O Bitcoin Where Art Thou? Insight into Large-Scale Transaction Graphs.

Bernhard Haslhofer
Austrian Institute of Technology
Digital Insight Lab
Vienna, Austria
bernhard.haslhofer@ait.ac.at

Roman Karl
Austrian Institute of Technology
Digital Insight Lab
Vienna, Austria
roman.karl@ait.ac.at

Erwin Filtz
Vienna University of Economics and Business
Institute for Information Business
Vienna, Austria
erwin.filtz@wu.ac.at

ABSTRACT

Bitcoin is a rising digital currency and exemplifies the growing need for systematically gathering and analyzing public transaction data sets such as the *blockchain*. However, the blockchain in its raw form is just a large ledger listing transfers of currency units between alphanumeric character strings, without revealing contextually relevant real-world information. In this demo, we present GraphSense, which is a solution that applies a graph-centric perspective on digital currency transactions. It allows users to explore transactions and follow the money flow, facilitates analytics by semantically enriching the transaction graph, supports path and graph pattern search, and guides analysts to anomalous data points. To deal with the growing volume and velocity of transaction data, we implemented our solution on a horizontally scalable data processing and analytics infrastructure. Given the ongoing digital transformation in financial services and technologies, we believe that our approach contributes to development of analytics solutions for digital currency ecosystems, which is relevant in fields such as financial analytics, law enforcement, or scientific research.

CCS Concepts

•**Information systems** → **Data analytics;**

Keywords

Bitcoin; Graph Processing; Anomaly Detection

1. INTRODUCTION

The rise of digital currencies such as Bitcoin [7] is an indicator for the ongoing digital transformation in financial technologies. In contrast to existing fiat currencies (e.g., EUR, USD), digital currency units are generated without central control (e.g., national banks) by a decentralized network of so called *miners*. Anyone can transfer money to anyone else in the world under minimal transaction costs, without disclosing real-world identity information, and without relying on traditional payment processors (e.g., banks). All ever executed transactions are accessible in the publicly visible *blockchain* and can systematically be gathered and analyzed for purposes such as financial analytics, scientific investigations, or law enforcement.

However, raw Bitcoin transactions only represent flows of currency units between alphanumeric character strings, without revealing contextually relevant real-world information. Therefore, within the GraphSense project, we aim at developing algorithmic solutions for real-time analytics of large-scale transaction graphs generated from transaction ledgers such as the Bitcoin blockchain. The expected result is a system that allows analysts to (i) explore transactions and trace the flow of digital currency units, (ii) make use of automated (address clustering) and manual semantic enrichment (tagging) techniques, (iii) search for paths and graph patterns within various graph perspectives, and (iv) filter data points and patterns deviating from typical structures by applying anomaly detection techniques. The combination of these features could, for instance, help analysts to identify illegitimate business transactions or trace fraudulent activities. The intended solution should be applicable for Bitcoin and any other form of digital currency transactions (e.g., Ethereum[1]).

Existing work in the field of cryptocurrency research already shows the potential of graph-based approaches for analyzing the structure and dynamics of digital currency ecosystems: using a variety of heuristics [7, 10, 8, 3] it is possible to aggregate Bitcoin addresses into clusters, which indicate common ownership of addresses; using so-called *ego-net* features in combination with unsupervised k-means clustering, it is possible to detect anomalous behavior in the Bitcoin transaction network [6]. However, algorithms presented in these works operate on static subsets of the blockchain and do not allow for interactive real-time analytics.

Therefore, the technical and scientific challenge addressed in GraphSense lies in the growing volume, velocity and semantically poor nature of digital currency transaction data. As of May 2016, the Bitcoin address graph consists of 144 million nodes (addresses) and 1 billion edges (transactions) and expands by roughly 200,000 transactions a day. We address this challenge by building analytics, semantic enrichment, and anomaly detection algorithms on horizontally scalable infrastructures and test their applicability using the dynamically growing Bitcoin blockchain dataset. Our main aims (and preliminary contributions) can be summarized as follows:

- We apply a graph-centric perspective on digital cur-

[1]https://www.ethereum.org/

SEMANTICS 2016: Posters and Demos Track
September 13-14, 2016, Leipzig, Germany

rency transactions and implement graph exploration and analytics procedures on horizontally scalable infrastructures with real-time streaming and graph computation capabilities (e.g., Apache Spark).

- We enrich the transaction graph using well-known address clustering heuristics and contextually relevant external knowledge expressed as tags.

- We investigate path and pattern search procedures, which allow the investigation of currency unit flows between Bitcoin addresses.

- We use anomaly detection techniques for identifying nodes that might be of interest to the analyst.

Development and algorithmic design is still in an early stage and a first public demo release is available and accessible via our project website[2]. It demonstrates an analytics infrastructure that supports stakeholders from various domains (e.g., FinTech, Law Enforcement) in interactively exploring the Bitcoin blockchain from a graph perspective.

2. BACKGROUND

The blockchain takes a central role in the Bitcoin ecosystem. It represents a complete and timely ordered history of all transactions ever carried out within the Bitcoin network and is continuously synchronized by Bitcoin clients over the Bitcoin P2P network. The core entities in this database are *blocks*, *transactions*, and *addresses*.

A *block* in the Bitcoin blockchain aggregates one or more transactions, provides a header with additional descriptive metadata (e.g., creation date, sequential id), and also contains a hash-value, which is computed over selected header fields and a hash over the set of encapsulated transactions. The hash serves as unique identifier for a block and is also used to refer to the previous (and next) block, which ensures that transactions within a block and the block sequence itself are non-mutable and therefore tamper-proof.

A Bitcoin *transaction* can be regarded as a generalization of regular bank transaction allowing multiple sending addresses (inputs) and multiple receiving addresses (outputs). Each transaction input must refer to an output of a previous transaction and contain a signature verifying ownership of Satoshis (Bitcoin subunits) associated with that output. A transaction output comprises the recipient's Bitcoin address and number of Satoshis credited to that address. Since transactions outputs can only be used once, it is possible to distinguish between *spent* and *unspent* transaction outputs.

An *address* is a hash over the public key of an asymmetric key-pair generated by the user. It can be shared publicly just like a traditional bank account is shared for receiving payments. However, the corresponding private key must be kept private in order to unlock and spend Bitcoins associated with addresses in the public blockchain. Users can use *wallet* software (e.g., Bitcoin Core, blockchain.info) to generate an arbitrary number of public/private key and to keep their private keys.

A number of different graph-centric perspectives have been proposed for Bitcoin so far:

[2]http://graphsense.info

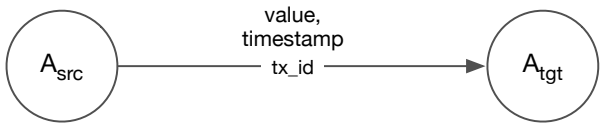

Figure 1: Bitcoin address graph model.

- *Transaction Graph* [8]: represents the flow of Bitcoins between transactions over time. Each vertex represents a Bitcoin transaction and each directed edge an output connecting two transactions with each other. Each edge also includes the transferred Bitcoin value and a timestamp.

- *Address Graph* [3]: represents the flow of Bitcoins between addresses. Each vertex denotes an address and each directed edge a particular transaction from a source address to a target address.

- *Entity Graph* [8, 10, 3]: represents the flow of Bitcoins between real-world entities over time. Each vertex represents a cluster of addresses possibly belonging to the same entity. A directed edge represents an input-output pair of a single transaction, where the input and output addresses are part of the source and target addresses.

3. APPROACH

Existing services, such as *blockchain.info*, already allow manual inspection of the main blockchain entities (block, transaction, address), but do not offer graph-centric exploration and analytics features. Our aim is to provide such features as well as algorithmic approaches for reducing the complexity of graphs, enriching graphs with contextual information, and detecting anomalous nodes.

3.1 Graph-centric transaction perspective

At the moment, GraphSense implements one specific type of graph-centric perspective: the so called *address graph*, which is a property graph model [9] representing the flow of Bitcoins between addresses over time. As shown in Figure 1, each vertex represents a Bitcoin address identified by a 26-35 alphanumeric character string. Each edge represents a transaction identified by its 32-bit hash and carries additional properties, such as the transaction time stamp and transaction value.

The address graph provides a useful abstraction for manually exploring and tracing flows of currency units through the Bitcoin ecosystem and identifying recurring patterns in transactions, such as frequently used target addresses. In combination with address clustering and node enrichment techniques, it can also be used for finding known, deanonymized addresses in the digital trace of digital currencies.

Since the volume of Bitcoin transactions already touches the physical boundaries of single-machine analytics environments, we implemented our graph construction and analytics procedures using Apache Spark[3], which provides horizontal scalability and streaming capabilities for dynamically

[3]https://spark.apache.org/

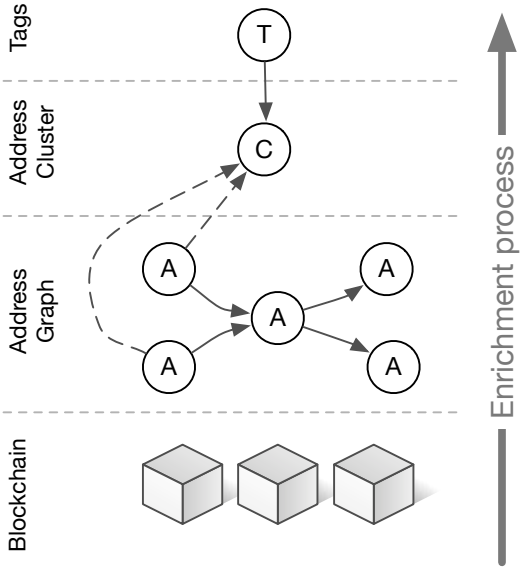

Figure 2: Transaction graph enrichment.

updating GraphSense internal data representations from raw blockchain data. With GraphX and the recently announced GraphFrame API[4], Spark also provides a variety of graph algorithms such as PageRank or computation of connected components as well as the possibility to implement custom graph algorithms using the Pregel API.

3.2 Transaction graph enrichment

The original Bitcoin paper [7] states that "a new key pair should be used for each transaction to keep them from being linked to a common owner" and many Bitcoin users follow that recommendation, as can be observed by the monthly number of newly generated Bitcoin addresses, which is greater than the monthly number of transactions [4]. Thus, without enrichment techniques, analysts are confronted with more than 120 million address nodes without further contextual information.

We address that problem by a two-step enrichment process, which is illustrated in Figure 2: in the first step, we automatically apply heuristics to group addresses (A) observed in the blockchain into address clusters (C), which are likely owned by the same real-world entity. In the next step, we offer the possibility to tag (T) single addresses — either manually or via bulk upload — with contextually relevant information, such as ownership information or appearance on certain Web sites. From a tag, which is explicitly assigned to a single address, we can then infer implicit contextual information for all other addresses in the same address cluster and consider this information in transaction graph exploration or search tasks.

Given the presence of roughly 2,000 "super-clusters", which contain approximately 16% of all addresses and are responsible for 23% of all transaction outputs, those transaction graph enrichment techniques also reduce complexity for analysts. It has also been shown that address clusters with high degree centrality often represent major darknet markets, gambling services, exchanges, or mining pools [4], which can be tagged accordingly.

[4]https://github.com/graphframes/graphframes

3.3 Path and graph pattern search

A fundamental requirement often requested by analysts is the ability to execute path queries. A typical use case is finding a path between given Bitcoin addresses or addresses carrying certain tags (e.g., exchange service). Another use case is to find the subgraph spanning a set of source and target nodes.

Traversing and finding the shortest path in a graph is a known and extensively studied problem [11, 5, 2]. However, in the context of Bitcoins, we are dealing with large graphs containing millions of nodes with a highly skewed degree distribution and billions of edges possibly partitioned over a number of physical machines. Therefore, we investigate novel optimized graph traversal algorithms taking into account Bitcoin specific graph properties such as temporal order of transactions, node degrees, or cluster membership of addresses. Furthermore, we aim at implementing those algorithms on top of distributed analytics infrastructures such as Apache Spark.

For such queries, our initial release implements a bidirectional breath-first search algorithm, which is supported by a number of statistics computed over the entire blockchain using Apache Spark.

3.4 Anomaly detection

Anomaly detection refers to the problem of finding patterns in data that do not conform to normal and expected behavior. Nonconforming patterns are often referred to as *anomalies* or *outliers* and have the common characteristics that they have real-world relevance and are interesting to the analyst [1]. In the context of Bitcoin, anomaly detection algorithms could guide analysts to specific addresses in a large cluster, which deviate from others by characteristics such as transaction frequency, node degree, or transaction volume. On a more macroscopic level, anomaly detection algorithms could provide insight into the state of the overall Bitcoin ecosystem and indicate anomalous events, such as attacks on the Bitcoin peer-to-peer network.

On a technical level, the task of anomaly detection algorithms is to compute and assign continuous anomaly scores to explicit (address, transaction) and derived (address cluster) nodes in the Bitcoin transaction network and flag those that exceed a certain threshold. The absence of annotated training data leaves us with two possible approaches: (i) parametric statistical techniques, which build a stochastic model from historical data to determine if a new observation is anomalous or not, and (ii) unsupervised clustering techniques, which assume that normal data instances occur in dense neighborhoods, while anomalies occur far from their closest neighbor. At the moment, we investigate how to compute anomalies for addresses in addresses clusters and compare performance and effectiveness of both technical approaches.

4. PRELIMINARY RESULTS AND DEMO

Our current implementation (release 0.1) consists of several components: a utility for extracting transaction data from the blockchain, a data transformation pipeline built on Apache Spark, a data storage backend exposing a REST API, and an initial Web interface, which supports users in

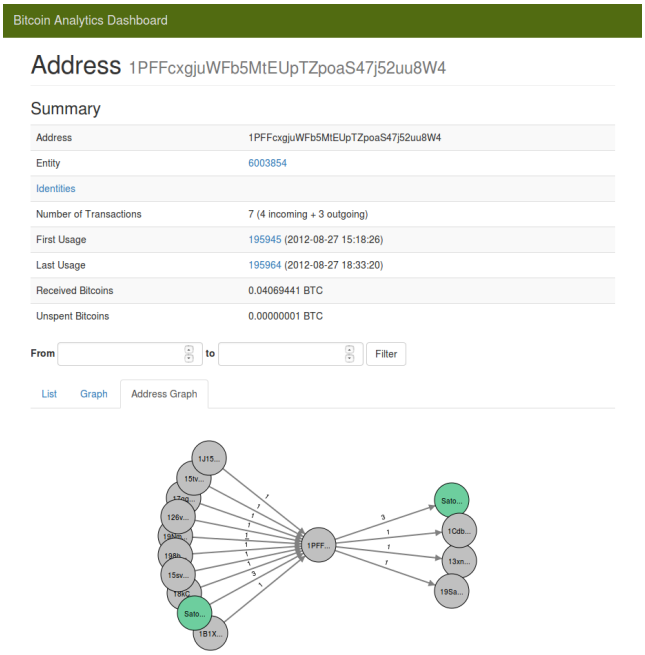

Figure 3: Address view in web interface.

the following tasks:

- **Search transaction graph**: using a Google-like search interface, users can search for Bitcoin blocks and transactions as well as for addresses either by their identifier or by assigned tags.
- **Search path**: by entering a source and target address, users can find and inspect the shortest path between these nodes.
- **Explore and traverse transaction graph**: all Bitcoin entities (blocks, transactions, addresses) are exposed as first class resources identified by a unique URI. Relationships between entities are represented as HTTP links.
- **Inspect address cluster**: each address is assigned to a cluster, which can be further inspected.
- **Explore address graph**: for each address, we display a reduced ego-net graph (see Figure 3), which allows users to inspect and traverse the address graph.

5. SUMMARY AND FUTURE WORK

GraphSense is a graph-centric analytics solution for digital currencies and is built on a horizontally scalable data processing and analytics platform. It allows users to explore transactions and follow the money flow, facilitates analytics by semantically enriching the transaction graph, supports search for paths and graph patterns, and guides analysts to anomalous data patterns. Its first release already provides a number of features for exploring, searching, and inspecting a semantically enriched transaction graph, which originates from the Bitcoin ecosystem.

Our future work will focus on graph computation scalability issues (address cluster, path search), implementation of known and investigation of novel clustering heuristics, improved user interaction, and abstracting the overall design to accommodate transactions from other digital currency systems.

6. ACKNOWLEDGMENTS

This work was funded by the Austrian research funding association (FFG) under the scope of the ICT of the Future program (contract # 849906).

7. REFERENCES

[1] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, July 2009.

[2] David Eppstein. Finding the k shortest paths. *SIAM J. Comput.*, 28(2):652–673, February 1999.

[3] M. Fleder, M. S. Kester, and S. Pillai. Bitcoin Transaction Graph Analysis. *ArXiv e-prints*, February 2015.

[4] Martin Harrigan and Christoph Fretter. The unreasonable effectiveness of address clustering. *arXiv preprint arXiv:1605.06369*, 2016.

[5] John Hershberger, Matthew Maxel, and Subhash Suri. Finding the k shortest simple paths: A new algorithm and its implementation. *ACM Trans. Algorithms*, 3(4), November 2007.

[6] Jason Hirshman, Yifei Huang, and Stephen Macke. Unsupervised approaches to detecting anomalous behavior in the bitcoin transaction network. Technical report, Stanford University, 2013.

[7] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008. Available at: https://bitcoin.org/bitcoin.pdf.

[8] F. Reid and M. Harrigan. An Analysis of Anonymity in the Bitcoin System. *ArXiv e-prints*, July 2011.

[9] Marko A. Rodriguez and Peter Neubauer. Constructions from dots and lines. *Bulletin of the American Society for Information Science and Technology*, 36(6):35–41, 2010.

[10] Dorit Ron and Adi Shamir. Quantitative analysis of the full bitcoin transaction graph. Cryptology ePrint Archive, Report 2012/584, 2012. http://eprint.iacr.org/.

[11] Jin Y. Yen. Finding the k shortest loopless paths in a network. *Management Science*, 17(11):712–716, 1971.