MoneroVis: Visual Analytics for Exploration and Traceability in a Privacy-Preserving Blockchain

Daniel Nikbakht Tehrani
TU Wien
Vienna, Austria
dnvie@icloud.com

Natkamon Tovanich
TU Wien
Vienna, Austria
natkamon.tovanich@tuwien.ac.at

***Abstract*—Monero is a privacy-preserving cryptocurrency that obscures the sender, receiver, and amount of each transaction using ring signatures, stealth addresses, and RingCT. We present MoneroVis, a visual analytics system for exploring and analyzing such transaction data. The system supports multi-level analysis, from block and mempool monitoring to detailed inspection of transaction structures. By integrating network visualization with heuristic cues, MoneroVis enables users to infer likely true inputs and reason about transaction flows under uncertainty. We demonstrate the usability of our system through a real-world forensic case study and a user study with domain experts.**

***Index Terms*—Visual Analytics, Monero, Privacy-Preserving Cryptocurrencies, Transaction Networks, Blockchain Data**

I. Introduction

Cryptocurrencies such as Bitcoin rely on Unspent Transaction Outputs (UTXOs) to record the creation and transfer of value on a distributed ledger. With this model, transactions spend previously created outputs and generate new ones, creating a traceable transaction history. While this design supports transparency and verifiability, it also makes transaction activities visible to anyone observing the blockchain.

As a result, UTXO-based blockchains offer pseudonymity rather than true anonymity. Users interact through public addresses, but once an address can be associated with a real-world identity, their transaction activities can be traced and analyzed. This traceability has important privacy implications, as analysts can reconstruct money trails, detect behavioral patterns, perform address clustering, and, in some cases, associate addresses with real-world entities.

Monero was introduced to address these limitations [1]. Like Bitcoin, it follows the UTXO model but implements additional cryptographic mechanisms to conceal the sender, receiver, and transaction amount. In particular, it employs ring signatures to obscure the true input among decoys, stealth addresses to hide the recipient, and Ring Confidential Transactions (RingCT) to encrypt transferred amounts [2], [3]. These mechanisms significantly improve anonymity but also make transaction tracing substantially more difficult, since tracing the flow of funds requires heuristic reasoning about which inputs in the ring are genuine spends and which serve as decoys.

A broad range of forensic and blockchain analysis tools has been developed to support transaction tracing, illicit activity detection, and network analysis. However, these approaches do not readily extend to privacy-preserving blockchains such as Monero. In this setting, tracing the flow of funds requires reasoning under uncertainty, as analysts need to infer which outputs correspond to real spending and which serve as decoys.

To address this gap, we present MoneroVis, an interactive visual analytics system for exploring and tracing Monero transactions. The system supports macro-level block and mempool monitoring, a micro-level transaction graph for inspecting ring structures, and a decoy-oriented tracing view that highlights output reuse and supports forward tracing of transactions. This paper makes the following contributions:

1) a visual analytics system for exploring Monero structures and transaction networks across multiple levels of detail;
2) a case study demonstrating how the system supports the WannaCry 2.0 transaction tracing on Monero; and
3) a user study with domain experts evaluating the usability of the tool for supporting understanding and investigating privacy-preserving blockchain transactions.

II. Background on Monero Traceability

Monero is designed to provide strong privacy guarantees through *untraceability* (anonymity within a set of candidate outputs) and *unlinkability* (the inability to link transactions to the same recipient) [4]. Nonetheless, prior research has shown that various heuristics and implementation flaws weaken these guarantees in practice.

Prior to late 2017, Monero traceability was largely driven by structural flaws in the protocol's early implementation:

- **Zero-Mixin Transactions:** Until 2016, users could opt to include zero decoys, making the spent output explicitly identifiable [4], [5]. This enabled the so-called "cascade attack," where identifying a real input in one transaction allows analysts to eliminate it as a decoy in others through a process of elimination [4], [5].
- **The Guess-Newest Heuristic:** Research found that because users typically spend coins soon after receiving

This work was supported by the CHIST-ERA grant *FairOnChain: Fair and modular blockchain data infrastructure for open science and society* by Agence Nationale de la Recherche (ANR-23-CHRO-0002).

them, the real input in a ring was the newest output roughly 80% to 98.5% of the time [4], [5]. This was due to a mismatch between actual spending behavior and the protocol's initial decoy-sampling algorithms [5].

The introduction of RingCT in January 2017 obscured transaction amounts, making value-based tracing obsolete, but new forensic methods emerged.

- **Output Merging Heuristic:** This heuristic assumes that if multiple transaction inputs reference outputs from the same prior "source" transaction, those outputs are likely the real funds being spent by a single entity consolidating their holdings [4].
- **Cross-Chain Traceability:** Analysts exploit currency hard forks (e.g., MoneroV, Monero Original), where outputs spent on both chains share the same key image. By comparing ring members across chains, the true input can be identified as their intersection [6].

Recent forensics focus on implementation bugs and behavioral fingerprinting:

- **The 10-Block Decoy Bug:** A significant flaw was discovered in the `wallet2` library (used by many wallets between 2019 and 2023) that prevented the selection of decoys exactly 10 blocks old. As a result, any ring member with this exact age can be identified as the true spend with near certainty [7].
- **Coinbase Decoy Heuristic:** Since regular users rarely possess coinbase outputs (mining rewards), their inclusion in input rings strongly suggests they are decoys. Moreover, because miners typically consolidate such outputs in transactions with many inputs, coinbase outputs appearing in transactions with few inputs can be treated as decoys in most cases [7].

While Monero's design increases the difficulty of transaction analysis, these heuristics demonstrate that partial traceability remains possible through structural, temporal, and behavioral signals that provide probabilistic cues and reduce the effective anonymity set.

MoneroVis visualizes multiple probabilistic heuristic cues without assigning calibrated confidence scores or asserting a true spent input, since ground-truth inputs and thus reliable false-positive rates are generally unavailable for real transactions. Instead, the value of the tool lies in supporting the exploration of money flows and reasoning under uncertainty.

III. Related Work

Blockchain enables the analysis and visualization of financial transaction data at an unprecedented scale. However, the large volume of data and cryptographic mechanisms designed to preserve user privacy introduce significant analytical challenges. A systematic review reported that most existing visualization approaches focus on transparent blockchains such as Bitcoin and Ethereum [8], leaving privacy-preserving systems such as Monero and Zcash comparatively underexplored.

Existing visualizations explore transaction graphs, address clustering, and money flows to support tasks such as transaction monitoring and fraud detection. *SilkViser* proposes visual metaphors for exploring blockchain data [9]. *BitConduite* enables clustering of Bitcoin entities based on transactional behavior [10], while *BitExTract* visualizes the evolution of user activity patterns [11]. *BitVis* provides interactive interfaces for analyzing relationships between accounts and transactions [12]. Forensic platforms such as *BlockSec*, *Chainalysis*, and *Iknaio* provide infrastructure for tracing monetary flows, supporting compliance and forensic investigations [13]–[15].

Several works visualize cryptocurrency transaction flows. *SilkViser* uses Sankey diagrams to show fund merging and splitting [9], while *BitConeView* and *Tendrils of Crime* trace tainted Bitcoin flows across blocks [16], [17]. *BlockchainVis* reveals obfuscation patterns such as mixing services [18], and McGinn et al. identify structures associated with transaction-rate attacks [19]. *Bubblemaps* visualizes major token-holder clusters and transfers to help detect governance manipulation and insider coordination [20]. These systems visualize transaction flow networks through directly observable relationships or deterministic tracing rules. However, none of them address privacy-preserving blockchains, such as Monero, where the inherent probabilistic traceability remains an open challenge.

Compared with prior work, MoneroVis integrates mempool, block, transaction, and flow visualizations with heuristic reasoning to support the analysis of privacy-preserving transaction networks. It enables analysts to inspect ring members, compare heuristic evidence to infer likely true spends, and explore plausible transaction flows under uncertainty.

IV. Data–Users–Tasks Analysis

Following the Data–Users–Task design methodology [21], we design MoneroVis considering the relationships among data characteristics, target users, and analytical tasks.

A. Data

Monero transaction data follows the UTXO model but differs due to its privacy-preserving mechanisms [22]. Each transaction includes inputs represented as ring signatures, where the true spending output is hidden among multiple decoys, and outputs as one-time stealth addresses. This results in an ambiguous structure in which each input corresponds to a set of possible source outputs, posing unique challenges for transaction tracing.

B. Users

MoneroVis targets both domain experts investigating transaction activity and advanced users seeking to understand privacy-preserving mechanisms in practice. We identify three primary user groups:

U1 **Forensic analysts**, who aim to trace suspicious transactions and identify potential links between activities. Unlike Bitcoin, Monero analysis requires reasoning under uncertainty to identify true spending inputs. These users require tools to filter out decoy noise and identify statistical anomalies, such as output merging, in which a single entity consolidates funds from multiple sources.

U2 **Blockchain analysts and researchers**, who study transaction patterns, privacy mechanisms, and network behavior. Rather than tracing specific funds, they evaluate Monero's privacy protections, for example, by analyzing decoy distribution, assessing whether temporal heuristics reduce anonymity, and uncovering systemic patterns, such as wallet fingerprinting or biases in decoy selection.

U3 **Advanced users and developers**, who aim to understand Monero's transaction structure and how its mechanisms operate in practice. They use the system for exploratory analysis to observe how transactions incorporate decoy outputs and to develop an intuition for tracing monetary flows in a privacy-preserving setting.

C. Tasks

Based on the data characteristics and user needs, we identify analysis tasks for Monero transaction data, ranging from macro-level monitoring to micro-level transaction analysis and uncertainty-aware multi-hop transaction flow tracing.

At the macro level, analysts monitor overall network activity in the mempool and examine block composition to identify patterns and detect potential anomalies.

T1.1 **Overview network activity:** Users explore the composition of transactions in the mempool and blocks, such as transaction volume, size, and fee distribution, to understand current network conditions and block composition.

T1.2 **Detect patterns and anomalies:** Users identify unusual patterns such as spikes in fees, large transactions, or irregular mempool or block characteristics that may indicate spam, congestion, or suspicious behavior.

At the micro level, analysts inspect individual transactions to reason about uncertainty introduced by privacy mechanisms and reconstruct potential flows of funds.

T2.1 **Identify and search specific blocks and transactions:** Users identify and search for blocks and transactions of interest, often based on external information such as known block height and transaction hashes.

T2.2 **Lookup transaction details:** Users examine the structure of a selected transaction, including inputs, ring members, and outputs, to understand how value is represented and obfuscated.

T2.3 **Identify likely true spending input:** Users assess which ring member is most likely to be the true spending input by applying heuristic cues, such as temporal patterns of decoy selection and known traceability heuristics.

T2.4 **Explore transaction relationships:** Users analyze structural links between transactions by identifying which transaction produced a given ring member and examining where outputs are later referenced as ring members in other transactions.

These tasks capture the multi-level analytical workflow required for investigating privacy-preserving transaction networks and motivate the design of MoneroVis.

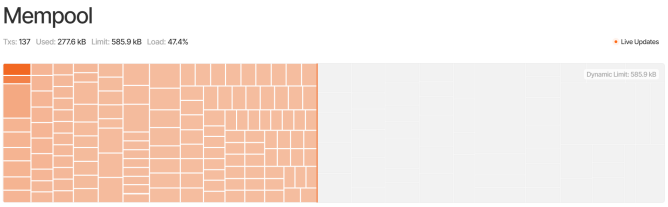

Fig. 1. The mempool view visualizes transactions as a treemap.

V. MoneroVis

MoneroVis consists of five views that support multi-level analysis from block and mempool monitoring to detailed transaction inspection and uncertainty-aware tracing. It supports Monero consensus protocol versions through v16 (the latest at the time of writing), with the displayed information adapted to version-specific features. The tool is available online at https://monerovis.com/, while the source code is released under the MIT License on GitHub at https://github.com/dnvie/MoneroVis.

Implementation. MoneroVis is implemented as a web-based visual analytics system. The frontend is built as a Single Page Application using Angular. The interactive visualizations are implemented using D3.js. The backend services are implemented in Go, providing RESTful APIs for data processing and query handling, as well as a WebSocket service for real-time network data.

Monero data is extracted from a locally synchronized node and stored in an SQLite database. Indexes on transaction hashes, output identifiers, and ring-member references support efficient forward tracing from outputs and backward tracing to their originating transactions.

A. Mempool View

The mempool view provides an overview of unconfirmed transactions awaiting inclusion in the next block (T1.1). It supports monitoring of global transaction activity and identifying patterns such as congestion or high-fee competition, which are key indicators of network activity (T1.2).

Visual Encoding. The mempool is visualized as a treemap (Fig. 1), with each rectangle representing a transaction. The rectangle size encodes transaction size, allowing users to identify large transactions at a glance. Color intensity (orange hue) encodes the relative transaction fee, with darker colors indicating higher fees.

The treemap is scaled relative to the estimated capacity of the next block. Since Monero uses a dynamic block size determined by the median block weight, the view conveys both occupied capacity (i.e., total size of transactions currently in the mempool) and remaining capacity available for inclusion.

Interactions. Users can interact with the treemap by selecting an individual rectangle to inspect a selected transaction in detail (T2.1). To complement the visual overview, the interface includes tabular views of mempool transactions and recent blocks, displaying attributes such as transaction hash, fee, size, and number of inputs and outputs.

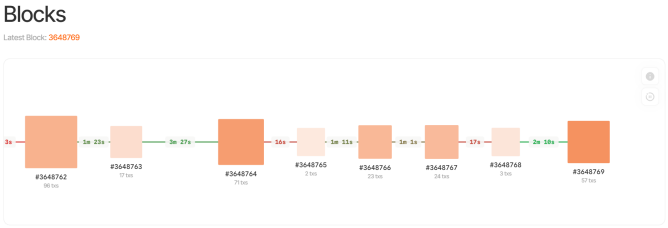

Fig. 2. The block sequence view visualizes a sequence of blocks as a connected chain.

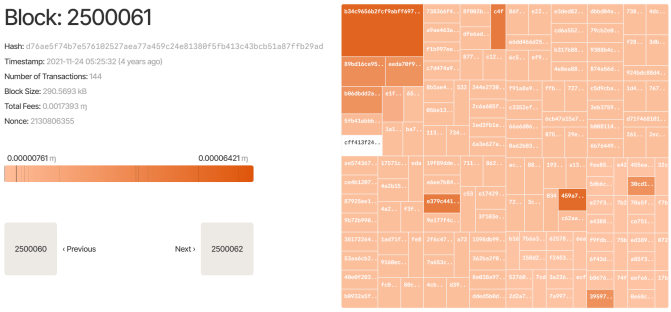

Fig. 3. The block view visualizes transactions within a block using a treemap.

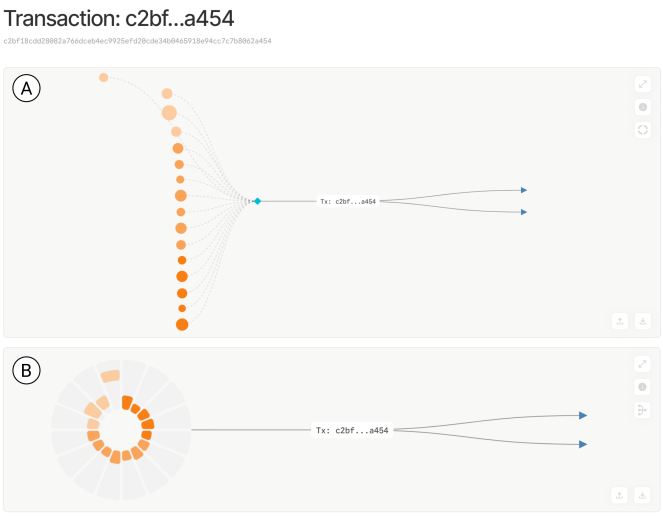

Fig. 4. The block details view in (A) graph mode visualizes the transaction graph as a node-link diagram, while (B) ring mode represents each transaction input as a circular arc, with ring members shown as segments.

Real-time Updates. The mempool view is updated in real time via a WebSocket connection. Incoming transactions are buffered and rendered in batches at regular intervals. This approach balances responsiveness and rendering efficiency to ensure smooth interactions during bursts of activity.

B. Block Sequence View

The block sequence view displays a paginated sequence of the latest blocks (Fig. 2). This view allows users to assess temporal dynamics and block-level metrics at a glance (T1.1) to detect irregular block production patterns, such as unusually long block times or spikes in transaction activity (T1.2).

Visual Encoding. Each block is represented as a rectangle. The rectangle size encodes the number of transactions in a block, allowing users to identify high-activity blocks. Color intensity (orange hue) encodes the total block reward, including both the base reward and transaction fees.

Blocks are connected to their predecessors through edges that encode temporal properties. Edge length represents the time interval between consecutive blocks, while edge color indicates deviation from the target block time ($\approx$ 2 minutes). Blocks close to the target interval are shown in green, whereas larger deviations are highlighted in red.

Interactions. Users can select a block of interest to navigate to its block details view (T2.1, Fig. 3). A complementary tabular representation lists blocks on the current page, including attributes such as block height, hash, transaction count, reward, and age. Selecting a row provides an alternative entry point to the block details view (T2.1).

C. Block Details View

The block details view displays a list of transactions in a single block, along with their associated attributes, including transaction size and fees (T1.1).

Visual Encoding. Transactions within a block are visualized using a treemap (Fig. 3), where each rectangle represents a transaction. The rectangle size encodes transaction size, and color intensity (orange hue) encodes transaction fee. The coinbase transaction is visually distinguished by a neutral color (white) because it does not include transaction fees.

To support analysis of fee distribution, the color legend includes a gradient slider with discrete markers, with each marker representing a transaction fee level within the block.

A complementary table below lists all transactions within the block, including transaction hash, number of inputs and outputs, fee, and size.

Interactions. Users can interactively explore transactions by hovering over fee markers on the legend to highlight corresponding transactions in the treemap, enabling quick identification of how transaction fees are distributed (T1.2). Selecting a transaction opens its detailed view (T2.1).

D. Transaction Details View

The transaction details view provides a graph-based visualization of Monero transactions, revealing the structure and temporal patterns of ring signatures (T2.2). The view supports the detection of likely input spends (T2.3) and reasoning about transaction flows under uncertainty (T2.4).

Graph Mode. In graph mode, each ◆ input ring is represented as a diamond-shaped node connected to ● ring member nodes (Fig. 4, A). These edges are rendered as dashed lines to indicate uncertainty, as only one ring member corresponds to the true spend, while the others are decoys.

Ring members are arranged horizontally by the age of their parent transaction, with older outputs placed further to the

left. This layout reflects the temporal distribution of candidate inputs and supports reasoning about likely spending behavior. Color encodes temporal recency using three levels of orange, distinguishing ● recent ($\leq$ 1,440 blocks), ● moderately recent ($\leq$ 21,600 blocks), and ● older outputs. Node size encodes reuse frequency, indicating how often a given output has been selected as a ring member across the blockchain.

To support inference of the true input (T2.3), we implemented and visually encoded a set of heuristics from [7] in distinct colors: ● coinbase outputs (dark blue), ● shared parent outputs (violet), ● 10-block anomaly (light blue), and ● unspendable outputs (red).

Ring Mode. Ring mode provides an alternative representation of the same data using a circular layout (Fig. 4, B). Each input is visualized as an arc, with ring members represented as segments of the arc.

Visual encodings in the ring mode are consistent with those in the graph mode: color and size retain their meaning, while temporal information is mapped radially, with older outputs positioned farther from the center.

Interactions. The view supports interactive exploration of transaction relationships (T2.4) through both backward and forward tracing. In backward tracing, users navigate the graph by selecting a ● ring member, which appends the parent transaction to the left of the visualization. In forward tracing, users select an ▶ output node to open the decoy map view, allowing them to explore subsequent transactions in which the given output is a ring member.

E. Decoy Map

The decoy map visualizes how a selected output is reused as a ring member across the blockchain, allowing users to explore its relationships with multiple transactions and reason about its potential role in transaction flows (T2.3, T2.4).

Visual Encoding. The decoy map is represented as a force-directed graph, where different node types encode distinct entities: ▬ transaction (rectangular), ▲ output (triangular), ● input (circle), and ● ring member (dot).

The ▲ selected output is centered in the graph and highlighted in red. Its ▬ parent transaction, along with associated ● inputs, ▲ outputs, and ● ring members, are in yellow, while surrounding structures are rendered in gray. This visual hierarchy supports focus+context exploration of decoy relationships.

Scalability. Unlike the other views, which use SVG elements, the decoy map is rendered using HTML5 Canvas to avoid the overhead of thousands of DOM elements. This enables efficient rendering and interactive performance during exploration involving several thousand nodes.

Interaction. By default, the view displays only inputs that reference the selected output as a ring member, allowing users to focus on its immediate usage context (T2.3). An optional *all inputs* mode expands the graph to include all inputs and associated ring members for visible transactions.

Users can pan and zoom the graph interactively and select nodes to highlight their relationships. Highlighting emphasizes connected elements while dimming unrelated ones, supporting focused analysis. Labels for ring members are shown on demand to reduce visual clutter.

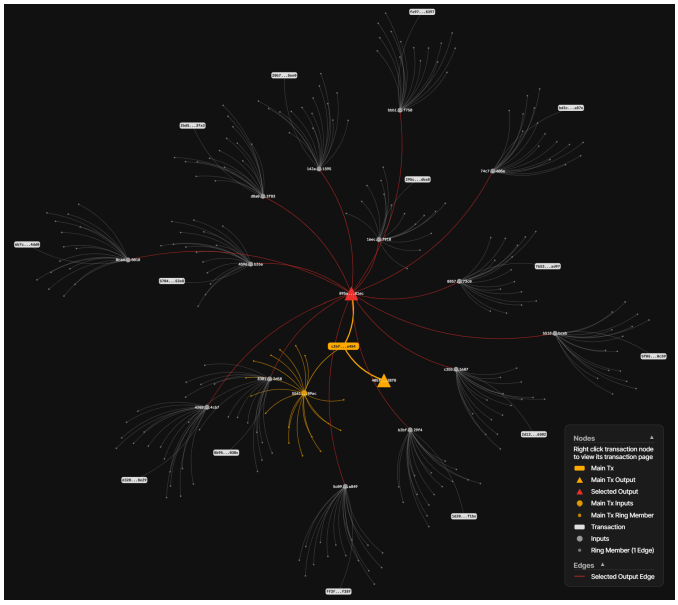

Fig. 5. The decoy map visualizes how a selected output is reused as a ring member across multiple transactions using a force-directed node-link diagram.

VI. Case Study

We demonstrate the application of MoneroVis in a real-world forensic scenario by tracing illicit funds associated with the WannaCry 2.0 ransomware [23].

To analyze this case, analysts applied a *poisoned-output attack*, also known as the *Eve-Alice-Eve (EAE) attack* [23], [24]. By using known transaction outputs (e.g., from exchange withdrawals) as anchors, they identified transactions in which multiple such outputs reappear as ring members. Because decoy outputs are selected from a large global pool, these co-occurrences provide strong probabilistic evidence of shared ownership, enabling the reconstruction of transaction flows.

A. Tracing WannaCry 2.0 Monero Transactions

In May 2017, the WannaCry 2.0 ransomware infected global computer networks, demanding ransom payments in Bitcoin (BTC). The attackers, attributed to the Lazarus Group, subsequently laundered 52 BTC by converting it into about 820 Monero (XMR) via the ShapeShift exchange on August 3, 2017. Using the ShapeShift API, analysts retrieved eight Monero transactions associated with this laundering and flagged their outputs as entry points for tracing.

Following the conversion, the attackers consolidated funds by spending multiple outputs in single transactions. Using the *output merging heuristic*, such consolidation events can be identified when multiple flagged outputs appear together as inputs, indicating a high likelihood of real spending rather than a decoy. For example, one identified transaction that spent five flagged outputs has an estimated probability of 2.6×10^{-21} (or

nearly zero) of occurring by chance, providing strong evidence of common ownership.

Finally, on November 2, 2017, the attackers converted XMR back into Bitcoin Cash (BCH) via ShapeShift. At the time, ShapeShift used publicly visible payment IDs, allowing analysts to link Monero deposits to BCH withdrawals and reconstruct the flow of funds.

B. Interactive Tracing Workflow on MoneroVis

To trace the transaction flows of WannaCry 2.0 funds on the Monero network, we implemented an *output tracing panel*, accessible via the top-left interface next to the search input. This component supports tracing of transactions involved in *poisoned-output attacks* (T2.3, T2.4) through two modes: (1) an *auto mode* that automatically detects consolidation patterns and constructs transaction flows, and (2) an *advanced mode* that provides intermediate results, allowing analysts to manually identify and construct output consolidation steps.

In the *auto mode*, transactions are automatically detected and mapped by consolidating poisoned outputs. The tracing process proceeds as follows:

1) **Inputting Initial Transactions:** The user provides a set of initial transaction hashes (Fig. 6, A). The system extracts all resulting outputs and marks them as the initial set of *poisoned outputs*.
2) **Detecting Consolidations:** The system retrieves all subsequent transactions that reference these outputs as ring members and identifies intersections across multiple outputs. Transactions appearing in multiple tracking lists are flagged as *consolidations*, indicating likely merging of controlled funds (Fig. 6, B). This process is applied recursively to newly generated outputs until no further intersections are found (Fig. 6, C–D).
3) **Visualizing the Transaction Graph:** The detected consolidation events are visualized as a transaction graph, where initial transactions are positioned on the left (Fig. 6, A) and successive consolidation rounds extend to the right (Fig. 6, B–D). The green-highlighted edges represent the inferred flow of funds.

Applied to the WannaCry 2.0 case study, Fig. 6, A shows the eight initial Monero transactions from ShapeShift to the attackers, each producing three outputs that form the initial set of poisoned outputs. Any subsequent transaction that includes multiple poisoned outputs as ring members is flagged as a consolidation transaction. In Fig. 6, B, three such transactions are detected, and their outputs are used as the next set of poisoned outputs. The process is repeated by identifying all future transactions referencing multiple of those outputs as ring members. This yields two new transactions in Fig. 6, C. A final iteration results in a single consolidation transaction (Fig. 6, D), indicating the end of the observable money flow.

By following these patterns, analysts can observe how funds propagate over time and reconstruct transaction chains, despite the uncertainty introduced by privacy-preserving mechanisms.

VII. User Study

We evaluate MoneroVis through a user study with domain-relevant participants to assess its usability for exploration and transaction tracing tasks. Each session lasted approximately one hour and followed a semi-structured interview protocol.

A. Procedure

At the beginning, participants received a short presentation introducing the Monero transaction structure, in particular, the UTXO model and key differences between Bitcoin and Monero. We then explained Monero's privacy mechanisms, including ring signatures, stealth addresses, and RingCT.

Next, participants were introduced to the MoneroVis interface, its visualization components, and interactions. This was followed by a free exploration phase, during which participants interacted with the tool without guidance while verbalizing their thoughts and feedback using a think-aloud protocol.

After the exploration phase, participants were given a guided transaction traceability task based on a WannaCry 2.0 case study. They were provided with a set of initial transaction hashes and asked to trace potential transaction flows using the output-tracing panel, again employing a think-aloud protocol.

After the session, participants were asked to share their impressions and overall feedback on the tool. Finally, they completed a post-study questionnaire collecting both quantitative ratings and open-ended responses to evaluate usability, usefulness, and perceived insights.

B. Participants

We recruited eight participants (seven online, one in-person) with varying levels of familiarity with cryptocurrencies and blockchain analytics. Half of the participants were aged 25–34, two were aged 35–44, and one each fell into the 45–54 and 55–64 age ranges. On average, participants had 5.75 years of domain experience (SD = 3.1, median = 5.5).

All participants reported experience studying or conducting research in the cryptocurrency domain and have monitored transaction activities. Seven participants currently own or have previously owned cryptocurrencies, and six reported having trading experience. Additionally, six participants indicated that blockchain-related work is part of their professional activities.

Regarding tool usage, participants reported regular use of blockchain explorers (e.g., XMRChain, Etherscan, and Blockchain.info): six use such tools daily, one weekly, and one monthly. Six reported analyzing raw blockchain data frequently, one very frequently, and one occasionally.

Regarding technical familiarity, seven participants reported intermediate to expert knowledge of the UTXO model. However, only two participants indicated prior familiarity with Monero-specific concepts such as stealth addresses, ring signatures, decoys, and RingCT.

C. Quantitative Evaluation

We collected verbal feedback during the think-aloud sessions and combined it with results from the post-study questionnaire. Table I presents the questionnaire results on a 5-point Likert scale with the number of responses.

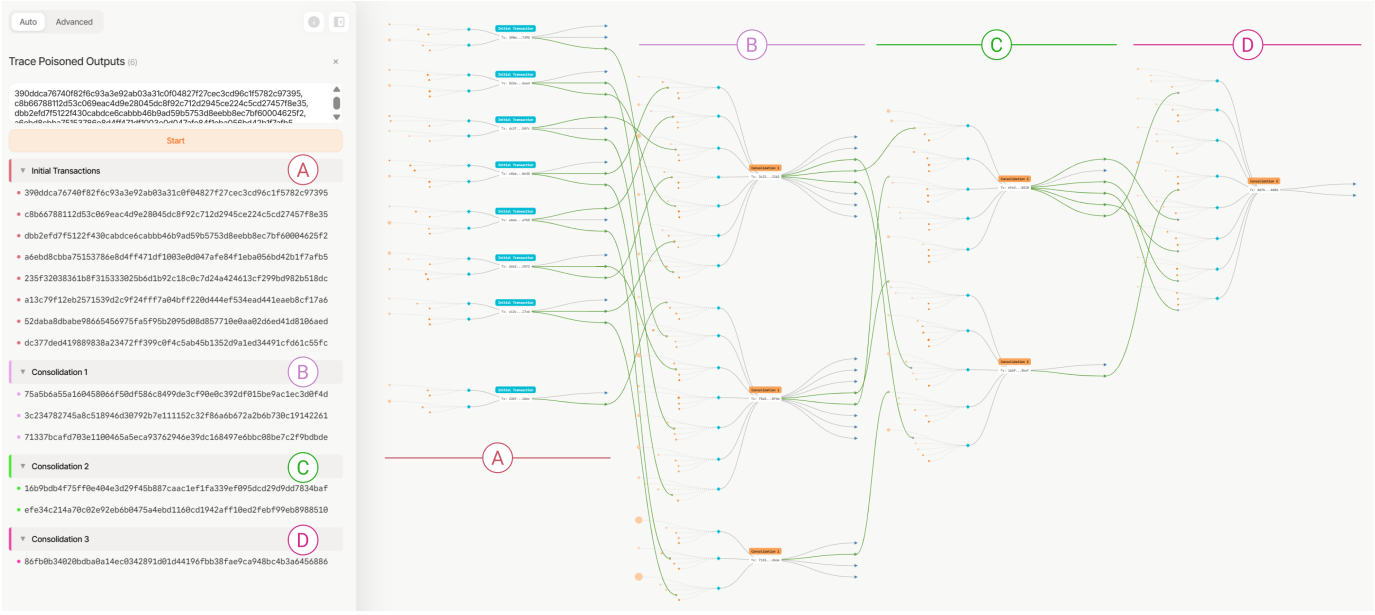

Fig. 6. Output tracing panel in auto mode. Users enter initial transaction hashes in the top-left input field to start the analysis. The left panel shows the input transactions (A) and recursively detected consolidation lists (B–D). The right panel visualizes the multi-hop transaction graph, where starting transactions (A) are followed by successive rounds of consolidation (B–D), with edges indicating inferred fund flows.

Block details view. Participants rated the block treemap highly effective for providing an overview of block activity (1.1) and for helping identify transactions worth further inspection (1.5). The encoding of transaction size and fee via area and color was perceived as intuitive, helping understand fee distributions (1.3). The interactive fee filtering was also highlighted as useful for isolating transaction tiers and detecting outliers (1.4).

Transaction details view. The transaction graph was positively evaluated, with most participants agreeing that it clarifies the relationships between inputs, ring members, and outputs (2.1) and supports tracing transaction flows (2.5). Layout and color encodings were found to be effective for reasoning about ring member age (2.2, 2.4), whereas responses regarding temporal spacing were more neutral (2.3). Overall, participants reported increased confidence in analyzing transaction-level uncertainty compared to traditional blockchain explorers (2.6).

Decoy map. Participants agreed that the decoy map clearly communicates how outputs are reused as decoys across transactions, supporting understanding of the anonymity set (3.1, 3.2). The ability to explore forward connections from an output was appreciated, as it reveals reuse and propagation patterns that are difficult to observe using standard tools (3.3, 3.4).

Output-tracing panel. The output tracing functionality, especially the automated mode, was rated as intuitive and effective for identifying output merging patterns (4.1). Participants found the generated multi-hop tracing graph easy to follow and useful for reconstructing likely transaction flows (4.2). In contrast, the advanced tracing mode received more mixed feedback, suggesting that additional guidance is needed for more complex analytical workflows (4.3).

Overall assessment. Participants agreed that MoneroVis improves their ability to analyze transactions under uncertainty and provides additional value compared to existing blockchain explorers (5.1, 5.3). The integration of multiple views was perceived as effective in supporting the transaction tracing workflow (5.2).

System usability. Using the System Usability Scale (SUS), participants rated the system as easy to use (S3) and well-integrated (S5). They generally disagreed with negative statements, indicating that the system was neither unnecessarily complex (S2) nor cumbersome (S8). Most participants also reported feeling confident using the system (S9) and believed that others could learn it quickly (S7), although they noted that understanding Monero's underlying concepts remains a necessary prerequisite (S10).

D. Qualitative Feedback

Participants described MoneroVis as a comprehensive tool for analyzing privacy-preserving transactions. They agreed that it improves their ability to reason about transaction structures under uncertainty and provides additional value compared to existing blockchain explorers. One participant mentioned that MoneroVis is "an excellent tool, even if, by design, we can get limited insights from Monero transactions" (P4).

Participants highlighted the *decoy map* view and *output-tracing* panel as the most valuable features of MoneroVis, with the heuristic categorization of decoys identified as a key takeaway. The graph-based visualizations were consistently praised for their clarity and expressiveness and were found to be "really helpful to understand the transaction patterns" (P2). Several participants emphasized that the system provides "very clear ways to perform tracing operations and to discover patterns" (P7).

TABLE I
POST-STUDY QUESTIONNAIRE RESULTS ON A 5-POINT LIKERT SCALE.
SA = STRONGLY AGREE, A = AGREE, N = NEUTRAL, D = DISAGREE, SD = STRONGLY DISAGREE, NA = NOT ANSWERED.

	Question	SA	A	N	D	SD	NA
Block Details View							
1.1	The treemap helps me compare transaction fees and transaction sizes more effectively than a standard list-based block explorer.	2	**5**	1			
1.2	The treemap allows me to quickly identify outlier transactions (e.g., unusually high fees or potential spam).	**5**	3				
1.3	The visual encoding (area and color) makes the overall fee distribution within a block easy to understand.	3	**5**				
1.4	The interactive fee filtering (e.g., legend hovering) is useful for isolating specific transaction tiers.	**5**	3				
1.5	The treemap provides a good high-level overview that helps me decide which transactions are worth further inspection.	1	**6**	1			
Transaction Details View							
2.1	The double-sided tree structure helps me understand the relationship between inputs, ring members, and outputs.	**5**	3				
2.2	The layout makes it easier to reason about which ring members are more likely to be real inputs.		**4**	**4**			
2.3	The temporal spacing effectively highlights risks related to the Guess-Newest heuristic.		2	**6**			
2.4	The color coding helps me intuitively assess the relative age of ring members.	2	**5**		1		
2.5	This visualization helps me trace how value flows through a transaction despite the presence of decoys.	1	**4**	3			
2.6	I feel more confident analyzing transaction-level uncertainty using this view than with a standard block explorer.	2	**4**	2			
Decoy Map							
3.1	The decoy map clearly communicates how a single output is reused as a decoy across multiple transactions.	3	**4**	1			
3.2	The visualization helps me understand the concept of an anonymity set in Monero.	**3**	2	**3**			
3.3	I can easily follow forward connections from an output to later transactions.	2	**5**	1			
3.4	This view reveals reuse or propagation patterns that are difficult or impossible to see in standard block explorers.	2	**5**	1			
3.5	The decoy map supports longer-term tracing and hypothesis building about transaction behavior.	**3**	2	**3**			
Output-Tracing Panel							
4.1	Tracing in auto mode is easy to use and the results are understandable.	**4**	2	2			
4.2	The resulting transaction graph in auto mode provides a good visual overview of the output-merging transactions.	**5**	2	1			
4.3	I understand how to use the advanced mode for tracing without additional information or help.		**3**	2	**3**		
Overall Assessment							
5.1	Using MoneroVis improved my ability to analyze Monero transactions under uncertainty.	**4**	**4**				
5.2	The combination of macro, micro, and forward-trace views works well together.	**4**	**4**				
5.3	Compared to existing Monero block explorers, MoneroVis provides additional analytical value.	**5**	2				1
System Usability							
S1	I think that I would like to use this system frequently.	1	**3**	**3**	1		
S2	I found the system unnecessarily complex.				**5**	3	
S3	I thought the system was easy to use.	1	**6**	1			
S4	I think that I would need the support of a technical person to be able to use this system.		1	1	**5**	1	
S5	I found the various functions in this system were well integrated.	3	**5**				
S6	I thought there was too much inconsistency in this system.				**4**	**4**	
S7	I would imagine that most people would learn to use this system very quickly.	**3**	2	1	2		
S8	I found the system very cumbersome to use.			1	2	**5**	
S9	I felt very confident using the system.	1	**6**	1			
S10	I needed to learn a lot of things before I could get going with this system.		2	2	**3**	1	

Participants further commended that the visual encodings effectively abstract Monero's complexity. One participant stated that "the visualization enables a non-expert user to understand the concept more easily" (P6), while another noted that "the combination of all the different visualizations is great, it helps a lot to understand everything" (P8).

In terms of improvements, one participant suggested that the usability of the output-tracing feature "could be polished with a clear how-to-use and process explanation box" (P1), and another emphasized that "a good video tutorial and documentation for non-experts will go a long way" (P8). Additional requests included support for exporting graph data (e.g., SVG, GEXF, CSV) and adding annotation or labeling capabilities (P7). Some participants also suggested extending functionality, including customizable color encodings (P7), preserving color encodings during selection (P3), filtering notable ring members (P1), and dark/light mode (P1).

Overall, the feedback highlights strong perceived value in the system and suggests opportunities to improve the accessibility of traceability features.

VIII. CONCLUSION

MoneroVis demonstrates how visual analytics can support forensic analysis of privacy-preserving blockchains. Because ground-truth spends are generally unavailable, the system does not assert which inputs are the true spends. Instead, it enables analysts to inspect ring structures, interpret heuristic cues, and reconstruct plausible transaction flows under uncertainty.

The WannaCry 2.0 case study illustrates how MoneroVis supports poisoned-output tracing. It should be regarded as a validation of the workflow rather than an independent ground-truth evaluation, as it reproduces a prior investigation. The user study further provides evidence that the tool improves participants' confidence and ability to interpret complex transaction structures, compared with conventional blockchain explorers.

Future major protocol upgrades, such as *Full-Chain Membership Proofs++* (FCMP++), will limit the applicability of the current approach, as FCMP++ transactions cannot be analyzed using ring-based heuristics. Future work will therefore require different transaction analysis and tracing strategies,

potentially based on statistical or probabilistic representations, to support forensic analysis under the new privacy model.

ACKNOWLEDGEMENT

We would like to thank all participants in our user study for their valuable feedback, which helped improve our tool.

REFERENCES

[1] N. van Saberhagen, "Cryptonote v2.0," Tech. Rep., 2013, Accessed: 2026-03-11. [Online]. Available: https://www.getmonero.org/resources/research-lab/pubs/cryptonote-whitepaper.pdf

[2] Koe, K. M. Alonso, and S. Noether, *Zero to Monero: Second Edition*, 04 2020, Accessed: 2026-03-11. [Online]. Available: https://web.getmonero.org/library/Zero-to-Monero-2-0-0.pdf

[3] SerHack, *Mastering Monero: The Future of Private Transactions*. Independently published, 2018, Accessed: 2026-03-11. [Online]. Available: https://masteringmonero.com/

[4] A. Kumar, C. Fischer, S. Tople, and P. Saxena, "A traceability analysis of Monero's blockchain," in *European Symposium on Research in Computer Security*. Springer, 2017, pp. 153–173.

[5] M. Möser, K. Soska, E. Heilman, K. Lee, H. Heffan, S. Srivastava, K. Hogan, J. Hennessey, A. Miller, A. Narayanan *et al.*, "An empirical analysis of traceability in the Monero blockchain," *arXiv preprint arXiv:1704.04299*, 2017.

[6] A. Hinteregger and B. Haslhofer, "Short paper: An empirical analysis of Monero cross-chain traceability," in *International Conference on Financial Cryptography and Data Security*. Springer, 2019, pp. 150–157.

[7] N. Hammad and F. Victor, "Monero traceability heuristics: Wallet application bugs and the Mordinal-P2Pool perspective," in *2024 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 2024, pp. 540–548.

[8] N. Tovanich, N. Heulot, J.-D. Fekete, and P. Isenberg, "Visualization of blockchain data: A systematic review," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 7, pp. 3135–3152, 2019.

[9] Z. Zhong, S. Wei, Y. Xu, Y. Zhao, F. Zhou, F. Luo, and R. Shi, "SilkViser: A visual explorer of blockchain-based cryptocurrency transaction data," in *2020 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 2020, pp. 95–106.

[10] C. Kinkeldey, J.-D. Fekete, T. Blascheck, and P. Isenberg, "BitConduite: Exploratory visual analysis of entity activity on the Bitcoin network," *IEEE Computer Graphics and Applications*, vol. 42, no. 1, pp. 84–94, 2021.

[11] X. Yue, X. Shu, X. Zhu, X. Du, Z. Yu, D. Papadopoulos, and S. Liu, "BitExTract: Interactive visualization for extracting Bitcoin exchange intelligence," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 162–171, 2018.

[12] Y. Sun, H. Xiong, S. M. Yiu, and K. Y. Lam, "BitVis: An interactive visualization system for Bitcoin accounts analysis," in *2019 Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, 2019, pp. 21–25.

[13] BlockSec, "MetaSleuth: Crypto tracking and investigation platform," https://docs.metasleuth.io/, Accessed: 2026-08-02.

[14] Chainalysis, "Chainalysis Reactor: Crypto and blockchain investigations," Accessed: 2026-08-02. [Online]. Available: https://www.chainalysis.com/product/reactor/

[15] Iknaio Cryptoasset Analytics, "Pathfinder: Cryptoasset transaction tracing platform," Accessed: 2026-08-02. [Online]. Available: https://iknaio.com/platform/

[16] G. Di Battista, V. Di Donato, M. Patrignani, M. Pizzonia, V. Roselli, and R. Tamassia, "BitConeView: Visualization of flows in the Bitcoin transaction graph," in *2015 IEEE Symposium on Visualization for Cyber Security (VizSec)*. IEEE, 2015, pp. 1–8.

[17] M. Ahmed, I. Shumailov, and R. Anderson, "Tendrils of crime: Visualizing the diffusion of stolen bitcoins," in *International Workshop on Graphical Models for Security*. Springer, 2018, pp. 1–12.

[18] S. Bistarelli and F. Santini, "Go with the Bitcoin flow, with visual analytics," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, 2017, pp. 1–6.

[19] D. McGinn, D. Birch, D. Akroyd, M. Molina-Solana, Y. Guo, and W. J. Knottenbelt, "Visualizing dynamic Bitcoin transaction patterns," *Big Data*, vol. 4, no. 2, pp. 109–119, 2016.

[20] Bubblemaps SAS, "Bubblemaps: Visual analytics for onchain trading," Accessed: 2026-08-02. [Online]. Available: https://bubblemaps.io/

[21] S. Miksch and W. Aigner, "A matter of time: Applying a data–users–tasks design triangle to visual analytics of time-oriented data," *Computers & Graphics*, vol. 38, pp. 286–290, 2014.

[22] C. G. Akcora, Y. R. Gel, and M. Kantarcioglu, "Blockchain networks: Data structures of Bitcoin, Monero, Zcash, Ethereum, Ripple, and Iota," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 12, no. 1, p. e1436, 2022.

[23] N. A. Bax, "Tracing the WannaCry 2.0 Monero Transactions," 2021, Accessed: 2026-03-11. [Online]. Available: https://medium.com/@nbax/tracing-the-wannacry-2-0-monero-transactions-d8c1e5129dc1

[24] Monero Community Workgroup, "Breaking Monero Episode 09: Poisoned Outputs (EAE Attack)," 02 2019, Accessed: 2026-03-11. [Online]. Available: https://www.youtube.com/watch?v=iABIcsDJKyM