) > 0 \\ 0 & \text{otherwise,} \end{cases}$$
$$(1)$$

where $\sigma(u_i) = \sum_{u_k \in U} a((u_k, u_i))$, i.e., the sum of the incoming transactions to u_i. Intuitively, this means that each output of a node traces back to all inputs of the node, proportionally to the transferred amounts. For an example of this calculation, see Figure 2. The Markov chain defined this way is an absorbing Markov chain, meaning that each walk from a sink of G must eventually reach an absorbing state with probability 1. For formal proof of this claim, see Section 2.7.

2.4.1 Other policies. While we consider the transaction graph model to be given, tracking money backwards is highly subjective and depends on the adversary's goals and background knowledge. The random walk's transition matrix $\mathbf{P}$ can capture these different, subjective background knowledge. Various adversaries with varying background knowledge can define their own subjective untraceability score using their "custom" transition matrices.

To illustrate the flexibility of our proposed formal model, we mention an alternative transition matrix inspired by the risk-scoring models of Möser, Böhme, and Breuker [34]. In particular, we define the transition matrices , $\mathbf{P}^{\text{pois}}$ corresponding to the *poison risk scoring policy*. The poison policy defines a set of tainted addresses $T \subseteq V$. If a transaction has even a single tainted input address, then all output addresses become tainted as well. The poison policy can be formulated as a transition matrix as follows.

$$\mathbf{P}_{i,j}^{\text{pois}} \overset{def}{=} \begin{cases} 1, & \text{for absorber self-loops, i.e., } u_i = u_j \in V' \\ \frac{1}{d(u_j)}, & \text{for reversed edges, } u_j \in V, \text{ for } d(u_j)) > 0 \\ 0 & \text{otherwise,} \end{cases}$$
$$(2)$$

where $d(u_j)$ is the number of tainted neighbor nodes of u_j in V. Similarly, other risk scoring or black listing policies can be easily incorporated into our formal model by adjusting the Markovian random walk's transition matrix $\mathbf{P}$ accordingly.

In the rest of this work, we will apply and evaluate on several cryptocurrency transaction graphs the transition matrix defined in Equation (1), cf. Section 5.

To calculate the absorbing probabilities, we follow the outline of Kemény and Snell [27]. Let G^* have $t = |V|$ transient states and $r = |V'|$ absorbing states. First, we rearrange the entries of $\mathbf{P}$ as follows. Let

$$\mathbf{P} = \begin{bmatrix} \mathbf{Q} & \mathbf{R} \\ \mathbf{0} & \mathbf{I}_r \end{bmatrix}, \qquad (3)$$

where $\mathbf{Q}$ is a t-by-t matrix representing transition probabilities from V to V, $\mathbf{R}$ is a t-by-r matrix representing transition probabilities from V to V' and $\mathbf{I}_r$ is the r-by-r identity matrix. The probability of transitioning from v_i to v_j in exactly k steps is the (i, j) entry of $\mathbf{Q}^k$. The probabilities of being absorbed in state $v_j \in V'$ starting from transient state $v_i \in V$ can be obtained as the (i, j) element of the $t \times r$ absorption probability matrix $\mathbf{B}$, which is computed as

$$\mathbf{B} = \lim_{n \to \infty} \mathbf{P}^n. \qquad (4)$$

Each row $\mathbf{B}_i$ of the resulting matrix $\mathbf{B}$ represents a probability distribution over the absorbing nodes V' when starting from node $v_i \in V$, i.e., $\forall i, j : 0 \le \mathbf{B}_{i,j} \le 1$ and $\sum_j \mathbf{B}_{i,j} = 1$.

2.5 Defining untraceability

Having the necessary mathematical framework in place, we can now continue by defining a quantitative measure of traceability. However, as it turns out, an *un*traceability score is more natural, as it is analogous to the degree of anonymity [15].

DEFINITION 1 (**UNTRACEABILITY SCORE**). *The untraceability score is defined as the amount of information the adversary lacks (in bits) for achieving total certainty in tracing the money. More formally, the untraceability $U_{S,G,\mathcal{A}}(v_i)$ of a sink v_i in the transaction graph G with respect to a source set S and adversary $\mathcal{A}$ is defined as*

$$U_{S,G,\mathcal{A}}(v_i) := H(\mathbf{B}_i), \qquad (5)$$

where $H(\mathbf{B}_i)$ is the Shannon entropy (in bits) of the probability distribution $\mathbf{B}_i$. Often, we omit $S, G, \mathcal{A}$ from the subscript of $U_{S,G,\mathcal{A}}(v_i)$ if they are clear from the context or irrelevant.

Intuitively, $U_{S,G,\mathcal{A}}(v_i)$ measures the uncertainty about the origins of money residing at a sink node v_i of the transaction graph G with respect to the sources S and knowledge $\mathcal{A}$. One might define untraceability with other entropy functions $H(\cdot)$ such as min-entropy or the many variants of Rényi-entropy [43]. However, we stick to the Shannon entropy to remain consistent with prior work that uses Shannon entropy to measure the degree of anonymity in various
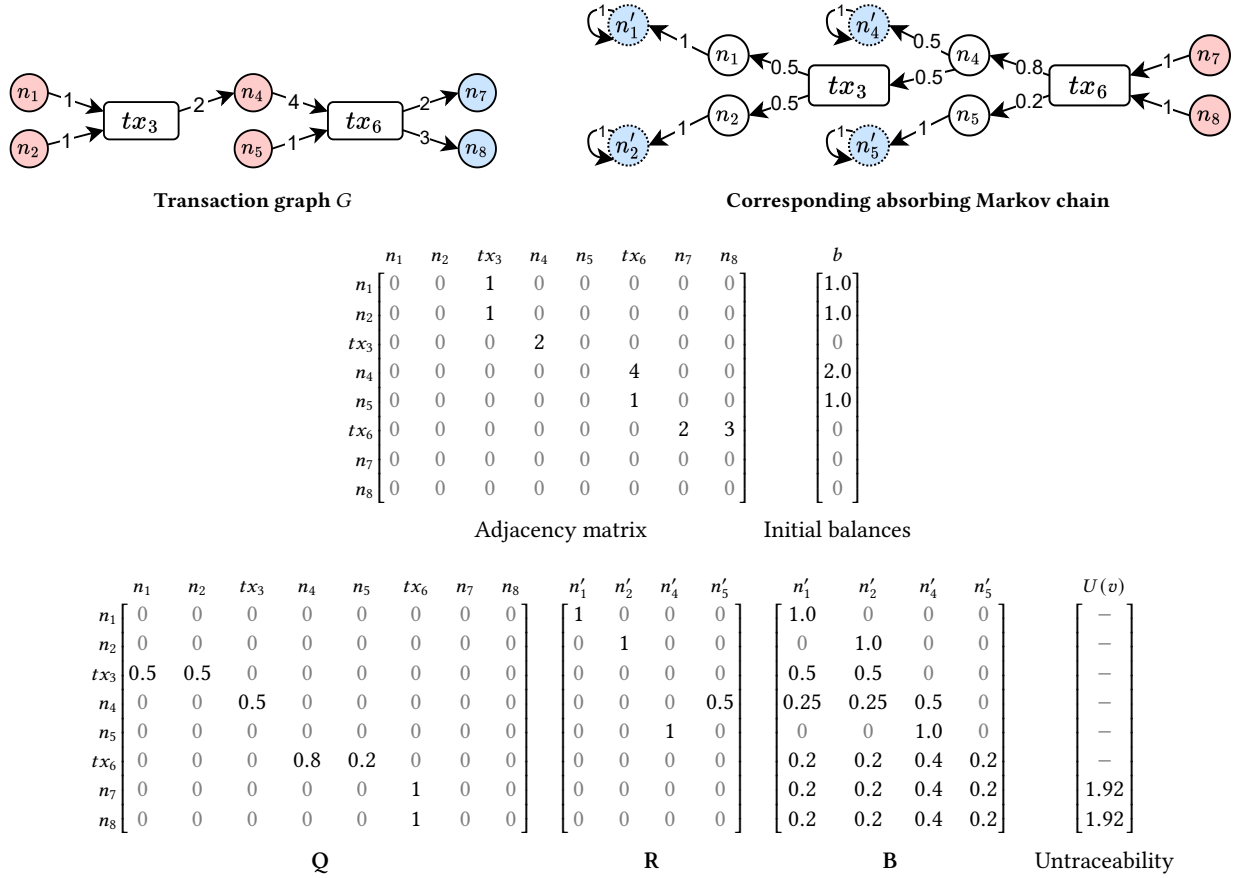
Figure 3: A simple example summarizing our method to obtain an untraceability score on a transaction graph G. Red nodes are sources, blue nodes are sinks. While outputs of transactions cannot be sources in a regular UTXO-based graph, the initial balance of node n_4 is possible when merging different UTXOs belonging to the same public key. We add an auxiliary absorber node in the corresponding Markov chain for each source in G, i.e., each node with an initial balance. The transition matrix of the Markov chain is split into Q and R as described in Equation (3) to calculate the absorbing probabilities $B = (I - Q)^{-1}R$. The untraceability score is defined as the Shannon entropy of the distribution of the absorbing probabilities for each sink in the original transaction graph G.

mix networks [15, 47]. For a full example of calculating untraceability on a transaction graph, see Figure 3. For an illustrative example calculation where G includes a directed cycle, see Figure 4.

Next, we argue informally why our introduced untraceability measure satisfies our previously introduced desiderata. The properties **range** and **fine-grained** are obviously satisfied. For **subjective**, see Sections 4 and 6: the definition of the transaction graph that one uses can easily incorporate epistemic knowledge while only using public information assumes the public as the subject. **Efficiency** and computation are discussed in Section 2.6.

Tamper-resistance is true due to the properties of the Markov chain, as any isolated subgraph with given inputs and outputs can be contracted into a single point with the same inputs and outputs, without changing the probability values calculated for any output of the graph. Finally, **decay** holds, as mixing two sources always results in the traceability decreasing for the output with respect to the maximum traceability of the outputs, while splitting a source

does not affect traceability. The same, applied to *un*traceability, can be expressed formally as $\forall v_1, v_2 \in V$ if $\exists (v_1, v_3), (v_2, v_3) \in E$: $U(v_3) \geq \min(U(v_1), U(v_2))$.

2.6 Computational considerations

Another characterization of **B** yields a more efficient way of computing our proposed score:

$$B = \lim_{n \to \infty} P^n = NR, \qquad (6)$$

where **N** is called the fundamental matrix. It can be shown that

$$N = \sum_{k=0}^{\infty} Q^k = (I_t - Q)^{-1}. \qquad (7)$$

We are also interested in computing *the length of the random walks* before they are absorbed in an absorber node. The expected number of steps before being absorbed when starting from transient state v_i is the ith entry of the vector $t = N1$, where **1** is a length-t column

Adjacency matrix

	n_1	n_2	n_3	n_4	n_5	n_6	b
n_1	0	8	0	0	0	0	8.0
n_2	0	0	14	0	0	0	0
n_3	0	0	0	16	0	0	2.0
n_4	0	0	0	0	10	6	0
n_5	0	0	0	0	0	0	0
n_6	0	6	0	0	0	0	0

Transaction graph G — **Corresponding absorbing Markov chain** — **Adjacency matrix** — **Initial balances**

	n_1	n_2	n_3	n_4	n_5	n_6		n_1'	n_3'		n_1'	n_3'		$U(v)$
n_1	0	0	0	0	0	0		1	0		1.0	0		–
n_2	0.43	0	0	0	0	0.57		0	0		0.857	0.143		–
n_3	0	0.875	0	0	0	0		0	0.125		0.75	0.25		–
n_4	0	0	1	0	0	0		0	0		0.75	0.25		–
n_5	0	0	0	1	0	0		0	0		0.75	0.25		0.81
n_6	0	0	0	1	0	0		0	0		0.75	0.25		–

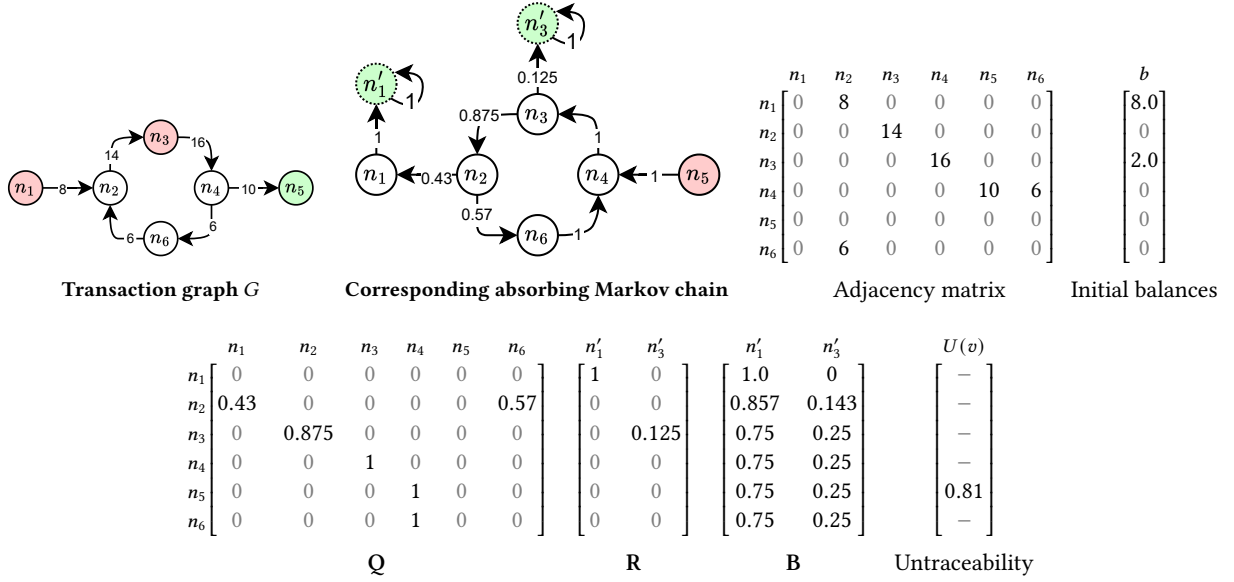$\mathbf{Q}$ — $\mathbf{R}$ — $\mathbf{B}$ — Untraceability

Figure 4: An illustrative example calculation of untraceability where the transaction graph G includes a directed cycle. Note that directed cycles in transaction graphs are only possible in the account-based model or, alternatively, in a UTXO-based model if one is able to merge multiple UTXOs corresponding to the same public key.

vector whose entries are all 1. Inverting the matrix $(\mathbf{I}_t - \mathbf{Q})$ is computationally infeasible in practice if the transition graph grows too large. In our large-scale evaluations, we approximate $\mathbf{B}$ instead as

$$\mathbf{B} = \left(\sum_{k=0}^{\infty} \mathbf{Q}^k\right)\mathbf{R} \approx \sum_{k=0}^{k_{\max}} (\mathbf{Q}^k \mathbf{R}) = \mathbf{R} + \mathbf{QR} + \mathbf{Q}^2\mathbf{R} + \ldots + \mathbf{Q}^{k_{\max}}\mathbf{R}. \quad (8)$$

The formula at the right-hand side of Equation (8) can be calculated by repeatedly multiplying a variable by $\mathbf{Q}$ and adding the value to an accumulator variable at each step. Instead of using a fixed constant $k_{\max}$, we stop the calculation when

$$\|\mathbf{Q}^k\mathbf{R}\|_{\max} \le \delta_{\text{threshold}} \quad (9)$$

is reached, which is guaranteed to happen as $\lim_{k\to\infty} \|\mathbf{Q}^k\mathbf{R}\|_{\max} = 0$, for a formal proof see Section 2.7. An advantage of approximating $\mathbf{B}$ this way is that the right-hand side of Equation (8) can be computed in parallel by dividing $\mathbf{R}$ column-wise and processing each piece separately. We discuss the choice of $\delta_{\text{threshold}}$ in our large-scale evaluation in Section 5.

2.7 Markov Chain Convergence

We prove that the random walks starting from sinks of the transaction graph indeed get absorbed with probability 1.

THEOREM 2.1. *In a Markov chain defined as in Section 2, a random walk starting from a sink of the transaction graph G and progressing with transition probabilities defined by the transition matrix P of Equation (1) ends up being absorbed in an absorber node $v' \in V'$ with probability 1.*

PROOF. We define the *excess* function $b : V \to \mathbb{Z}$ as

$$b(u) = \sum_v a(v, u) - \sum_v a(u, v). \quad (10)$$

This means that $b(u) > 0$ for sinks and $b(u) < 0$ for sources of G.

Let us consider the transaction graph extended with auxiliary nodes V', using each edge backward. Let V^r denote the set of nodes v reachable from a sink s on a directed path in this graph, including the sinks as well.

We prove that each $w \in V^r$ is connected to at least one absorber v' with a finite state sequence of non-zero probability. To see this, start a walk from a sink s to w and continue as follows. For each edge (u, v) of the walk, set $a(u, v) := a(u, v) - 1$. If this results in $a(u, v) = 0$, delete the edge from the graph. Continue this until either reaching an absorber of the Markov chain (i.e. u with $b(u) < 0$), or a node where there are no more incoming edges.

The walk from the sink s to a absorber cannot progress infinitely since the maximum number of steps taken cannot exceed $\sum_{u,v} a(u, v)$. Further, apart from the starting and ending nodes, $b(u)$ remains identical for all intermediate nodes u. And finally, for a node u with $b(u) \ge 0$, there must be at least one incoming edge where we can continue the walk. Since the only nodes with $b(u) < 0$ are $u \in V'$, the walk must terminate in an absorber.

Thus, from any node $v \in V^r$, there is a nonzero chance of at least ϵ_v of reaching an absorber in k steps. Let $0 < \epsilon = \min_v \epsilon_v$. Hence, the probability that a random walk from a sink does not reach an absorber in $k \cdot n$ steps is bounded by $(1 - \epsilon)^n \to 0$. □

Note that the proof still allows separate components without sinks or sources to diverge, i.e., when $V^r \ne V \cup V'$. Such a component could be composed of, for example, a directed cycle with equal weights on all of the edges. These are not interesting to us since

we are only interested in the absorbing entropies for sinks of the transaction graph G, so we can simply delete the relevant nodes.

3 GRAPH REPRESENTATION OF CRYPTOCURRENCY TRANSACTIONS

In this section, we show how one can arrive at a transaction graph from real-world cryptocurrency ledgers. We handle UTXO- and account-based cryptocurrencies separately and describe two ways of handling the temporality of transactions.

3.1 Cryptocurrencies with the UTXO-model

In cryptocurrencies that use the unspent transaction output (UTXO) model, such as Bitcoin [38] or Zcash [45], money is represented as coins, also known as UTXOs. In a simplified form, each UTXO has a nominal value and defines a public key that can spend that output. Transactions consist of input and output coins. Each input references outputs of previous transactions that are not yet spent, and in return, transactions create new coins, referred to as output UTXOs. Transactions are identified by their cryptographic hash. A critical invariant enforced by cryptocurrency network participants is that the sum of all input values must be greater or equal to the sum of all output values in a transaction, i.e., it is impossible to print money out of thin air.

The UTXO model lends itself in a straightforward way to a weighted directed graph representation. Every transaction input and output gets a node in the graph. Every transaction is also represented by an identifier node in the graph. Each node corresponding to an input coin is connected to the transaction node with the appropriate value of the input coin as weight. Similarly, the transaction node is connected to each output node with a weight corresponding to the value of the output coin, *ignoring transaction fees*. For an illustrative example, see Figure 2a.

Since transaction outputs become the inputs of subsequent transactions, we obtain a transaction graph as desired. Once the transaction graph is obtained, we can transform the graph into an absorbing Markov chain as outlined in Section 2.4. Note that the Markov chain transition probabilities only depend on the nominal values of the transaction inputs and outputs if we assume no background knowledge and apply the transition matrix described in Equation (1).

Multi-edges between nodes are, by definition, not possible in a UTXO-based representation. It is possible, however, to treat UTXO-based systems as account-based ones by merging nodes that belong to the same public key. We describe how we handle account-based systems in Section 3.2.

3.2 Account-based Cryptocurrencies

Numerous cryptocurrencies use an account-based model, e.g., the most popular being Ethereum [59]. In this model, each transaction moves value between a source and a destination account. Accounts can be smart contracts or externally owned accounts (EOA), i.e., when a key pair controls the account. We represent both EOAs and smart contracts by nodes in the transaction graph. For every transaction, we add an edge between the source and the destination with a weight equal to the corresponding transacted amount, again *ignoring transaction fees*. Transforming the transaction graph to an absorbing Markov chain is done as in Section 2.4.

Multi-edges between nodes are possible in account-based systems. We handle these by merging all $u \rightarrow v$ transactions into a single (u, v) edge, with the weight being the sum of the transferred amounts. Note, however, that this representation still allows for opposite edges, i.e., (u, v) and (v, u), to exist simultaneously.

3.3 Modeling shielded pools

Certain cryptocurrencies apply privacy-enhancing techniques such as shielded pools, for example, Zcash [45].

Zcash is a privacy-preserving cryptocurrency that applies zero-knowledge proofs to keep funds untraceable. It has two types of addresses: transparent and shielded. Transactions from, to, or between shielded addresses are hidden from the public using zero-knowledge proofs and, thus, cannot be traced. We model all shielded addresses as a single address representing the shielded pool as a whole. Each shielding transaction transfers money to this imaginary address, while each de-shielding transaction is represented by the pool address sending money to the recipient. We do not model transfers between shielded addresses, as these convey no useful information: they would materialize as self-loops of the pool address, leaving absorbing probabilities unaffected.

However, not acknowledging the specific nature of the shielded pool could result in an unreasonably low untraceability score. Therefore, according to public knowledge, we model shielded pools not as a single node but as a collection of incoming transactions to the shielded pool. As a result, any output that is traced back to the pool ends up distributed between a large number of sources accurately representing the nature of the mixing characteristics of the transaction graph.

3.4 Temporality of Transactions

In a cryptocurrency system, each transaction has an implicitly assigned timestamp. Therefore, there is a total ordering for all transactions in a public ledger. One can either consider or disregard this temporal information when assessing traceability. Consequently, we consider two types of random walks on the cryptocurrency transaction graphs: *stationary* and *temporal*. We use the stationary version when evaluating a short time span, while we consider the temporal version more realistic for longer timespans.

Stationary transaction graphs. In a stationary random walk, we disregard the timestamp information of transactions. Specifically, whenever we traverse back on the transaction graph, we allow the random walk to continue along any of the transactions, regardless of temporal order. In the example of Figure 5, four transactions tx_1, tx_2, tx_3, tx_4 were included in the blockchain at times t_1, t_2, t_3, t_4, with $t_1 < t_2 < t_3 < t_4$. When a random walk starts at tx_2, we allow it to continue to either tx_1 or in tx_3, even though tx_2 happened earlier than tx_3. A stationary transaction graph captures the intuition that on a short enough time scale, the true source of an outgoing transaction of an address can be any of the incoming transactions, regardless of their temporality. This could more accurately represent the transfer of value between addresses in a short time frame.
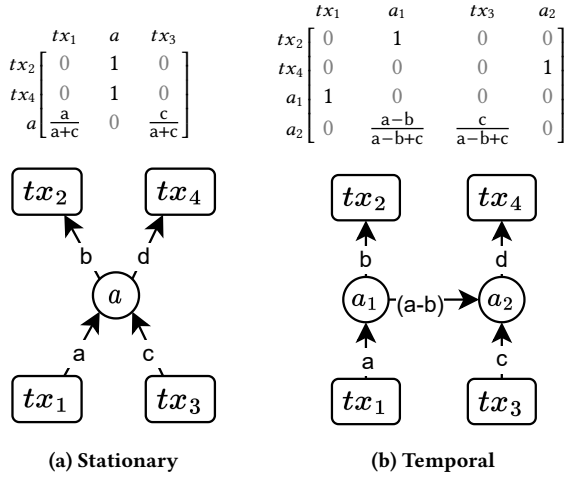
$$
\begin{array}{c|ccc}
 & tx_1 & a & tx_3 \\
\hline
tx_2 & 0 & 1 & 0 \\
tx_4 & 0 & 1 & 0 \\
a & \frac{a}{a+c} & 0 & \frac{c}{a+c}
\end{array}
\qquad
\begin{array}{c|cccc}
 & tx_1 & a_1 & tx_3 & a_2 \\
\hline
tx_2 & 0 & 1 & 0 & 0 \\
tx_4 & 0 & 0 & 0 & 1 \\
a_1 & 1 & 0 & 0 & 0 \\
a_2 & 0 & \frac{a-b}{a-b+c} & \frac{c}{a-b+c} & 0
\end{array}
$$

(a) Stationary   **(b) Temporal**

Figure 5: Temporality of transactions. Four transactions, two incoming and two outgoing target the same address a. Transactions are created in the order tx_1, tx_2, tx_3, tx_4. Weights on the edges represent the amount of the cryptocurrency transferred in the transactions. The stationary and temporal way of representing them as a transaction graph is presented, with transition probability matrices also included in the figure. The stationary transaction graph disregards temporality information, while the temporal transaction graph incorporates it. Note, as a result, the $tx_3 \to tx_2$ path is blocked for a random walk in the temporal graph.

Temporal transaction graphs. In contrast to stationary transaction graphs, temporal transaction graphs take the timing of transactions into consideration to block older outgoing transactions from backtracking to newer incoming transactions. This is achieved by creating a new node representation u_{i+1} for each time an account u receives an incoming transaction and adding a (u_i, u_{i+1}) edge from the previous representation u_i transferring the balance of u_i. This is reminiscent of the way UTXO-based systems work. Continuing with the example of Figure 5, starting from tx_2, we can only backtrack to tx_1 during the random walk. It is not possible anymore to backtrack to tx_3, since tx_2 is older than tx_3, i.e., $t_2 \le t_3$.

In UTXO-based systems, the transaction graph is inherently temporal, so there is seemingly no difference between temporal and stationary representations. However, public key reuse makes this question relevant even in UTXO-based graphs, as the outputs of transactions can be treated as belonging to the same account when they share a public key. Though the practice of public key reuse is discouraged [12], it often happens in practice due to ease of use [36].

4 APPLICATIONS

The flexibility of our proposed untraceability score allows for various applications. This section reviews a handful of these possible applications in detail. The two essential parameters of the proposed untraceability score $U_{S,G,\mathcal{A}}(v_i)$ are the considered source set S to which we wish to compute the absorbing probabilities, and the

assumed adversarial knowledge $\mathcal{A}$ applied by the analyst who computes the untraceability score. We characterise three applications by their S and $\mathcal{A}$ parameters.

Common-knowledge untraceability. This variant of the untraceability wants to establish an upper bound on the untraceability of unspent coins with regard to coinbase UTXOs. The common-knowledge untraceability score solely assumes the knowledge of the transaction graph and disregards any additional knowledge that might decrease the untraceability score. In Section 5, we thoroughly evaluate this type of untraceability score for several major cryptocurrency transaction graphs. The two parameters are:

- S = $\{v_i | v_i \in G \land deg_{in}(v_i) = 0\}$, i.e., all coinbase addresses.
- $\mathcal{A} = \mathbf{P}$, cf. Equation (1).

A cryptocurrency exchange. Cryptocurrency exchanges in most jurisdictions need to comply with anti-money laundering (AML) and know-your-customer (KYC) regulations. To that end, they collect, store, and process extensive information about the transaction graph. This additional knowledge allows them to cluster user addresses more effectively than only having access to the raw blockchain data. Most cryptocurrency exchanges do not allow money from illicit funds to be deposited. Let blackList $\subset V$ denote the set of nodes in the transaction graph that are blacklisted by the cryptocurrency exchange. A cryptocurrency exchange is interested in computing the probability distribution that a certain user fund residing at node $v \in V$ might contain funds from addresses in the blackList. Exchanges might apply the poison policy to transfer the taint between cryptocurrency transactions. The poison policy dictates that every transaction is tainted that has at least one dirty predecessor, no matter how many generations above [34].

- S = blackList.
- $\mathcal{A} = \mathbf{P}^{pois}$, cf. Equation (2).

Privacy-focused cryptocurrency wallet software. Most privacy-cautious users want to break links from a set of addresses that they previously exposed through some means, e.g., social media. Let taintedUserAddresses $\subset V$ denote this subset of the transaction graph. A privacy-focused wallet must ensure that new user addresses are hardly traceable to previously leaked ones. Our untraceability metric can assist in this application scenario.

- S = taintedUserAddresses.
- $\mathcal{A} = \mathbf{P}$, cf. Equation (1).

5 EVALUATION

In this section, we compute and evaluate the untraceability metric proposed in Section 2 for various cryptocurrencies. To make the presented values comparable, we compute the traceability metric for each cryptocurrency for the same time interval between 2021 February 1st and 8th. This one-week time interval might seem like a short period, but the resulting transaction graphs already produce considerable transition graphs with millions of nodes and edges, cf. Table 1 and Table 2. These transaction graphs require significant computational effort to measure the proposed untraceability metric. Running time optimization for even larger transaction graphs are beyond the scope of the current paper.
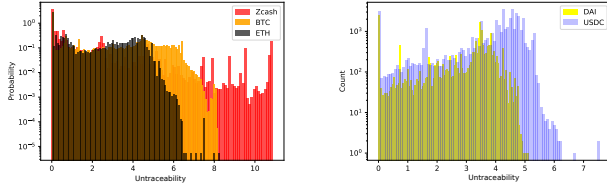
Figure 6: The distribution of untraceability of the unspent balances in the five studied cryptocurrencies (Bitcoin, Ethereum, Zcash (left), and DAI, USDC (right)) during the analyzed time period (2021 February 1-8).
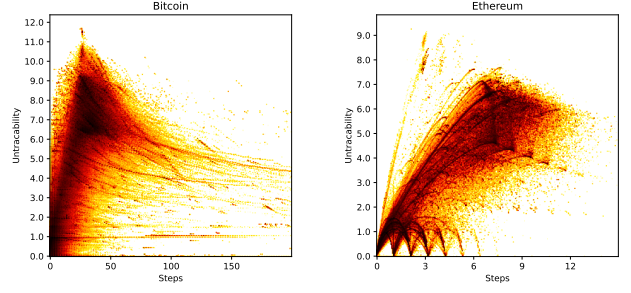

Figure 7: Scatter plots of untraceability and expected number of steps till absorption for Bitcoin (left) and Ethereum (right). Longer walks typically imply larger untraceability.

We study both the stationary and the temporal transaction graph variants for each network, as described in Section 3.4. We evaluate the traceability metric in the stationary variant in Section 5.1 while also enclosing the evaluation of the temporary transaction graphs in Section 5.3. For a summary of our results in the stationary variant, see Table 1 and Figure 6. The computations are carried out as described in Section 2.6. We set a value of $\delta_{\text{threshold}} = 0.001$ in Equation (9), which we think is a reasonable tradeoff between precision and efficiency when computing the untraceability of addresses.

In the evaluations presented in this Section, we treat Zcash as a special case. By taking a one-week snapshot of the transaction graph, most walks using de-shielding transactions are absorbed in the shielded pool, resulting in a small fungibility score. Since the shielding mechanism is a core component of Zcash, we instead modify the entropy values of Definition 1 to include one more step after reaching the shielded pool, regardless of whether the step is during the considered one-week interval. We do this to account for the uncertainty introduced by shielding transactions that happened before the considered time interval. Formally, calculating the modified entropy h' is done by observing that if

$$h = -\sum_i \log_2(p_i)p_i, \quad \text{then} \quad h' = h - p_k \sum_j \log_2(q_j)q_j, \quad (11)$$

where h is the original untraceability score of the output according to absorbing probabilities p_i, the value p_k is the probability of being absorbed at the node of the shielded pool, and q_j is the distribution of in-edge values of the shielded pool from before the considered time interval. While the modified score is only an approximation, we feel it more accurately reflects the untraceability properties of Zcash. For reference, we also include the original results with unmodified entropy values.

5.1 Stationary Transaction Graphs

5.1.1 UTXO-based Cryptocurrencies.

Bitcoin. We analyzed the Bitcoin transaction graph from block no. 668 548 till no. 669 613 and computed the fungibility of all 4 237 925 unspent coins in this transaction graph segment. Note that many UTXOs have 0 entropy. The mean entropy of coins is 3.96 bits of entropy, while the median is 4.07 bits. The maximum entropy of a coin in this transaction graph segment is 11.82 bits. On average, it takes 25.23 steps till an unspent coin is absorbed in a sink node, cf. Table 1.

Zcash. The evaluated transaction graph segment corresponds to the transactions from block no. 1 131 715 until no. 1 139 743 with 94 100 transparent UTXOs. We observed the highest fungibility variance (15.36) among all studied cryptocurrencies. Perhaps unsurprisingly, we found the UTXO with the highest fungibility (15.70) in Zcash. Moreover, on average, we witnessed the longest expected time till absorption (52.66) in Zcash.

Zcash unmodified untraceability. For completeness, we enclose the untraceability results for Zcash when we did not attach any shielding (t-to-z) transactions issued before our chosen time interval. Naturally, this decreases the measured untraceability of unspent UTXOs. The unmodified Zcash graph in our time window yielded an average untraceability 1.40 with median 0.68 and variance 1.99. Once again, these results underscore the importance of the shielded pool, since the introduction of the shielding transactions in Section 5 increased untraceability on average by 0.88.

5.1.2 Account-based Cryptocurrencies.

Ethereum. We created a transaction graph from all transactions in the Ethereum blockchain that transferred non-zero amounts of Ether from block no. 11 766 939 till no. 11 812 441. Out of 1 768 192 sink addresses, we observed a maximum untraceability of 10.86 bits of entropy. We found the least number of expected steps till absorption (mean 7.27, median 2.0) in the transaction graph, see Table 1. We attribute these phenomena to the nature of the account-based model, where large initial balances lead to walks being absorbed early with high probability.

ERC-20 token transaction graphs. We computed our untraceability measure for two stablecoins: USD Coin (USDC) and DAI. In the studied time interval these cryptoassets produced the smallest transaction graphs with 46 and 146 thousand addresses, respectively. Interestingly, both USDC and DAI random walks are absorbed after long random walks of expected length 26.39 and 16.40, respectively.

5.2 Expected Number of Steps Till Absorption

We computed the expected number of steps till absorption for every address with unspent balances in the studied transaction graph segments; see Figure 8. It is striking in Figure 8 how quickly Ethereum random walks are absorbed, especially compared to Bitcoin and

Table 1: Untraceability and the expected number of steps till absorption of unspent UTXOS (and addresses) in various cryptocurrencies. This table shows the results for the *stationary transaction graph* variants; see Section 3.4.
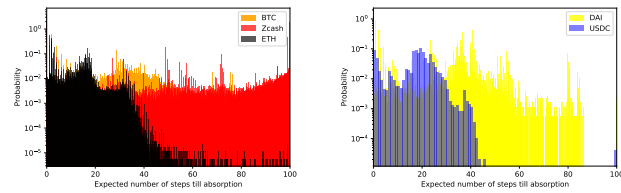
	$|V|$	$|E|$	Untraceability ($U(v)$)				Expected steps		
			Mean	Median	Variance	Max	Mean	Median	Variance
Bitcoin	8598k	12801k	3.96	4.07	10.12	11.82	25.23	23.56	615.32
Zcash	170k	748k	2.63	0.89	15.36	15.70	52.66	56.61	1755.94
Ethereum	3031k	5863k	3.45	3.94	5.97	10.86	7.27	2.0	80.51
DAI	46k	70k	2.60	3.30	1.98	5.14	26.39	33.69	414.49
USD Coin	146k	234k	3.32	4.10	2.91	8.12	16.40	18.35	74.13

Table 2: Untraceability and the expected number of steps till absorption of unspent UTXOs (and addresses) in various cryptocurrencies. This table shows the results for the *temporal transaction graph* variants; see Section 3.4.

	$|V|$	$|E|$	Untraceability ($U(v)$)				Expected steps		
			Mean	Median	Variance	Max	Mean	Median	Variance
Bitcoin	11441k	13679k	3.97	1.18	4.81	10.34	20.64	12.0	632.21
Zcash	946k	1042k	1.02	0.0	11.47	15.94	533.96	286.0	394k
DAI	172k	252k	3.13	3.23	4.00	8.64	37.39	23.80	1489.19
USD Coin	508k	723k	3.62	3.68	3.83	9.48	39.80	27.87	1577.23

Zcash. Furthermore, it is fascinating that the DAI transaction graphs contain significantly longer random walks on average (26.39) than the USDC transaction graph (16.40). This holds despite the fact that the USDC graph ($|V| = 146$k, $|E| = 234$k) is considerably larger than the DAI transaction graph ($|V| = 46$k, $|E| = 70$k).

We provide the scatter plots of untraceability and the expected number of steps till absorption for USDC, DAI, and Zcash in Figure 9. In USDC and DAI, there is a clear trend: the longer it is expected that a random walk is absorbed, the higher the untraceability it will achieve. However, larger random walks tend to decrease the achieved untraceability after a particular cutoff value, see Figure 9.

(a) USD Coin.

(b) DAI.

Figure 8: The expected number of steps till absorption for the studied cryptocurrencies (Bitcoin, Ethereum, Zcash (left), and DAI, USDC (right)).

On the other hand, Zcash exhibits completely different characteristics regarding untraceability and the expected lengths of random walks. First, Zcash does not need lengthy random walks to achieve high untraceability. We contribute this behavior to the shielded pool that effectively mixes together funds from different sources. Second, it seems that longer random walks decrease the untraceability score. Finally, we emphasize that Zcash achieves *one magnitude longer* random walks on average than any other analyzed cryptocurrency in this work, cf. Table 2.

(c) Zcash.
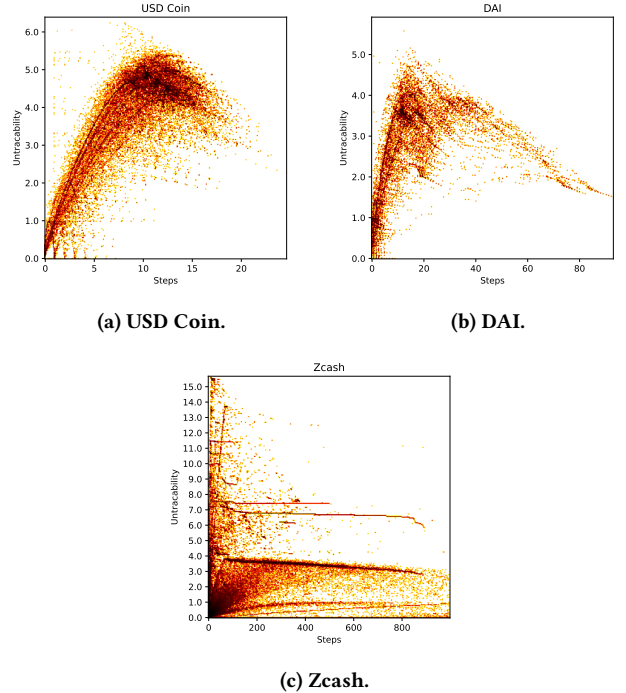
Figure 9: Contrasting entropies with the expected number of steps till absorption in the stationary transaction graphs.

5.3 Temporal transaction graphs

For each cryptocurrency discussed in the previous section, we analyze the temporal transaction graph as defined in Section 3.4. We

transform these temporal transactions graphs to absorbing Markov chains as discussed in Section 2. We measure the untraceability and the expected number of steps till absorption for all unspent balances in the transaction graph segments. Due to the introduced auxiliary nodes, we observe larger graphs and Markov chains for all five considered cryptocurrencies. We believe these auxiliary nodes greatly contribute to *reducing the untraceability* and the number of expected steps of all five cryptocurrencies. For the complete quantitative summary of results on temporal transaction graphs, see Table 2.

5.4 Discussion

Generally, we can observe many different untraceability characteristics in the studied cryptocurrency networks. Some of the results are largely unsurprising, such as the relatively low mean expected number of steps in account-based currencies, as account balances can quickly absorb random walks. Similarly, a significant untraceability variance in Zcash is expected, as the two different kinds of Zcash addresses (i.e., shielded and transparent) exhibit significantly different untraceability characteristics.

On the other hand, the low mean and median untraceability of Zcash is surprising, as un-shielded transactions use the same UTXO technology as Bitcoin. We observe significant differences between DAI and USD Coin, which use virtually identical technology, i.e., the ERC-20 token standard. These statistics are affected by the number of transactions included in the one-week interval, as more transactions provide more mixing capability. Still, the differences lead us to believe that typical real-world usage of the systems also heavily affects observed untraceability besides technological choices. This is especially pronounced when observing that USDC has the largest median untraceability (4.10 bits) and comparable mean untraceability to Ethereum, even though both are account-based systems, and the latter saw more than 25 times as much transaction activity over 20 times as many nodes in the studied one-week time interval.

The two main cryptocurrency networks, Bitcoin and Ethereum, achieved comparable mean and median untraceability scores, but Bitcoin ultimately achieved a higher score in both. The similarity is remarkable since their technological choices result in different transaction graph structures, most evident when measuring the expected number of steps till absorption; see also Figure 7.

6 LIMITATIONS AND EXTENSIONS

Limitations of our evaluation. When we take a one-week interval to study, the transaction graph's inputs are not particularly meaningful: they do not represent any known entities in general, unlike in some potential applications, see Section 4. Still, as our primary goal is to quantify the natural global mixing ability of each network, we believe that our experimental results demonstrate useful results towards this goal.

In our evaluation, certain services increase the untraceability score, even if the actual flow of money through the service can be unambiguously inferred from public knowledge. Consider star-shape transaction graphs, where numerous accounts send and receive coins from a single address, e.g., Satoshi dice[1], decentralized

---

[1]See: https://satoshidice.com/.

exchanges, e.g., Uniswap [1], etc. One could refine the transition matrix, see Figure 10 for an illustrative example, by connecting the in- and out-flow of money through these services, using the additional knowledge of which input address corresponds to which output address.

Obtaining a measure of anonymity. So far, we have exclusively focused on creating a formal model to calculate the untraceability in cryptocurrency transaction graphs. Intuitively, one can consider our proposed metric as a way to compute an upper bound on the uncertainty about the origins of unspent coins. However, in many occasions, one could apply heuristics [6, 36, 60], private knowledge, or additional information stored in the blockchain or obtained from other sources to reduce the uncertainty about the origins of coins. This knowledge could be incorporated into the definition of the transition probabilities, cf. Equation (1). Since these heuristics and additional information sources are strongly dependent on the adversaries, this notion could measure *the particular adversary's uncertainty* in assessing the origins of certain coins. Hence, this modified metric could be considered as an anonymity metric. Consider the transaction graph segment in Figure 10a, depicting a mixing transaction, e.g., Tornado Cash [40], with two deposit and two withdraw addresses. The two withdraws are equally likely to come from either deposit, resulting in the transition matrix presented in Figure 10b.

(a) Mixing transaction example using Tornado Cash [40].

	in_1	in_2	tx	o_1	o_2		in_1	in_2	tx	o_1	o_2
in_1	0	0	1	0	0		0	0	0	1	0
in_2	0	0	1	0	0		0	0	0	0	1
tx	0	0	0	0.5	0.5		0	0	0	0	0
o_1	0	0	0	0	0		0	0	0	0	0
o_2	0	0	0	0	0		0	0	0	0	0

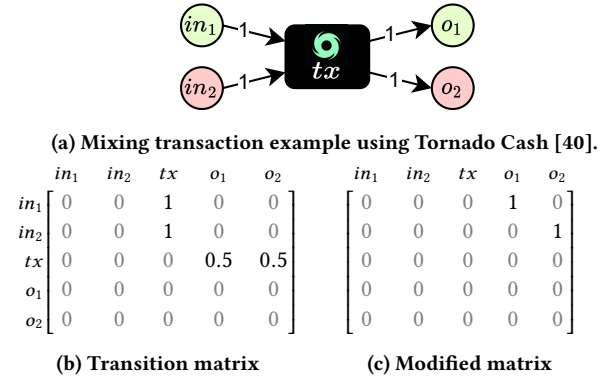**(b) Transition matrix**   **(c) Modified matrix**

Figure 10: A toy example illustrating how to incorporate epistemic adversarial knowledge when computing our untraceability measure. A mixing transaction (a) with its corresponding transition matrix (b). If one can utilize additional information or heuristics to match deposit and withdraw addresses belonging to the same entity, then the transition matrix can be modified to reflect this knowledge (c).

However, if one can infer that in_1 and in_2 belong to the same entities as o_1 and o_2 respectively, for example by pairing heuristics based on certain usage characteristics described in [6], then the transition matrix can be modified to reflect this knowledge, see Figure 10c.

7 RELATED WORK

There are several previous attempts in the literature to *quantify* traceability and related notions, such as anonymity, privacy, or

Approach	Range	Fine-grained	Decay	Tamper-resistant	Subjective	Efficiently Computable
k-anonymity [49]	✓	✗	✓	✓	✓	✓
Degree of anonymity [15, 47]	✓	✓	✓	✓	✓	✗
System's anonymity [18, 22]	✓	✓	✓	✓	✓	✗
Differential privacy [17]	✓	✗	✗	✓	✗	✓
Boltzmann score [30]	✓	✗	✓	✓	✓	✗
Wicht et al. [57]	✓	✗	✓	✓	✓	✓
Our approach	✓	✓	✓	✓	✓	✓

Table 3: Comparison of major approaches for measuring cryptocurrencies' (un)traceability or related notions applicable to transaction graphs, see Section 7. For a list of features, see Section 2.3.

fungibility in the context of cryptocurrencies. A commonly used privacy metric called k-anonymity, was introduced by Sweeney [49]. The notion of k-anonymity is sometimes also referred to as plausible deniability. It is extensively used to argue about the anonymity guarantees of a *single cryptocurrency transaction*, e.g., CoinJoin transactions [31] or cryptocurrency mixers [6]. However, k-anonymity does not generalize to multiple transactions meaningfully. Additionally, k-anonymity is inherently incapable of capturing the probabilistic nature of anonymity, i.e., the probability of each object in the anonymity set is uniform. The probabilistic nature of anonymity is captured by an information-theoretic approach proposed by Diaz et al. [15] and Serjantov et al. [47]. In their landmark works, the adversary outputs a probability distribution of guessing a target user in the anonymity set. The Shannon entropy of this distribution is considered as the degree of anonymity, i.e., the adversary's uncertainty about the target. Our work can be interpreted as generalising this entropic approach to the blockchain setting.

Boltzman score was proposed by LaurentMT[2] to quantify the anonymity guarantees of single Bitcoin transactions [30]. The Boltzman score computes all the possible groups of the transaction's outputs that can be obtained by grouping the transaction's inputs. The number of input-output assignments is then divided by the possible number of input-output assignments. This notion is only computable for small Bitcoin transactions, i.e., with a small number of inputs or outputs, because computing this score reduces to the NP-complete subset sum problem [29]. In contrast, in this work, we are interested in defining a privacy notion that is efficiently computable even for large real-world transaction graphs.

Very recently, Wicht et al. [57] introduced an untraceability and unlinkability metric for transaction graphs. They introduce a novel transaction graph model for cryptocurrencies, in which they model the transaction graph as a bipartite graph where one class of nodes consists of the inputs to transactions, while the other contains the transaction outputs. At a high level, their untraceability notion could be obtained by computing matchings on their own transaction graph model. Their notion can be seen as a generalization of the k-anonymity notion to the blockchain/transaction graph setting. Unfortunately, this metric is unable to express the probabilistic nature of the observed privacy notions. More importantly, they did not

implement and evaluate their proposed metric on cryptocurrency transaction graphs. Hence, unfortunately, we cannot compare our empirically evaluated untraceability metric with theirs.

Several *qualitative* frameworks grasp various notions of privacy and anonymity in anonymous communication systems by Backes et al. [4] and in cryptocurrency systems by Amarasinghe [3]. They define numerous flavours of unlinkability, indistinguishability, and anonymity in their corresponding settings. A significant limitation of these works is that they do not extend to multiple subsequent transactions, i.e., they solely analyze their privacy and anonymity notions for isolated transactions. Our main goal is to measure these notions for entire transaction graphs obtained from public ledgers.

To the best of our knowledge, no efficient metric has been proposed in the literature to measure the (un)traceability of money. A theory of fungibility was developed in [48]. However, it does not provide a way to measure fungibility in cryptocurrencies. The closest related work was done by Pontiveros et al. [42], where the authors proposed a notion called mint centrality. Mint centrality measures for a coin u, how many other coinbase coins (minted coins) can be traced back to u via transactions. Our metric is a strict generalization of mint centrality as we also measure how much a particular coinbase coin has contributed to a specific coin's fungibility throughout possibly many transactions.

8 CONCLUSION AND FUTURE WORK

In this work, we proposed a mathematical model based on absorbing Markov chains and Shannon entropy for a quantitative measure assessing the untraceability of cryptocurrencies. Additionally, we extensively evaluated the introduced untraceability measure in the same one-week time period on multiple UTXO- and account-based cryptocurrencies. Finally, we discussed how one can transform this untraceability measure into a degree of anonymity, incorporating exogenous adversarial knowledge into the computation of the untraceability measure. Future work entails studying and comparing the quantitative untraceability benefits of additional privacy-enhancing techniques and cryptocurrency designs, e.g., ring signatures applied in Monero [52], the various Coinjoin protocols of Bitcoin [20, 21], stealth addresses deployed on Ethereum [13, 55], confidential transactions [9, 41] or the Darksend design of Dash [16].

---

[2]See: Boltzman score: https://gist.github.com/LaurentMT/e758767ca4038ac40aaf.

REFERENCES

[1] Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, and Dan Robinson. Uniswap v3 core. *Tech. rep., Uniswap, Tech. Rep.*, 2021.

[2] Mansoor Ahmed, Ilia Shumailov, and Ross Anderson. Tendrils of crime: Visualizing the diffusion of stolen bitcoins. In *Graphical Models for Security: 5th International Workshop, GraMSec 2018, Oxford, UK, July 8, 2018, Revised Selected Papers 5*, pages 1–12. Springer, 2019.

[3] Niluka Amarasinghe, Xavier Boyen, and Matthew McKague. The cryptographic complexity of anonymous coins: A systematic exploration. *Cryptography*, 5(1):10, 2021.

[4] Michael Backes, Aniket Kate, Praveen Manoharan, Sebastian Meiser, and Esfandiar Mohammadi. Anoa: A framework for analyzing anonymous communication protocols. In *2013 IEEE 26th Computer Security Foundations Symposium*, pages 163–178. IEEE, 2013.

[5] J Barragan. Cryptocurrency crime and anti-money laundering report, february 2021. *CipherTrace*, 2021.

[6] Ferenc Béres, István A Seres, András A Benczúr, and Mikerah Quintyne-Collins. Blockchain is watching you: Profiling and deanonymizing ethereum users. In *2021 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS)*, pages 69–78. IEEE, 2021.

[7] Alex Biryukov, Daniel Feher, and Giuseppe Vitto. Privacy aspects and subliminal channels in zcash. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1813–1830, 2019.

[8] Joseph Bonneau, Arvind Narayanan, Andrew Miller, Jeremy Clark, Joshua A Kroll, and Edward W Felten. Mixcoin: Anonymity for bitcoin with accountable mixes. In *International Conference on Financial Cryptography and Data Security*, pages 486–504. Springer, 2014.

[9] Benedikt Bünz, Shashank Agrawal, Mahdi Zamani, and Dan Boneh. Zether: Towards privacy in a smart contract world. In *International Conference on Financial Cryptography and Data Security*, pages 423–443. Springer, 2020.

[10] Chainalysis. The 2020 state of crypto crime. 2020.

[11] Chainalysis. The chainalysis 2022 crypto crime report. 2022.

[12] Mauro Conti, E Sandeep Kumar, Chhagan Lal, and Sushmita Ruj. A survey on security and privacy issues of bitcoin. *IEEE Communications Surveys & Tutorials*, 20(4):3416–3452, 2018.

[13] Nicolas T Courtois and Rebekah Mercer. Stealth address and key management techniques in blockchain systems. In *ICISSP 2017-Proceedings of the 3rd International Conference on Information Systems Security and Privacy*, pages 559–566, 2017.

[14] Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. Flash boys 2.0: Frontrunning, transaction reordering, and consensus instability in decentralized exchanges. *arXiv preprint arXiv:1904.05234*, 2019.

[15] Claudia Diaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity. In *International Workshop on Privacy Enhancing Technologies*, pages 54–68. Springer, 2002.

[16] Evan Duffield and Daniel Diaz. Dash: A privacycentric cryptocurrency, 2015.

[17] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.

[18] Matthew Edman, Fikret Sivrikaya, and Bulent Yener. A combinatorial approach to measuring anonymity. In *2007 IEEE Intelligence and Security Informatics*, pages 356–363. IEEE, 2007.

[19] Giulia Fanti and Pramod Viswanath. Anonymity properties of the bitcoin p2p network. *arXiv preprint arXiv:1703.08761*, 2017.

[20] Ádám Ficsór, István András Seres, Yuval Kogman, and Lucas Ontivero. Wabisabi: Centrally coordinated coinjoins with variable amounts. 2021.

[21] Simin Ghesmati, Walid Fdhila, and Edgar Weippl. Sok: How private is bitcoin? classification and evaluation of bitcoin privacy techniques. In *Proceedings of the 17th International Conference on Availability, Reliability and Security*, pages 1–14, 2022.

[22] Benedikt Gierlichs, Carmela Troncoso, Claudia Diaz, Bart Preneel, and Ingrid Verbauwhede. Revisiting a combinatorial approach toward measuring anonymity. In *Proceedings of the 7th ACM Workshop on Privacy in the Electronic Society*, pages 111–116, 2008.

[23] Steven Goldfeder, Harry Kalodner, Dillon Reisman, and Arvind Narayanan. When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies. *arXiv preprint arXiv:1708.04748*, 2017.

[24] Ethan Heilman, Leen Alshenibr, Foteini Baldimtsi, Alessandra Scafuro, and Sharon Goldberg. Tumblebit: An untrusted bitcoin-compatible anonymous payment hub. *Cryptology ePrint Archive*, 2016.

[25] Ethan Heilman, Leen Alshenibr, Foteini Baldimtsi, Alessandra Scafuro, and Sharon Goldberg. Tumblebit: An untrusted bitcoin-compatible anonymous payment hub. In *Network and Distributed System Security Symposium*, 2017.

[26] Abraham Hinteregger and Bernhard Haslhofer. Short paper: An empirical analysis of monero cross-chain traceability. In *Financial Cryptography and Data Security: 23rd International Conference, FC 2019, Frigate Bay, St. Kitts and Nevis, February 18–22, 2019, Revised Selected Papers 23*, pages 150–157. Springer, 2019.

[27] John G Kemeny and J Laurie Snell. *Finite Markov chains: with a new appendix" Generalization of a fundamental matrix"*. Springer, 1983.

[28] Merve Can Kus Khalilov and Albert Levi. A survey on anonymity and privacy in bitcoin-like digital cash systems. *IEEE Communications Surveys & Tutorials*, 20(3):2543–2585, 2018.

[29] Jon Kleinberg and Eva Tardos. *Algorithm design*. Pearson Education India, 2006.

[30] LaurentMT. A python script computing the entropy of bitcoin transactions and the linkability of their inputs and outputs. https://gist.github.com/LaurentMT/e758767ca4038ac40aaf, 2017.

[31] Gregory Maxwell. Coinjoin: Bitcoin privacy for the real world. In *Post on Bitcoin forum*, 2013.

[32] Sarah Meiklejohn and Rebekah Mercer. Möbius: Trustless tumbling for transaction privacy. 2018.

[33] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M Voelker, and Stefan Savage. A fistful of bitcoins: characterizing payments among men with no names. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 127–140, 2013.

[34] Malte Möser, Rainer Böhme, and Dominic Breuker. Towards risk scoring of bitcoin transactions. In *Financial Cryptography and Data Security: FC 2014 Workshops, BITCOIN and WAHC 2014, Christ Church, Barbados, March 7, 2014, Revised Selected Papers 18*, pages 16–32. Springer, 2014.

[35] Malte Möser and Arvind Narayanan. Effective cryptocurrency regulation through blacklisting. *Preprint*, 2019.

[36] Malte Möser and Arvind Narayanan. Resurrecting address clustering in bitcoin. *arXiv preprint arXiv:2107.05749*, 2021.

[37] Malte Möser, Kyle Soska, Ethan Heilman, Kevin Lee, Henry Heffan, Shashvat Srivastava, Kyle Hogan, Jason Hennessey, Andrew Miller, Arvind Narayanan, et al. An empirical analysis of traceability in the monero blockchain. *arXiv preprint arXiv:1704.04299*, 2017.

[38] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, page 21260, 2008.

[39] Arvind Narayanan and Malte Möser. Obfuscation in bitcoin: Techniques and politics. *arXiv preprint arXiv:1706.05432*, 2017.

[40] Alexey Pertsev, Roman Semenov, and Roman Storm. Tornado cash privacy solution version 1.4. 2019.

[41] Andrew Poelstra, Adam Back, Mark Friedenbach, Gregory Maxwell, and Pieter Wuille. Confidential assets. In *International Conference on Financial Cryptography and Data Security*, pages 43–63. Springer, 2018.

[42] Beltran Borja Fiz Pontiveros, Mathis Steichen, and Radu State. Mint centrality: A centrality measure for the bitcoin transaction graph. In *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 159–162. IEEE, 2019.

[43] Alfréd Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, volume 4, pages 547–562. University of California Press, 1961.

[44] Tim Ruffing, Pedro Moreno-Sanchez, and Aniket Kate. Coinshuffle: Practical decentralized coin mixing for bitcoin. In *European Symposium on Research in Computer Security*, pages 345–364. Springer, 2014.

[45] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474. IEEE, 2014.

[46] István András Seres, Dániel A Nagy, Chris Buckland, and Péter Burcsi. Mixeth: efficient, trustless coin mixing service for ethereum. *Cryptology ePrint Archive*, 2019.

[47] Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In *International Workshop on Privacy Enhancing Technologies*, pages 41–53. Springer, 2002.

[48] Jamsheed Shorish, Matt Stephenson, and Michael Zargham. A practical theory of fungibility. 2021.

---

[3]See: https://blockchair.com.

[49] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[50] Muoi Tran, Loi Luu, Min Suk Kang, Iddo Bentov, and Prateek Saxena. Obscuro: A bitcoin mixer using trusted execution environments. In *Proceedings of the 34th Annual Computer Security Applications Conference*, pages 692–701, 2018.

[51] Luke Valenta and Brendan Rowan. Blindcoin: Blinded, accountable mixes for bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 112–126. Springer, 2015.

[52] Nicolas Van Saberhagen. Cryptonote v 2.0, 2013.

[53] Friedhelm Victor. Address clustering heuristics for ethereum. In *Financial Cryptography and Data Security: 24th International Conference, FC 2020, Kota Kinabalu, Malaysia, February 10–14, 2020 Revised Selected Papers 24*, pages 617–633. Springer, 2020.

[54] Anton Wahrstätter, Jens Ernstberger, Aviv Yaish, Liyi Zhou, Kaihua Qin, Taro Tsuchiya, Sebastian Steinhorst, Davor Svetinovic, Nicolas Christin, Mikolaj Barczentewicz, et al. Blockchain censorship. *arXiv preprint arXiv:2305.18545*, 2023.

[55] Anton Wahrstätter, Matthew Solomon, Ben DiFrancesco, Vitalik Buterin, and Davor Svetinovic. Basesap: Modular stealth address protocol for programmable blockchains. *IEEE Transactions on Information Forensics and Security*, 2024.

[56] Zhipeng Wang, Xihan Xiong, and William J Knottenbelt. Blockchain transaction censorship:(in) secure and (in) efficient? In *The International Conference on Mathematical Research for Blockchain Economy*, pages 78–94. Springer, 2023.

[57] François-Xavier Wicht, Zhipeng Wang, Duc V Le, and Christian Cachin. A transaction-level model for blockchain privacy. *Cryptology ePrint Archive*, 2023.

[58] Fredrik Winzer, Benjamin Herd, and Sebastian Faust. Temporary censorship attacks in the presence of rational miners. In *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 357–366. IEEE, 2019.

[59] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014.

[60] Mike Wu, Will McTighe, Kaili Wang, Istvan A Seres, Nick Bax, Manuel Puebla, Mariano Mendez, Federico Carrone, Tomás De Mattey, Herman O Demaestri, et al. Tutela: An open-source tool for assessing user-privacy on ethereum and tornado cash. *arXiv preprint arXiv:2201.06811*, 2022.

[61] Haaroon Yousaf, George Kappos, and Sarah Meiklejohn. Tracing transactions across cryptocurrency ledgers. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 837–850, 2019.

A CHANGE OF MEASURED UNTRACEABILITY WITH WALK LENGTH

Last but not least, we analyzed the effect of time on untraceability. More precisely, starting from every absorber node, e.g., coinbase addresses, we computed the achieved untraceability. Afterwards, we proceeded in a random walk from the absorber node and recorded how untraceability changes, see Figure 11. This experiment emulates the effect of subsequent transactions on untraceability. As Figure 11 shows, untraceability rapidly increases after the first handful of transactions. However, it quickly saturates, and further transactions seem to decrease untraceability.
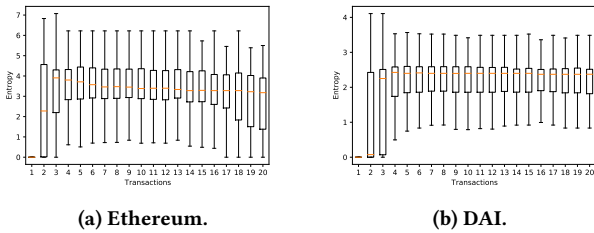

(a) Ethereum. (b) DAI.

Figure 11: The change in untraceability through time.