Bitcoin address analysis

Karl Söderbäck Vera Antonov
Email: {karso527, veran441}@student.liu.se
Supervisor: Niklas Carlsson, niklas.carlsson@liu.se
Project Report for Information Security Course
Linköpings universitetet, Sweden

Abstract

Using Bitcoin transfers provides great anonymity for all parties involved in a transaction. While user integrity is good, it also enables easy malicious usage of the cryptocurrency. This report addresses what information can and should be retrieved from the Bitcoin ledger about an address and its transactions, to enable further analysis that could eventually help stopping the malicious usage. A dataset is proposed and a tool that generates the proposed dataset is implemented.

1. Introduction

Traditional currency is issued by governments and has accounts managed by banks. The banks and financial institutions have become a trusted third party for online transactions. There is just one problem, the problem of trust. For traditional online transactions there is no way to make them truly non-reversible since the financial institutions have to solve disputes. When buying commerce in person, cash exchange is one such non-reversible transaction, on the internet we have to rely on third parties. A technology that can be used to create non-reversible transactions free from the connection to financial institutions was created to solve this problem by an anonymous pseudonym called Satoshi Nakamoto who introduced the currency Bitcoin[1].

Bitcoin is a virtual currency which is managed in a distributed way. Traditional currencies are usually managed by one central player, often a bank. Bitcoin is instead distributed on all of its users and holds mechanisms that keep the exchange honest and guards against single parties gaining power of the currency. This is done by keeping track of transactions and validating them in a peer-to-peer network between the bitcoin users[2].

Bitcoins are not only being utilized for good transactions but also for causes with malicious intent. This project will aim to create a tool that can be used for extracting information about Bitcoin transactions. Such information could be to find all transactions connected to an address for instance. This information could then be used in a later stage to analyze the money flow of Bitcoin in order to identify scams and other bad uses of Bitcoin. The expected result of the project is a dataset with interesting information about a set of Bitcoin transactions as well as the tool created and used to extract the dataset.

2. Theory

Bitcoin is a peer-to-peer, decentralized online crypto currency. This is one of the most widely used crypto currencies in the world today. Bitcoin uses a public ledger to keep track of the transactions made with bitcoins as well as the technology blockchain to secure the validity of the ledger. Bitcoin uses anonymous wallets and addresses for users and therefore privacy and anonymity questions connected to Bitcoin are widely discussed, both the good and bad that can come from it.

2.1 Bitcoin

The ledger is a public record of all transactions made using Bitcoin. It can be compared to someone keeping track of the tab amongst friends on a night out. The ledger is distributed so multiple nodes have copies and new transactions are broadcasted.

The ledger contains information about transactions, addresses involved in the transaction, amount of Bitcoins in the transaction and much more.

The ledger is built from the technology of Blockchain. This is a way of securing that everyone has the same information in the ledger. New transactions that are to be put into the ledger are gathered in a bundle together with the hash of the previous block on the chain. This bundle is then released to the public for anyone to find a ”proof-of-work”. The proof-of-work is a sequence of characters which inserted in the end of the given bundle give the hashed output of the block a given number of zeros in the beginning of the hash-string. This proof-of-work can only be guessed by simple trial and error. The first to find the correct proof of work is awarded with a given amount of brand new Bitcoin for their work. This is what's called mining. The block is then added to the chain.

2.2 Bitcoin privacy

Bitcoin addresses and wallets are not publicly linked in any way to a physical person. There are guidelines [3] on the internet for users of Bitcoin upon how to maintain their privacy whilst using the currency.

Users of Bitcoin are advised to avoid using addresses more than once, called "address-reuse" but rather to create a new address for each transaction. Doing this makes it harder to track a person's identity through their transactions in the ledger. Other measures to be taken is to

The privacy measures described above can indeed provide the user a higher level of anonymity but this comes with a backside. The anonymity makes it hard to trace transactions back to physical persons which can be needed for example when Bitcoin has been used in illegal activity. Since Bitcoin is an online currency and the ledger is public there are measures with which one can find out the true identity behind an address. Such measures can consist of searching the internet to see if the user has published their address together with their name on for example a blog. The identity can also be found though various providers such as a domain provider if the person has bought a domain in their own name and published their Bitcoin address there, through the ISP or through the use of an online hosting service for Bitcoin.

With the help of the ledger one can by the use of for example clustering algorithms find addresses that seem to be used by the same physical person. This can then be called a "wallet cluster"[3]. If two such clusters are known then one can see that transactions between addresses in the clusters are between the two individuals.

3. Method

The aim of the project was to extract interesting information about a given address from the transactions in the Bitcoin ledger and create a dataset that can be used for further analysis in a later stage. To do this in an efficient way, a tool that can do this in an automated way is much desired. It would be a very tedious task to manually search the Bitcoin ledger and gather all the transactions related to an address in the cases where there might be a lot of transactions. An automated tool can also more quickly find other related information. This tool should be able to extract information about transactions related to a Bitcoin address and it could be extended to find other interesting related information as well.

The script is the main component of this project. The tool was implemented as a Python based command line application that takes the Bitcoin address to find information about as a command line parameter. It yields output in form of a csv file for the transactions it has been involved in, a txt file for the resulting addresses and a json file with all the data collected about the address from the API-call.

3.1 Programming language

For implementing the tool, Python was chosen as the programming language. It is a quick and easy language for writing small scripts and makes it easy writing to and from files. Also if the script would in the future want to be extended with for example crawling the web for further information about the address Python supports good tools for this, for example Selenium.

3.2 Choice of API

To avoid doing redundant work, available Bitcoin ledger APIs were investigated. There are already several APIs created and made available for use that can extract the type of information that is relevant for this tool. Thus there is no reason to implement methods for accessing the Bitcoin ledger from scratch.

Three different REST APIs were compared; the Blockchain Data API provided by blockchain.com, the

Address- and Transaction API provided by blockcypher.com and finally the API provided by btc.com. A summary of the comparison can be seen in Table 1.

After comparing the three most interesting APIs for our particular needs, the API provided by Blockcypher looked the most promising and thus it became the API used in our tool. This decision was made by reading the usage documentation where their API seemed to be able to extract the most detailed information in the most smooth way, compared to the other two APIs. By using the three endpoints described in Table 1 related to Blockcypher, the API could extract all required information using only a Bitcoin address. This all made it feel like the best fit for what was to be achieved in the tool.

3.3 Evaluation and Comparison of API:s

At this stage, we were aware of one other project where similar work is being made. This is another group of students created their version of this tool with the same end goal as this project. They chose a similar approach where they wanted to utilize existing APIs for accessing and gathering information in the Bitcoin ledger. After communicating our choice of using the Blockcypher API, the other group chose to use the Blockchain API in order to make the different projects a bit more diverse.

The method used in the project seems to be a reasonable approach to achieve an efficient solution in a time efficient manner. It enabled getting reasonable results and datasets without demanding a complex or time consuming implementation.

3.4 Blockcypher

The Blockcypher API [6] is available through a language SDK for Python. This is installed by running

```
pip install blockcypher .
```

Then it can be used by importing blockcypher to the Python script. Then all API functions can be reached by simply calling the function. The data is returned as a JSON object.

API provider	**Relevant endpoints**	**Useful**
Blockchain.com[5]	Single Transaction - get transaction info using transaction hash Single Address - get info about address (transactions, balance etc…) Multiple Address - get info about multiple addresses in one query Limit 50 transactions	Useful
Blockcypher.com [6]	Address Endpoint - get info about an address and its related transactions Address Full EndPoint - get info about an address and its complete transaction records Transaction Hash Endpoint - detailed information about a specific transaction	Very useful
Btc.com [7]	Get Transactions - get info about one or more transactions Get Addresses - get info about addresses, lacks detailed info about related transactions	Lacks wanted detail level, not very useful

Table 1. A comparison of three available Bitcoin ledger APIs

3.5 The script

The script takes a txt file containing Bitcoin addresses, one address per row. The script will then parse one address at a time. A lightweight API call is made to Blockcypher using the get_address_overview() method to check whether or not the address is involved in any transactions. If not so, the address is discarded and a new address from the input file is chosen. If it has

transactions an extended API call using the get_address_full() method is conducted. This call returns a JSON object of the full data associated with the address. There is a limit of getting maximum 50 transactions in one API call and therefore the "hasMore" flag is checked to see whether or not we received all addresses. If not, the call is made again but this time using the "before" argument to get older transactions which occurred before the oldest of our currently received transactions.

The JSON object for the address is then parsed and interesting information is collected for each transaction. More about which data is collected can be found in the Output section below.

3.6 Output

Output is generated into a folder named after the address. Inside this folder, three different output files are located:

- A .csv file with all the transactions related to the address as well as metadata for each transaction..
- A .json file which contains the complete JSON object retrieved from the Blockcypher get_address_full() method call.
- A .txt file which lists all the addresses which are present as contributors to an input transaction when the given address is making an input to a transaction.

The csv file contains the following information of each transaction from the ledger where the given address has been involved:

- Transaction Hash
- Timestamp
- Sent/Received
- Address
- Transaction value
- Total transaction value

A further explanation of the fields can be found in Appendix 1.

4. Results

Using our method, we were able to successfully create datasets connected to bitcoin addresses. The result is a Python script called lookup.py that takes a list of bitcoin addresses as a command line parameter. The output consists of three output files:

- full_info_{address}.json,
- transactions_{address}.csv
- related_addresses_{address}.txt.

The utilized Blockcypher API is limited in the amount of API calls that are allowed, the limit depends on which of their plans are used. For this project, the free plan was used which means that the script is limited to making 200 calls per hour and 2000 calls daily. In order to adjust for this limitation, the script keeps track of how many calls it makes and pauses its execution if the hourly limit is reached. Once an hour has passed, it continues execution where it was paused. If the script reaches its daily limit, the entire script execution is interrupted and it logs which address was currently being processed when interrupted.

5. Discussion

Implementing a script and gathering information about a given Bitcoin address proved to be a fairly simple task with the help of the Blockcypher API. The API is extensive and gives the possibility to retrieve a lot more information from the Bitcoin ledger than what was utilized in this project. The only downside to using the API was its limitation in allowed calls, since we only had access to the free plan. The trickier part of the project became formatting the data and pinpointing relevant information in the otherwise huge dataset that was retrieved.

To make the output relevant to the task, which is to find information about the transactions related to an address, we wanted to identify metadata that is relevant to someone analyzing the transactions in a later stage. When building the .csv file, the transaction hash was the first metadata included since it is the unique identifier of a transaction. To analyze a transaction it should be relevant to know when it was sent and also the value of the transaction, thus the timestamp and transaction value. We included the field that tells if the address has sent or received the transaction in order to know the role of the address as well as to be able to identify a chain of actions or some kind of pattern. The same idea was applied when listing the other addresses (the ones the transaction was sent either to or from) involved in the transaction.

We also wanted to list all addresses that have been involved in transactions together with the address since it could be possible to find relations and patterns there as well upon further analysis. The .json-file with the full

information dump was added in order to enable users of retrieving other information that is not included in the .csv file or the .txt file.

By extracting these three files and the respective information within them, this tool should enable and simplify analysis of Bitcoin addresses and their transactions a bit. It is simple to input a list and let the script work without any supervision. The output is then formatted in a well structured manner which is simple to browse.

After analysing the output of the Blockcypher API calls and identifying which data is relevant, the dataset created should be a great aid for further analysis.

6. Conclusion

The goal of this project was to create a tool which outputs a dataset with information about a Bitcoin address and its transactions. A dataset that can ultimately be used for further analysis to prevent the use of Bitcoin with malicious intent.

A tool was created in the form of a Python script and it is able to take a list of addresses and extract datasets for each individual address. For each address, there is a vast amount of information available in the Bitcoin ledger. This makes the need for gathering and presenting relevant parts of this data in an automated way, a necessity for enabling further analysis.

By utilizing the existing API provided by Blockcypher it was possible to provide a proper solution that is easy to use, although it has its limitations of how much it can be used in a certain time window. Apart from analyzing the data provided by the tool, the dataset could also be used to search for more information in the future, for instance by looking up information about the related addresses.

Currently, datasets for hundreds of Bitcoin addresses have been extracted using the tool. These are addresses that have been found to be involved in malicious contexts. The addresses were found in these contexts using web crawlers implemented by other student groups. The tool is ultimately supposed to be run in a much bigger capacity, as the lists of interesting addresses contains thousands. Extracting datasets for all addresses and analyzing the results are however out of scope for this project.

References

[1] Satoshi Nakamoto, ”Bitcoin: A Peer-to-Peer Electronic Cash System | Satoshi Nakamoto Institute”, okt. 31, 2008. https://nakamotoinstitute.org/bitcoin/ (åtkomstdatum apr. 02, 2020).

[2] R. Böhme, N. Christin, B. Edelman, och T. Moore, ”Bitcoin: Economics, Technology, and Governance”, J. Econ. Perspect., vol. 29, nr 2, s. 213–238, maj 2015, doi: 10.1257/jep.29.2.213.

[3] M. Harrigan och C. Fretter, ”The Unreasonable Effectiveness of Address Clustering”, i 2016 Intl IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), juli 2016, s. 368–373, doi: 10.1109/UIC-ATC-ScalCom-CBDCom-IoP-SmartWorld.2016.0071.

[4] E. Androulaki, G. O. Karame, M. Roeschlin, T. Scherer, och S. Capkun, ”Evaluating User Privacy in Bitcoin”, i Financial Cryptography and Data Security, Berlin, Heidelberg, 2013, s. 34–51, doi: 10.1007/978-3-642-39884-1_4.

[5] ”Blockchain Data API - blockchain.info”. https://www.blockchain.com/api/blockchain_api (åtkomstdatum apr. 09, 2020).

[6] ”Introduction – Blockchain Developer API for Bitcoin, Ethereum, Testnet, Litecoin and More | BlockCypher”. https://www.blockcypher.com/dev/bitcoin/#introduction (åtkomstdatum apr. 09, 2020).

[7] ”API Documentation - BTC.com”. https://btc.com/api-doc (åtkomstdatum apr. 09, 2020).

Appendix A

This appendix is an explanation of the fields in the csv output file.

Transaction Hash
The transaction hash

Timestamp
The received timestamp for the transaction

Sent/Received
Whether ot not the transaction was sent by the address or received

Address
The other address/addresses who were on the other end of the transaction. If the transaction is of type sent this field consists of the address or addresses which received the bitcoins. If the transaction is of type received this field is of the address or addresses which sent the bitcoin.

Transaction value
The transaction value which the given address has been part of. If the transaction is of type sent, it's the sum of the unspent transaction output connected to the given address. If the transaction is of type received it's the sum of the transaction outputs connected to the address.

Total transaction value
The total value of the transaction. If the transaction is of type sent,tThis does not only include the part contributed by the analyzed address but the entire sum of the transaction.