BitConeView: Visualization of Flows in the Bitcoin Transaction Graph

Giuseppe Di Battista, Valentino Di Donato, *Student Member, IEEE*, Maurizio Patrignani, Maurizio Pizzonia, Vincenzo Roselli, and Roberto Tamassia, *Fellow, IEEE*

Abstract—Bitcoin is a digital currency whose transactions are stored into a public ledger, called blockchain, that can be viewed as a directed graph with more than 70 million nodes, where each node represents a transaction and each edge represents Bitcoins flowing from one transaction to another one. We describe a system for the visual analysis of how and when a flow of Bitcoins mixes with other flows in the transaction graph. Such a system relies on high-level metaphors for the representation of the graph and the size and characteristics of transactions, allowing for high level analysis of big portions of it.

Index Terms—Bitcoin, cryptocurrency, money flow, graph visualization, visual analysis, fraud detection

1 Introduction

Recent years have witnessed the growth of Bitcoin [14], a novel type of digital currency. Bitcoin is a virtual currency, stored and exchanged only in digital form. In a radical departure from previous proposals of e-cash, Bitcoin does not rely on a trusted entity like a bank or government authority. Instead, it is based on an open social model of trust and on incentivized collaboration.

After an initial period when Bitcoin was known only to a small group of enthusiasts, Bitcoin has gained considerable popularity as it can be used anywhere there is an Internet connection. A large number of merchants today accept payments in Bitcoins and several exchanges allow trading Bitcoins against traditional currencies. The current value of all Bitcoins in circulation is over US $3B.

Bitcoin is at the same time transparent and private. Transparency is enabled by open standards and by recording all transactions in a public ledger. The integrity of transactions is in turn backed by widely accepted cryptographic methods. Privacy is achieved by associating owners of Bitcoins with opaque cryptographic identifiers that conceal the actual identities behind them.

The anonymity granted by the Bitcoin protocol provides compelling benefits in a privacy-conscious society that is wary of government surveillance. At the same time, it can be used as an alternative to cash for the purchase of illegal goods and services.

The Bitcoin public ledger, called *blockchain* is a large directed acyclic graph storing all Bitcoin transactions ever executed. In this DAG, each node is a transaction that serves as a container of Bitcoins. Edges represent flows of Bitcoins between transactions. Namely, an edge from transaction t to transaction t' is labeled with the amount of Bitcoins transferred from t to t'.

The blockchain provides a wealth of information about the use of Bitcoin. It can be used to track and analyze the flow of bitcoins over time, discovering patterns of transactions of interest, such as the presence of chains, trees, or high-degree nodes. For example, the blockchain can be used to deanonymize transactions flowing from a "tainted" transaction, where the identity of the Bitcoin owner or existence of an illegal purpose is leaked. For this reason, online services have been created that mix Bitcoins from tainted transactions with "clean" Bitcoins. The transactions executed by these services can often be pinpointed in the blockchain itself as they usually correspond to subgraphs with a large number of outputs, all reachable from a large number of inputs.

We present *BitConeView*, a tool for the visual analysis of Bitcoin flows. BitConeView allows to graphically track how Bitcoins from given sources (transaction inputs) are spent over time by means of transactions and are eventually stored at multiple sinks (unspent transaction outputs). BitConeView is the first graphical tool for the analysis of flows in the blockchain. It can be used to reveal Bitcoin flow patterns of interest to the analyst in many application domains, such as accumulation, distribution, and mixing.

More in detail, the requirements of BitConeView are the following. Given an amount M of Bitcoins specified by identifying a starting transaction (or a set of starting transactions) the user should be able to: (R1) follow M and its usage over time, understanding when M is mixed up with other Bitcoins and what parts of M are unspent at a certain time; (R2) obtain succinct information about the "degree of mixing" of M with other Bitcoins and how this varies over time; (R3) inspect possible Bitcoin transfers between entities through the transaction graph; and (R4) support investigations for discovering potential anomalies in the mixing pattern of M.

This paper is organized as follows. Basic concepts about Bitcoin transactions and their associated graph are introduced in Section 2. In Section 3, we present the visualization principles, user interface, use cases and system architecture of BitConeView. Section 4 overviews a user study we have conducted to evaluate BitConeView. Related work is discussed in Section 5. Section 6 describes future research directions. Finally, the appendix shows a full-page snapshot of BitConeView.

2 Background on the Bitcoin Currency

In this section we give a simplified description of the Bitcoin protocol aimed at defining the Bitcoin transaction graph. For a broader introduction to Bitcoin, see, e.g., the original paper describing Bitcoin [14] and a recent survey [7].

Bitcoin (in what follows *BTC*) is a currency stored and exchanged only in digital form, defined by open standards and maintained by a peer-to-peer network with open membership, called the *Bitcoin network*. BTCs are transferred between parties by means of transactions. All transactions are recorded in a public ledger, called *blockchain*, a copy of which is kept at each network node. Each new transaction is propagated through the network, validated by each network node, and eventually added to the blockchain.

A *transaction* (in what follows *tx*) t has a set of inputs $i_t^1, \ldots, i_t^h$ and a set of outputs $o_t^1, \ldots, o_t^k$, each associated with a cryptographic identifier, called *address*, and a BTC amount. We denote such amounts with $A(i_t^1), \ldots, A(i_t^h)$ and $A(o_t^1), \ldots, A(o_t^k)$, respectively. Tx t transfers BTCs from its inputs to its outputs. The sum of the input amounts is greater than or equal to the sum of the output amounts. In case of inequality, the difference, called *transaction fee*, is implicitly transferred to the maintainers of the bitcoin network. Since transaction

- *Giuseppe Di Battista, Valentino Di Donato, Maurizio Patrignani, Maurizio Pizzonia, and Vincenzo Roselli are with Roma Tre University. Italy. E-mail: {gdb, didonato, patrigna, pizzonia, roselli}@dia.uniroma3.it.*
- *Roberto Tamassia is with Brown University. U.S.A. E-mail: rt@cs.brown.edu.*

978-1-4673-7599-3/15/$31.00 ©2015 IEEE

fees are usually very small, in the rest of the paper we assume that $\sum_{j=1}^{h} A(i_t^j) = \sum_{j=1}^{k} A(o_t^j) = A(t)$.

Outputs of txs are denoted *txo*s. At a certain time T, each txo of a tx t can be unspent (*utxo*) or spent (*stxo*). The only way to spend a utxo o_t of t is to use it as the input $i_{t'}$ of a tx t' (with $t \neq t'$). The amounts $A(i_{t'})$ and $A(o_t)$ should be equal. When a utxo is spent, it becomes an stxo. The set of the utxos of all the txs at time T can be viewed as the set of the BTCs "circulating" at time T.

We define a directed graph, called *Bitcoin transaction graph* (*tx-graph*), at a given time as follows. Nodes are txs. Each node is decorated with its inputs and outputs. Nodes t and t' are connected by a directed edge (t,t') if one output o_t of t is used as an input $i_{t'}$ of t'. We denote by $A(t,t')$ the amount of BTCs of o_t and $i_{t'}$. More precisely, the tx-graph is a multigraph, since several outputs of t can correspond to inputs of t'. For the sake of simplicity, we will refer to such a graph without the prefix "multi".

The blockchain is divided into "pages" called *blocks*. Each block contains, roughly, the txs issued in a time interval of ten minutes. The block sequence number is its *block height*. We refer to a block using its height.

Even if all txs are public, the identity of the parties that used or can use the outputs of the txs (stxos or utxos) are not. Indeed, an address is the cryptographic hash of a public key and the owner of the corresponding private key remains anonymous. Also, to further preserve anonymity, users typically generate one or more new addresses for each new tx.

As of June 2015, the blockchain consists of about 360,000 blocks and contains 83 million txs, that is the number of nodes of the tx-graph. On average, each tx has two inputs but the graph contains dense subgraphs. Roughly, 100,000 txs are added to the blockchain per day.

3 A System for the Visual Analysis of the Tx-Graph

In this section, we first describe BitConeView's core concepts and analytical tools. Second, we describe its user interface, also with the help of some use cases. Finally, we provide technical details of its architecture.

3.1 BitConeView's Core Concepts

In the following, we outline the main concepts that support the visual analysis offered by BitConeView. Leveraging on these concepts, BitConeView provides a succinct representation of the complex structure of the tx-graph in a few suggestive graphic features and parameters. Note that specific portions of the tx-graph are visualized on demand [18].

BitCone. The *BitCone* is the cornerstone of the metaphor adopted by BitConeView. It conveys how the BTCs of a specific transaction s, called *source tx*, mix with other BTCs over time. The apex of the BitCone is s and the BitCone enlarges when new BTCs mix with the current ones in the BitCone. From the point of view of the tx-graph, the BitCone of s is the subgraph reachable from s within a certain time interval. We restrict to a given time interval because it would be unwieldy to look at the entire history of the blockchain, which currently spans more than 5 years.

Time. Time is discrete. To refer to the time of a tx, BitConeView uses the height of the block containing the tx. Indeed, the timestamp of a block is not considered reliable [6]. Hence, if a tx is in a block of height b, we say that the tx took place at "time b". A BitCone is characterized by a source tx, s, and a time, T, which indicates the end of the considered time interval. We say that a block belongs to a BitCone if it contains a tx in the BitCone.

Utxos. A BitCone identifies those outputs of txs of the BitCone that were not spent at time T. Intuitively, these can be considered the outputs of a BTC mixing process. From the point of view of the tx-graph, those utxos are the "frontier" of the BitCone, corresponding to edges that either (1) belong to the tx-graph but are not included in the BitCone because their destinations are txs that take place after time T; or (2) do not belong to the tx-graph but could be added to it in the future.

Inputs. The BitCone also identifies the inputs of txs that insert new BTCs into the BitCone. These inputs are responsible for the mixing process. From the point of view of the tx-graph, these inputs correspond to edges entering the BitCone from outside.

3.2 BitConeView's Analytical Tools

We have selected our core concepts in such a way to enable several analytical tools for the study of BTC flows in the tx-graph. The most important are:

Budget Analysis. Given a BitCone, we define the *budget* of a block b, denoted with $A(s,b)$, of the BitCone rooted at s, as the total amount of BTCs that enters the BitCone until time b. We have: (*i*) For the block containing the source s, the budget is just the sum of the inputs of s. (*ii*) For any other block b, $A(s,b)$ is the sum of $A(s,b-1)$ and the amounts of the inputs of the txs of b belonging to the BitCone that come from outside the BitCone. These inputs, called *intruders*, correspond to outputs of txs not belonging to the BitCone, that is, BTCs that mix into the BitCone at time b.

Utxos Analysis. For each block b of the BitCone, we define $utxos(b)$ as the sum of the amounts of the utxos belonging to the BitCone having their txs in b. These correspond to the outputs of txs at time b that remain unspent until at least time T.

Purity Analysis. In order to visualize how much the BTCs of the source tx s have been "contaminated" with other BTCs, we enrich the BitCone with a diagram showing how the *purity* of the BTCs from s evolves over time, for some definition of purity. Our definition is given in Section 3.4. However, the system can be customized to display other types of purity.

Transfer Analysis. Given the source tx s and a utxo u of a BitCone, one could ask what is the maximum amount of the BTCs coming from s that could be transferred to u. This can be computed considering the tx-graph of the BitCone as a flow network with source s and sink u, and where for each edge e, the capacity of e is $A(e)$.

3.3 User Interface

The interface of BitConeView displays a BitCone with source s and time T as illustrated in Figure 1. The timeline flows from top to bottom. Each block of the BitCone (i.e., containing a tx that belongs to the BitCone) is represented by a rectangular region filled in light gray whose vertical position corresponds to the block height. Such a rectangular region provides a compact representation of the subgraph of interest by aggregating all transactions that fall into the same block. Its width and placement display the sources of BTCs entering the BitCone from the outside and the amounts of BTCs that remain unspent until time T. Namely, the x-coordinate of the right boundary of the rectangle representing block b is proportional to $A(s,b)$. Also, the x-coordinate of the left boundary is proportional to the sum of $utxos(b')$ for all blocks $b' < b$ in the BitCone. Within each block, the upper part is filled either with dark gray rectangles representing intruder inputs or with colored rectangles representing inputs of source txs. The lower part is filled with rectangles representing *utxos* at time T. They are filled dark gray if they are spent after T and are filled black if they are never spent.

So far we have considered a BitCone with a single source tx. However, the user might be interested in simultaneously looking at the flows of BTCs originating from multiple source txs. Hence, we generalize the concept of BitCone allowing to have several source txs. Namely, suppose to have source txs $s_1, \ldots, s_m$, and for simplicity, assume they are in increasing time order. The inputs of tx s_1 are the apex of the BitCone. Inputs of txs from s_2 to s_m are placed on the right, in the same position as if they were intruders. Each of $s_1, \ldots, s_m$ has a different color. See Figure 2. Observe that the purity increases when the second source enters the BitCone. In fact, the purity refers to the BTCs of all the sources and the entrance in the BitCone of "fresh" pure money increases the purity.

The design choices described above are justified by the following simple considerations. Instead of using an explicit node-link representations, the BitCone metaphor: (i) allows the user to better perceive the amount of flow of Bitcoins entering (intruders) and exiting (utxos) the cone; (ii) provides a clear representation of the temporal collocation of intruders and utxos; and (iii) aggregates transactions belonging

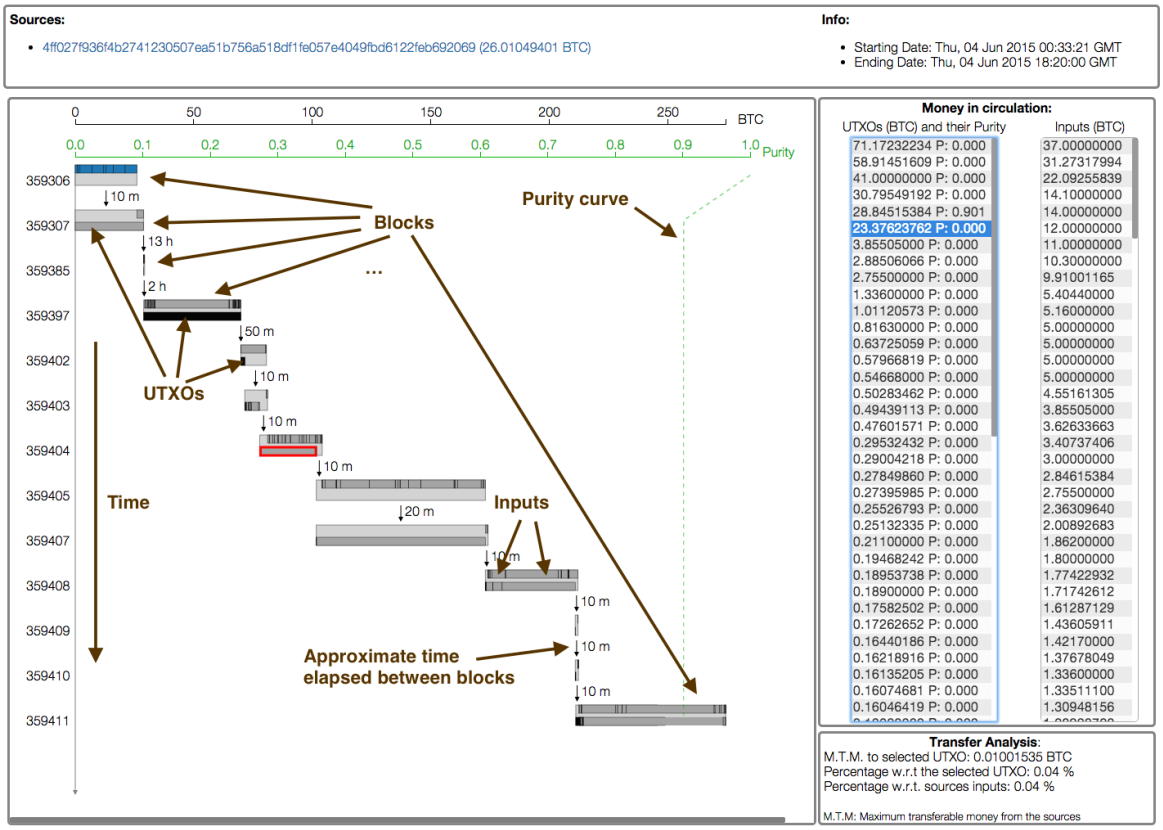

Figure 1. A BitCone displayed by BitConeView. The hash of the source tx is `4ff027f936f4b2741230507ea51b756a518df1fe057e4049fbd6122feb692069`. We explicitly report the entire hash so that the reader can easily look up the transaction using blockchain exploration services such as Blockchain.info. The source tx has about 26 BTCs in input. The selected time interval consists of 106 blocks (including the block of the source), corresponding to about 18 hours. The associated graph consists of 62 nodes and 77 edges. All these transactions are contained in the 13 blocks that are shown. In block 359307, issued about 10 minutes after the first block, one small intruder input arrives, injecting into the BitCone about 2.84 BTCs. Other intruder inputs enter the BitCone in the following hours, injecting into the BitCone a total of about 250 BTCs. The trend of the purity is shown by the green dashed curve. The upper right box shows details about utxos and inputs. The lower right box shows the result of a transfer analysis involving the source tx and the utxo highlighted in red. It displays that the maximum transferable flow between the two is 0.01 BTCs, that is 0.04% of the capacity of the utxo and 0.04% of the capacity of the source input. Additional information about a utxo are displayed on mouse hovering.

to the same block to unique visual components and therefore reduces the potential cluttering generated by standard node-link diagrams.

3.4 The Concept of Purity: Purpose and Definition

In this section, we motivate and formally define *purity*. This concept is inspired by "taint analysis", a commonly used method in the Bitcoin community to determine the provenance of the BTCs of a tx from another one.

Once BTCs enter a tx, it is undefined how they are distributed among the outputs. As an example, consider Figure 3(a) where tx t with inputs i_t^1 and i_t^2 and outputs o_t^1 and o_t^2, where $A(i_t^1) = 100$, $A(i_t^2) = 400$, $A(o_t^1) = 200$, and $A(o_t^2) = 300$. It is impossible to determine whether the 100 BTCs of i_t^1 (1) went all into o_t^1, (2) went all into o_t^2, or (3) were distributed among o_t^1 and o_t^2. Hence, we make a uniform distribution assumption and allocate 40 BTCs to $A(o_t^1)$ and 60 BTCs to $A(o_t^2)$, proportionally to values of $A(o_t^1)$ and $A(o_t^2)$. This assumption is implicit in the taint analysis provided by Blockchain.info [1].

More formally, each txo o_t of tx t of the tx-graph is labeled with the expected amount $E(o_t)$ of BTCs reaching it from the source tx. For each txo o_s of the source tx, s, we set $E(o_s) = A(o_s)$. Consider a tx v with inputs $u_1 \dots u_h$. Let $o_{u_1}, \dots, o_{u_h}$ be their corresponding outputs, and let $o_v^1, \dots, o_v^k$ be the outputs of v. We define

$$E(o_v^j) = \sum_{i=1}^{h} E(o_{u_i}) \frac{A(o_v^j)}{A(v)}$$

Let $u_1, \dots, u_n$ be the utxos of the BitCone with source s and time T. Define $E(s,T) = \sum_{i=1}^{n} E(u_i)$. We observe that $E(s,T)$ is independent from T and we have $E(s,T) = A(s)$. Let $A(s,T) = \sum_{i=1}^{n} A(u_i)$. This value, informally defined in Section 3.1, characterizes the overall amount of BTCs that enter the BitCone until T.

A possible definition of purity at time T could be the ratio $\frac{A(s)}{A(s,T)}$. This is the ratio of the amount of BTCs entering the BitCone from s to the overall amount of BTCs entering the BitCone. Observe that this ratio is made visually evident from the BitConeView metaphor. However, the above definition can be misleading. Consider the flow of 100 BTCs entering from s (see Figure 3(b)). Suppose that at time T it splits into two subflows of 99 and 1 BTCs, respectively. Suppose that the first subflow mixes with 1 BTC coming from an intruder and that the second subflow mixes with 99 BTC coming from another intruder. The above definition of purity would yield a value of 0.5. On the other hand, it is clear that: (1) the BTCs of the first subflow are still very pure and, (2) the first subflow is much more representative of the original flow than the second one. Hence, we give a weighted definition of purity that takes into account the expected amount of BTCs in a flow.

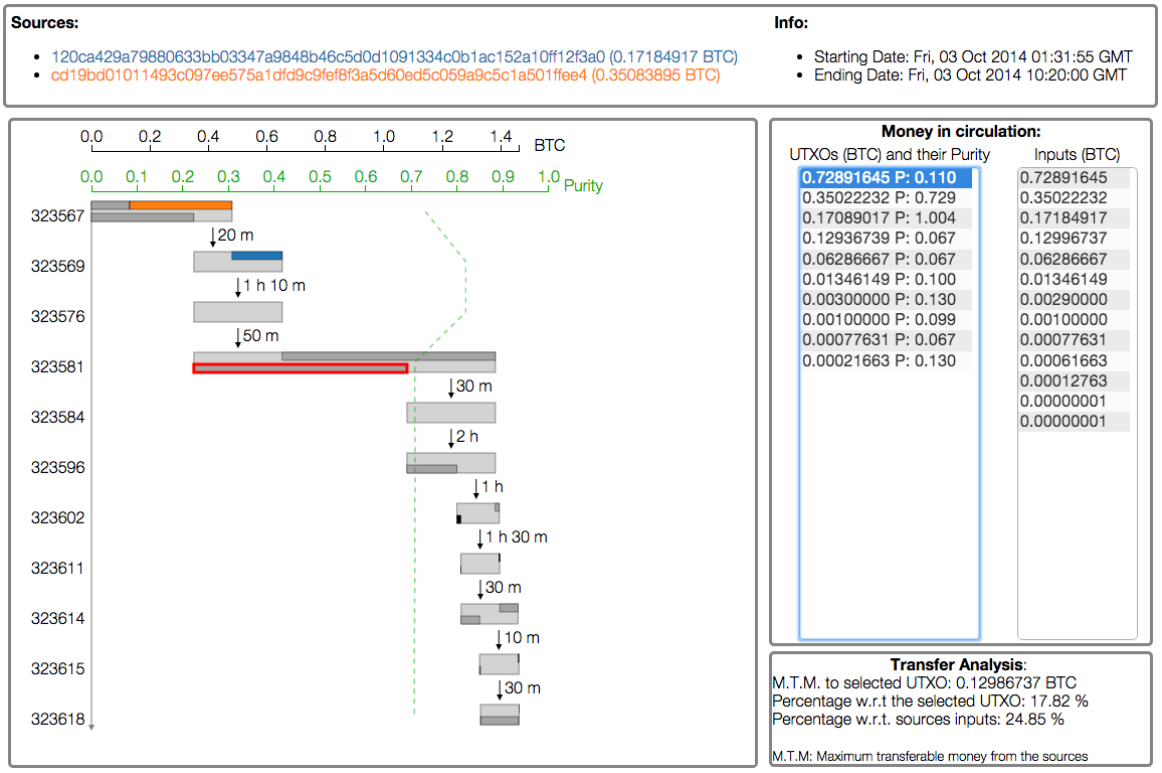

Figure 2. A BitCone with two source txs. Their inputs are filled in orange and blue, respectively.

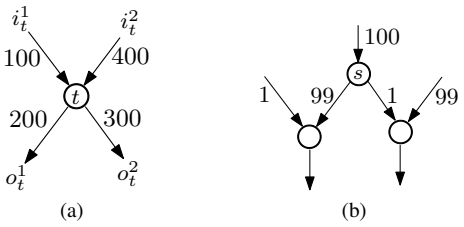

Figure 3. Examples for the definitions of purity given in Section 3.4

Namely, we define the *purity* $P(s,T)$ of the BTCs of the source tx s at time T as follows, where $u_1,\ldots,u_n$ are the utxos at time T of the BitCone of s:

$$P(s,T) = \sum_{i=1}^{n} \frac{E(u_i)}{A(u_i)} \cdot \frac{E(u_i)}{E(s,T)} = \sum_{i=1}^{n} \frac{E(u_i)^2}{A(u_i)A(s)}$$

3.5 Use Cases

We present two use cases. In UC1, a toy-use case, we illustrate how a user can interact with BitConeView to achieve the goal of deepening his/her understanding on a flow of BTCs. In UC2, we analyze with BitConeView several experiments presented in [13] that deal with money laundering. In the following, we sometimes refer to Bitcoins as money.

UC1. Alice is interested in investigating the flow of money originated by a certain tx s_1. In the Bitcoin world each tx is known through its hash value, and there are many services allowing to select and explore txs (see, e.g., [1]). Alice would like to understand what happened to that money during a certain time interval T, say the week after s_1 was issued. Hence, Alice specifies to BitConeView tx s_1, using its hash, and time T (one week later).

Looking at the BitCone provided by BitConeView, Alice immediately observes that the money has been used as input for other txs. She also studies the temporal distribution of such a usage. Looking also at the width of the BitCone, she notices that there is a time t_1 when a substantial amount of money enters the BitCone and mixes with the money from s_1, i.e., she identifies a significant intruder (see the above terminology).

Alice wonders whether she spotted a money laundering activity. To validate this hypothesis, she looks at the purity curve superimposed on the BitCone and observes that at time t_1, the purity has a negligible decrease. She concludes that only a small part of the flow coming from s_1 mixes with the new incoming money.

The fact of not finding what she looked for does not discourage Alice, who has strong suspicions on the flow from s_1. Hence, she extends T to two weeks. After the extension, she finds a time t_2 when more money enters the BitCone. Also, at time t_2 the purity drops dramatically because of the flow entering from intruder s_2.

Using a different source of information, Alice knows that there is a third tx s_3 that mixes money with s_1. Hence, she asks BitConeView to display the BitCone defined by s_1, s_2, and s_3. In this case, she finds that the purity remains high over time. Alice concludes that the money of s_1, s_2, and s_3 has been accurately mixed.

Alice observes that there are some utxos that collect most of the money of the BitCone. She suspects that the money from s_1 is targeted to one of them, called u. Unfortunately, this is impossible to prove, since the money has been mixed. At least, Alice would like to know if this is possible. Hence, she invokes the transfer analysis tool. The result confirms that it is indeed possible that money flows from s_1 to u. Hence, Alice has found what she was looking for.

UC2. In this use case, the goal of Bob is to check the experiments performed in [13] to assess the effectiveness of the available money laundering services. In [13] the authors inject BTCs into two popular services: *BitLaundry* and *BitcoinFog*. The authors also inject BTCs into a shared wallet provided by *Blockchain.info Send Shared*. After injecting the money, they withdraw it and study the txs involved in the

process trying to verify whether the source txs could be traced back starting from the withdrawal ones.

Bob starts examining the BitLaundry experiments looking at the three txs associated with the injection of BTCs into the service. He selects the three tx hashes and for each of them, launches BitConeView and visualizes the three corresponding BitCones (see Figure 4).

Figure 4(a) shows that the injected money is mixed, after about ten hours, with a large quantity of BTCs coming from other txs. This results in a sudden drop of the purity (from 1 to 0.05). Figure 4(b) shows a case in which BitLaundry is less effective, as also reported in [13]. In fact, the purity curve remains quite stable, which means that it is easy to track back inputs from outputs.

Bob then focuses on the experiments conducted on BitcoinFog. While navigating the tx-graph, the authors revealed an interesting pattern: The BTCs used by BitcoinFog as payout often come from txs that are part of long chains in which each tx distributes and/or collects small amounts of BTCs. At the apex of such chains they often found very large txs in which a certain number of outputs are bundled into one. Therefore, Bob selects the hashes of these txs and feeds them to BitConeView. Figure 5 shows two of these chains. Notice that the starting txs of Figures 5(a) and 5(b) aggregate five big inputs (with a total of about 6,000 BTCs) and 279 big inputs (with a total of about 50,000 BTCs), respectively. All these inputs pass almost untouched through both chains.

As a last experiment, Bob analyzes the evolution of the purity for a BitCone whose source tx injects BTCs into Blockchain.info Send Shared. Figure 6 shows the resulting BitCone. In this experiment, the authors inject into the shared wallet 0.40 BTCs and shortly after they withdraw such a money. The two txs are both included in Block 238219. However, looking at the purity curve extended for one more day, it is clear that the injected BTCs mix with other BTCs of the shared wallet thus making this system an effective tool to improve anonymity.

3.6 System Architecture and Prototype

BitConeView visualizes data extracted from the tx-graph, which is implicitly encoded in the Blockchain available to any Bitcoin node. In order to validate transactions and blocks, Bitcoin nodes only need to traverse the tx-graph backward in time. Instead, BitConeView traverses the tx-graph forward in time and requires operations on the tx-graph that are not supported by the standard Bitcoin node software. In our experiments, we used both *Insight-API* [4] and the REST API offered by *Chain.com* [3]. They both allow to traverse the tx-graph forward in time, but provide different trade-offs between efficiency and reliability. In the following, we call *graph server* this component.

BitConeView is a Web-based application: Its logic comprises the exploration of the tx-graph, whose output is the dataset that is shown to the user. The computation of the drawing starts from this dataset. The application server, implemented in Python and using the Flask framework, is in charge of querying the graph server to obtain the needed portion of the tx-graph, and of running a maximum flow algorithm to implement the Transfer Analysis feature (see Section 3.1). Finally, the rendering of the BitCone is performed by the Web browser, which is also in charge of the user interaction. This component is based on the graphic library D3.js [8].

We have developed a prototype of BitConeView, and we have made it publicly available.[1]

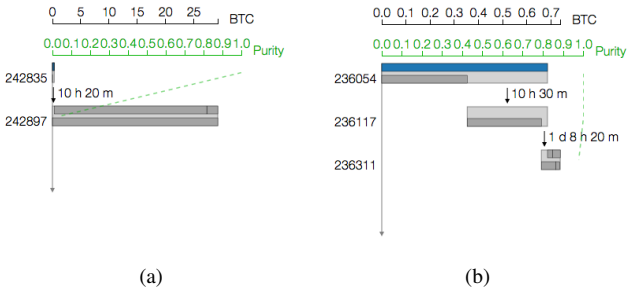

Figure 4. Two BitCones related to money laundering activity of the BitLaundry service.

Figure 5. Chains of transactions generated by money laundering services.

4 Evaluation

We evaluated the BitConeView tool by means of an informal usability study in which we collected feedback from a set of users. In accordance with some of the principles reported in [19], our purpose was to evaluate BitConeView with respect to the requirements specified in Section 1. In particular, we asked the following questions: Are the proposed metaphors effective and intuitive for conveying information about the usage of the BTCs coming from the source tx (Req. R1)? Can the analysis tools (Budget, Utxo, Purity, and transfer analysis) be useful for further investigating the flows of BTCs and their degree of mixing in the tx-graph (Req. R2 and R3)? Does BitConeView help discovering potential money laundering patterns (Req. R4)?

We gathered a small set of users (nine people, all males). Six were engineers in the 30–35 age range and three were detectives of an Italian Investigation Division in the 40–50 age range. In this user group, two were Bitcoin experts, two had some background, while the others barely heard of Bitcoin before. Hence, we first gave the users a 30 minute tutorial on Bitcoin. Second, we extensively demonstrated BitConeView on a collection of source txs, answering questions posed by the users. Finally, we allowed the users to play themselves with the interface, interacting with BitConeView's facilities and exploring real-world data.

At the end of the training session, users were asked to fill out forms with six questions. For each question, users were asked to give a score from one to five (1 for "no, absolutely" and 5 for "yes, absolutely") and write a justification for the score.

The questions are shown in Table 1 together with their average scores. Questions Q1, Q2, and Q3 were used to assess the effectiveness and intuitiveness of the metaphors. Comments for the lower scores were as follows: metaphors are not intuitive since they need to

[1] http://www.bitconeview.info

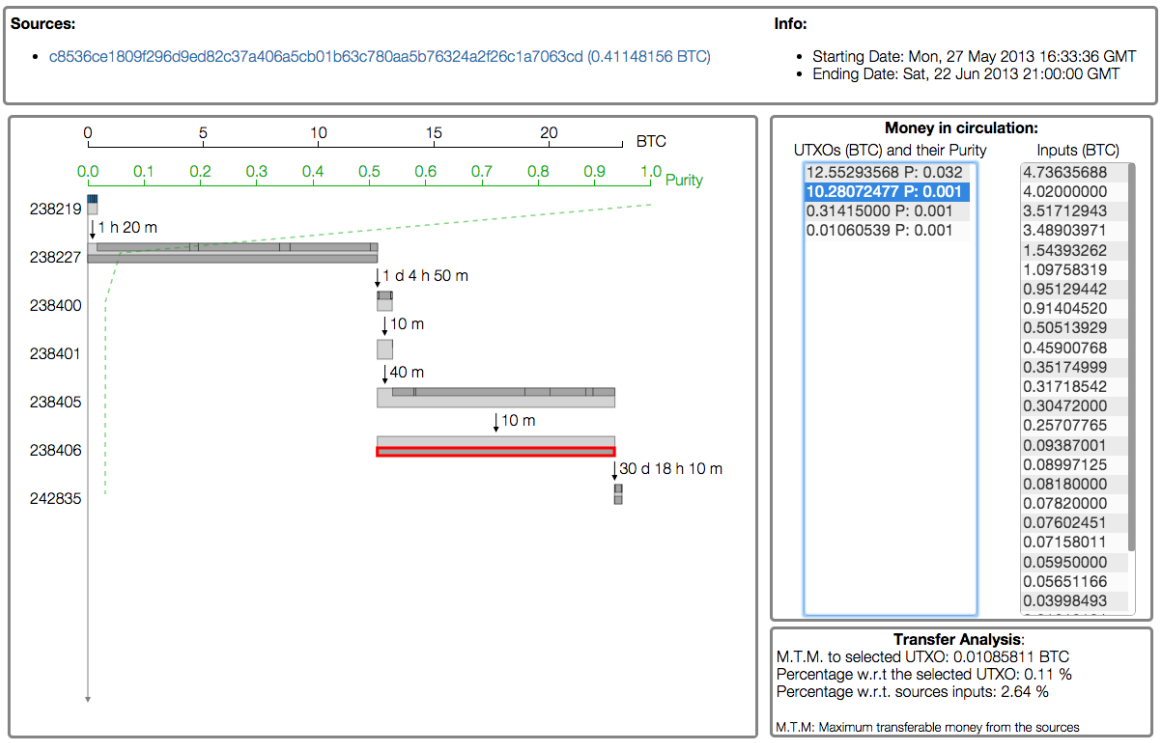

Figure 6. A Bitcone representing an experiment performed on Blockchain.info Send Shared.

Table 1. Questions asked in the usability study, corresponding requirements, and average scores.

#	Req.	Question	Avg.
Q1	R1, R2	Does BitConeView help understanding how much the BTCs of the source txs have been involved in txs with big amounts of BTCs within time T?	4.22
Q2	R1, R2	Does the tool help understanding whether the usage of the BTCs of the source txs has been uniform thorough time or if it has not been used for long periods of time?	3.67
Q3	R1	Does the tool help understanding whether there are txos that are not spent during T?	4.00
Q4	R2	Does the Purity Analysis help understanding how much the BTCs of the source txs has been mixed?	3.67
Q5	R3	Does the transfer analysis help understanding how much a txo may be receiving the BTCs of a source txo?	3.44
Q6	R4	Would BitConeView help discovering money laundering transactions?	3.78

be explained (Q1); uniform vertical spaces between blocks do not help in understanding the timings (Q2); it is difficult to fully reconstruct BTCs movements (Q1, Q2, Q3). With respect to Q4, one user commented that the definition of purity is involuted; another one lamented that overlapping the purity curve with blocks was cluttering; and a third user observed that it is a rough measure. Regarding question Q5, a user observed that transfer analysis may yield false positives, while three users commented that the concept of maximum flow is not intuitive. Finally, question Q6 had significant variance of scores: all 4s and 5s with the exception of one 1 and one 2. The users who gave 1 and 2 commented that they lacked enough information to identify money laundering or that the task was too hard.

Detectives had average scores similar to those of engineers on all questions, with the exception of Q5, where they were more confident about the benefits of transfer analysis (average score 4.33 vs. 3.00).

Overall, we believe that the user study shows that the metaphors of BitConeView are promising and that the exploration of the tx-graph carried out with BitConeView can be effective for discovering patterns of interest. However, there is still room for improvement, especially in the direction of providing graphical encodings for purity and transfers.

5 Related Work

Even if the tx-graph is an ideal playground for network visualization, only a few systems exist that aim at analyzing Bitcoin transactions using visualization tools.

CoinAnalytics JARVIS [10] is conceived for performing in-depth investigations across the blockchain. Its user interface is devoted to analyzing relationships and to explore the tx-graph, which is enriched with addresses (possibly clustered) and other entities. However, it does not provide any feature to help an analyst investigate Bitcoin mixing processes.

CoinViz [5] is a financial visualizer for Bitcoin txs. It contains a tool, based on a force-directed approach, that shows in real-time txs entering the blockchain. For each tx, the sender and recipient addresses are shown. However, no Bitcoin flow analysis is provided and this simple tool seems to have mostly educational purposes. Similar features are exhibited by **bitcoin-tx-graph-visualizer** [11].

Other tools, like **Blockchain.info** [1] and **blockr.io** [2] allow the user to navigate the tx-graph and present several charts on bitcoin financial data. The taint analysis tools provided by **Blockchain.info** seems to be related to the concept of purity. However, no information

is provided on its semantics, it is presented in a tabular form, and it does not allow to analyze multiple sources of taint.

Several papers (see, e.g., [12, 16, 17]) analyze the tx-graph and include drawings of subgraphs of interest, often highlighting flows, that are either laboriously created by hand or generated with standard force directed graph drawing tools that yield cluttered layouts unsuitable for visual analysis.

Finally, looking at the financial fraud detection literature, one work of interest is [9]. It describes a visual analytics system that helps discovering suspicious (traditional) bank wire transactions by giving the analyst multiple coordinated visualizations.

6 Conclusions and Ongoing Work

We presented BitConeView, a system for the visual analysis of Bitcoin flows in the blockchain. The tool encompasses the notion of *purity* that allows analysts to gain an immediate understanding of when and how Bitcoins are mixing in a suspicious way. We analyzed several experiments on real money laundering systems, and showed the effectiveness of BitConeView in detecting mixing processes and patterns. We evaluated the system by means of a usability study in which we collected feedback from nine people, two of whom are detectives investigating financial crimes.

We are extending BitConeView in many directions, as follows. First, we are addressing the scalability of the visualization when the BitCone contains more blocks than can comfortably fit on the screen. This issue is common among systems that visualize a large number of data items (see, for example, [15]). To tackle this problem, we are developing a clustering mechanism that groups together consecutive blocks, as illustrated in Figure 7.

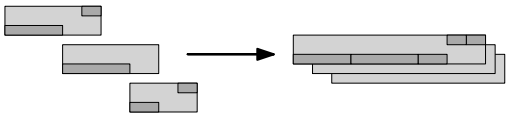

Figure 7. Clustering blocks for vertical scalability

Next, we are adding animation features so that when the specification of the BitCone is changed by the user, the old representation is morphed into the new one in order to preserve the user's mental map. In addition, we are providing a drill-down feature to enable the user to explore the subgraph of the BitCone within a selected block. This subgraph is visualized using a Sugiyama style algorithm [20], which is suitable for directed acyclic graphs. Furthermore, we would like to support the analysis of the blockchains of different types of cryptocurrencies. Finally, we are interested in integrating BitConeView with existing blockchain exploration platforms.

Acknowledgments

This research is supported in part by: (i) the Italian Ministry of Education, University, and Research (MIUR) under PRIN 2012C4E3KT national research project "AMANDA – Algorithmics for MAssive and Networked DAta", (ii) the EU FP7 project "Preemptive - Preventive Methodologies and Tools to Protect Utilities,"grant no. 607093, and (iii) grant CNS–1228485 from the U.S. National Science Foundation.

References

[1] Blockchain.info. `https://blockchain.info/`.
[2] Blockr.io. `http://blockr.io`.
[3] Chain.com. `https://chain.com/`.
[4] Insight API. `https://github.com/bitpay/insight-api`.
[5] A. Aghaseyedjavadi, B. Bloomer, and S. Giudici. Coin viz. `http://people.ischool.berkeley.edu/~shaun/infoviz/bitcoin/index.html`.
[6] A. M. Antonopoulos. *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*. O'Reilly Media, Inc., 1st edition, 2014.
[7] J. Bonneauand, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten. SoK: Research perspectives and challenges for Bitcoin and cryptocurrencies. In *Proc. IEEE Symp. on Security and Privacy*, pages 104–121, 2015.
[8] M. Bostock, V. Ogievetsky, and J. Heer. D3: data-driven documents. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):2301–2309, 2011.
[9] R. Chang, M. Ghoniem, R. Kosara, W. Ribarsky, J. Yang, E. Suma, C. Ziemkiewicz, D. Kern, and A. Sudjianto. WireVis: Visualization of categorical, time-varying data from financial transactions. In *Proc. IEEE Symp. on Visual Analytics Science and Technology*, pages 155–162, 2007.
[10] Coinalytics. Jarvis. `http://coinalytics.co/jarvis.html`.
[11] W. Lu. bitcoin-tx-graph-visualizer. `http://www.npmjs.com/package/bitcoin-tx-graph-visualizer`.
[12] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage. A fistful of Bitcoins: Characterizing payments among men with no names. In *Proc. ACM Internet Measurement Conf.*, IMC, pages 127–140, 2013.
[13] M. Moser, R. Bohme, and D. Breuker. An inquiry into money laundering tools in the Bitcoin ecosystem. In *eCrime Researchers Summit*, eCRS, pages 1–14, 2013.
[14] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.
[15] D. Phan, J. Gerth, M. Lee, A. Paepcke, and T. Winograd. Visual analysis of network flow data with timelines and event plots. In *Proc. Workshop on Visualization for Computer Security*, pages 85–99, 2007.
[16] F. Reid and M. Harrigan. An analysis of anonymity in the Bitcoin system. In Y. Altshuler, Y. Elovici, A. B. Cremers, N. Aharony, and A. Pentland, editors, *Security and Privacy in Social Networks*, pages 197–223. Springer, 2013.
[17] D. Ron and A. Shamir. Quantitative analysis of the full Bitcoin transaction graph. In A.-R. Sadeghi, editor, *Financial Cryptography and Data Security*, volume 7859 of *LNCS*, pages 6–24. Springer, 2013.
[18] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proc. IEEE Symp. on Visual Languages*, pages 336–343, 1996.
[19] D. Staheli, T. Yu, R. Crouser, S. Damodaran, K. Nam, D. O'Gwynn, S. McKenna, and L. Harrison. Visualization evaluation for cyber security: Trends and future directions. In *Proc. Workshop on Visualization for Cyber Security*, VizSec, pages 49–56. ACM, 2014.
[20] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Trans. on Systems, Man and Cybernetics*, 11(2):109–125, 1981.

Appendix

In this Appendix we show another example of a visualization constructed by BitConeView. In the following, we refer to Figure 8.

The top rectangle corresponds to block 360524. The light blue small rectangle within it represents the 2.96 BTCs entering the BitCone. The tiny dark gray rectangle in the bottom left represents a utxo of 0.008 BTCs. Those BTCs will be spent at a time beyond the time interval of the BitCone. The fact that the next block shown, 360530, is one hour later means that no node (tx) of the tx-graph belonging to the BitCone is contained in blocks 365025 to 365029.

The rectangle for block 360530 has no input entering the BitCone from the outside (no intruder). It has a very small utxo. The next Block, 360541, has neither inputs nor utxos. It is shown because it contains nodes of the tx-graph belonging to the BitCone. Hence, the user can click on it to inspect that portion of the graph. None of the next blocks until 360659 have an intruder. Hence, the right sides of their rectangles are vertically aligned. Several intruders enter the BitCone at block 360676, almost doubling the amount of BTCs in the BitCone. This is clearly represented by the x-coordinate of the right side of the rectangle. Observe that even if the amount doubles, the purity does not drop significantly. This is due to the fact that most of the new BTCs mix with a small portion of the original amount.

The black rectangle at the bottom left corner of block 360676 represents a utxo of 0.48 BTCs that remains unspent till the present time. In the rest of the time interval, other intruders arrive and other utxos show up. At the end of the time interval, there are about 23 BTCs in the BitCone.

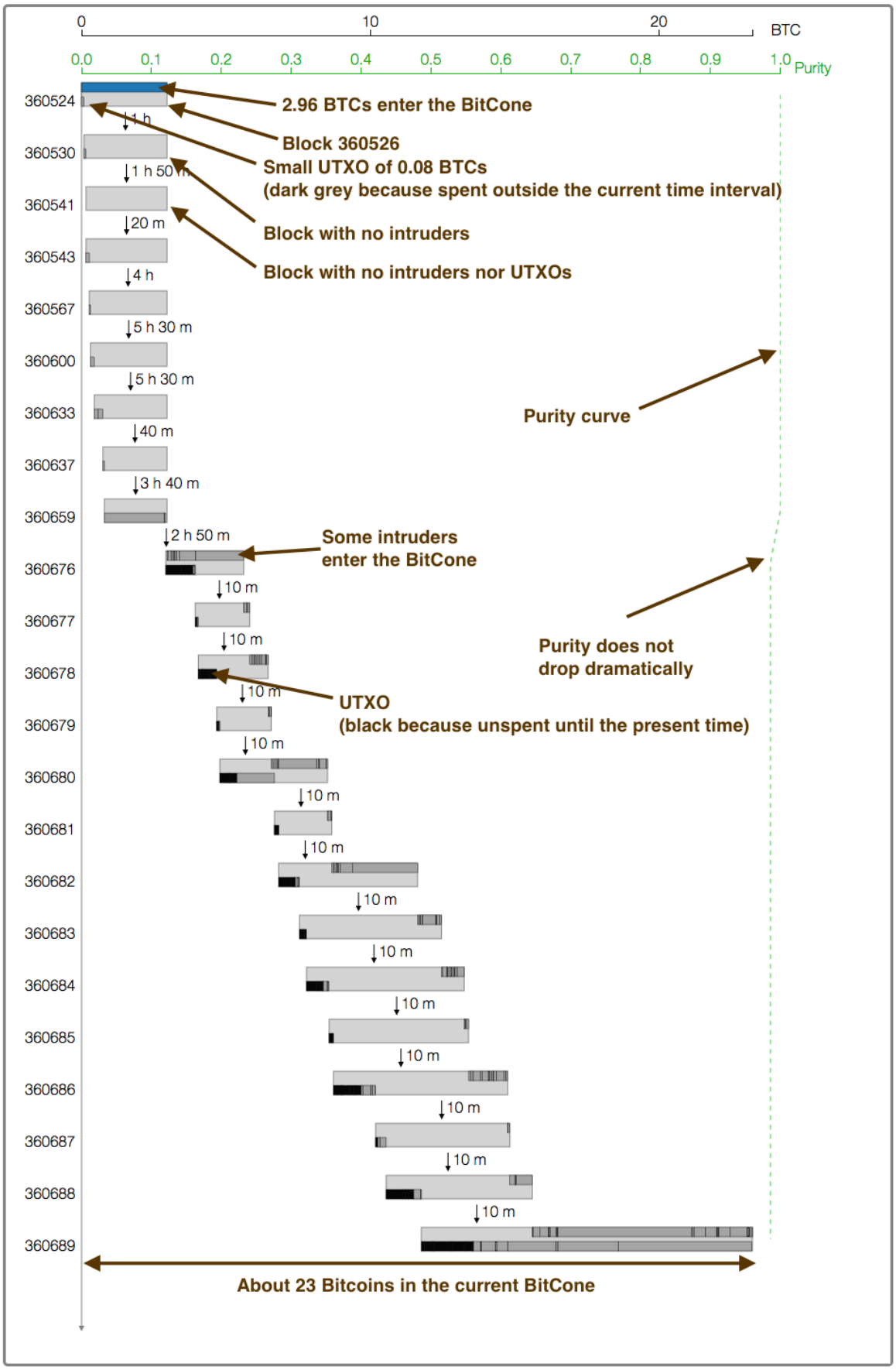

Figure 8. BitCone of tx `f97aa3841863b05d941f63d51c243f3e901de21506a234a9ef9ef3c645f8cd05` and time 360689.