Final Project: System for Tracing Cryptocurrency Transactions

William S. Ventura

Whiting School of Engineering, Johns Hopkins University

EN.605.662.81: Data Visualization

Dr. Jesus Caban

January 30th, 2023

Abstract

This project undertakes the task of undercover the complex transactional relationships occuring on the blockchain network. At present moment, due to the complexity of the architecture it is difficult to visualize the transactions occuring between wallets and the inflow and outflow of ethereum. This has allowed it for illicit activites to take place through the used of cross-chain bridges. This paper will attempt to utilize multi-directed network graphs and Sankey diagrams to uncover the transactional flow and processes that occur when facilitating transactions through intermediaries and through other procresses such as *staking*.

Introduction

This project will proposed a method for tracing cryptocurrency transactions. Cryptocurrencies are based on the blockchain technology and have gaining more and more adoption over the world. Introduced in 2009, Bitcoin was the first decentralized cryptocurrency developed that had gathered adoption in its early years[1]. It was used as a way for people to conduct transactions over the internet without the intervention of central banks or financial institutions. Its emergence amid the global financial crisis; that deter trust in banks and government, gave Bitcoin a golden opportunity to gain adoption among consumers. Originally utilized as the preferred currency for illicit activities, it did not gain global traction until years later with most recently in 2021 reaching a high of about $60,000 USD per Bitcoin. Since then, numerous other cryptocurrencies have been developed such as Ethereum or Solana, with numerous countries increasing their adoption of the cryptocurrency. With it’s rapid adoption and anonymity, there have been a growing concerns for its use in illicit activity. To counteract such activity, governments have imposed stricter laws and regulations to assure due diligence such as Know Your Consumer (KYC), requiring exchanges such as Coinbase and Binance to verify the identity of new users.

However, newer stricter and laws and regulations are only able to help so much as new methods are developed to help facilitate illicit activities. In November of 2022 the U.S. Attorney's office for the Eastern District of Texas charge 21 people in a transnational crypto money-laundering network, in which law enforcement officials estimate that an annual flow of over $300 million in laundered transactions was disrupted and millions were seized. Another notable case comes from August of 2022, where RenBridge, a cross-chain bridge (software applications that enable transactions to occur between various blockchains) facilitated the laundering of at least $540 million in proceeds of crimes since 2020. This is performed through a method known as chain hopping – converting one form of cryptocurrency into another and moving it across multiple blockchains[4]. Elliptic, a blockchain analytics company stated that cross-chain bridges provide "an unregulated alternative to exchanges for transferring value between blockchains."

Background

It is currently difficult for victims of hacks and law enforcement to trace the attackers or criminals laundering money and visualize it within the network since hundred of thousands of transactions are occurring in any given moment. The blockchain architecture is organized in blocks, hashes, timestamps, nonces and block data. They are defined as such below[2]:

I. **Blocks:** *a data structure within a blockchain that permanently records data. It records the most recent transactions pending to be validated by the network. Once it is validated, the block is closed and added to the blockchain and new block is created.*
II. **Hashes:** *bit strings of fixed size derived from hash functions transforming a given set of data*
III. **Time Stamps:** *The timestamp to place on the block in the blockchain*
IV. **Nonces:** *number used only once that is added to blockchain to be used to validate the block, this is the encrypted number that a "miner" must solve to validate the block and close it*
V. **Block Data:** *information within the block such as transactions*

VI. **Block header:** *contains information about the block i.e. (Hash of the previous block header, Timestamp, Nonce, Hash of Block data)*

The current block header contains the hash of the previous block header, a timestamp, a hash of the current block data, and the block data which is list of the transactions. This current block header is then made into a hash and sent to the next block. An illustration of such is shown below.

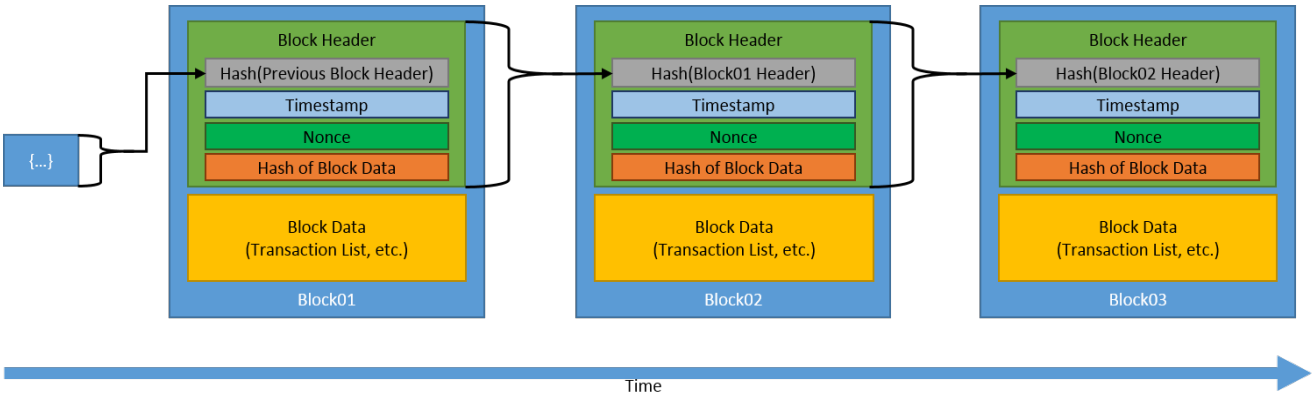

Figure 1: Blockchain Architecture

Approach

In order to visualize the transactions of a given address, this project will be making API calls to *etherscan.io*, an Ethereum block explorer, to retrieve the transaction list of a given address within a specified block range[3]. The API with return a JSON file with the results for *account_balance*, *normal_transactions*, and *internal_transactions*. An exampled of the output JSON file is shown below:

```
{
   "status":"1",
   "message":"OK",
   "result":[
      {
         "blockNumber":"14923678",
         "timeStamp":"1654646411",
         "hash":"0xc52783ad354aecc04c670047754f062e3d6d04e8f5b24774472651f9c3882c60",
         "nonce":"1",
         "blockHash":"0x7e1638fd2c6bdd05ffd83c1cf06c63e2f67d0f802084bef076d06bdcf86d1bb0",
         "transactionIndex":"61",
         "from":"0x9aa99c23f67c81701c772b106b4f83f6e858dd2e",
         "to":"",
         "value":"0",
         "gas":"6000000",
         "gasPrice":"83924748773",
         "isError":"0",
         "txreceipt_status":"1",
         "input":"0x6101606040527f6e71edae12b1b97f4d1f60370fef10105fa2faae0126114a169c64845d61
         "contractAddress":"0xc5102fe9359fd9a28f877a67e36b0f050d81a3cc",
         "cumulativeGasUsed":"10450178",
         "gasUsed":"4457269",
         "confirmations":"122485",
         "methodId":"0x61016060",
         "functionName":""
      },
```


Figure 2: JSON result of Normal Account Transactions

This JSON file will then be converted into a dataframe containing the blockNumber, timeStamp, hash, nonce, blockHash, transactionIndex and all the other variables shown above. These following variables will then be transformed:

I. *timeStamp* → datetime (%year-%month-%day - %hr : %min : %sec)

II. *value(wei)* → ether = wei/10^{18}

III. *value_usd* → ether * current conversion rate to USD

IV. *from & to* → will be trunacted to show the last 5 digits to simplify visualization since addresses can be over twenty characters long

The visualizations that will be rendered from this data will be a multi-directed network graph using NetworkX, and PyVis and will allow for the user to selected between the *Barnes Hut, Repulsion, Force Atlas 2 Based, and Hierachical Repulsion* physic models. Edge width will be scaled in regards to the transactional amount, and node size will be adjusted in regards to degrees of connections to other nodes. A Sankey Chart will also be rendered to demonstrated the flow of incoming and outcoming

transactions between wallets. This will also be useful in the case of a multi-directed network graph with hundreds of diluted transanctions such in the case of laundering software where thousands of minute transactions are disperse thousands of wallets before going back to another wallet or the user's wallet.

The color coordination of these graphs will represent green as incoming transactions to the user's wallet, and red as outgoing transactions, additionally yellow arrows will be used to represent transactions that could have occurred internally between two of the user's wallets. The project will make use of Streamlit to create an interactive dashboard that is able to retrieve the data from the API call functions, process it and display it for the users to visualize.

Results

While the application is designed so that the user is not dependent on internal data and could use any ethereum address, some addresses were provided in the "Get Started" tab in the StreamLit application. Theses were the addresses provided and for simplicity of visualization will be adjusted to display the last five characters as such:

1. 0xdC8Cc5E9dD179500F0a684F55EefbE0Bb06108Ef → 108Ef
2. 0x0713E18D6974123BBC1c019b420c4de8E63F382C → F382C
3. 0x9c5F8f1c544d21a3f7D34DC7e8DFDEce1eb28e7b → 28e7b
4. 0xC37C0d0723af8280e4442f423ba0196842f9fdEC → 9fdEC
5. 0xd2Cfda2F27227526Db4C3d73E969De676493940e → 3940e
6. 0xcC03F78f7Ed73Dfe1147F301eC76bc265ADf85D8 → f85D8

Below are the results for wallet address, 1, 2, 3, and 5:

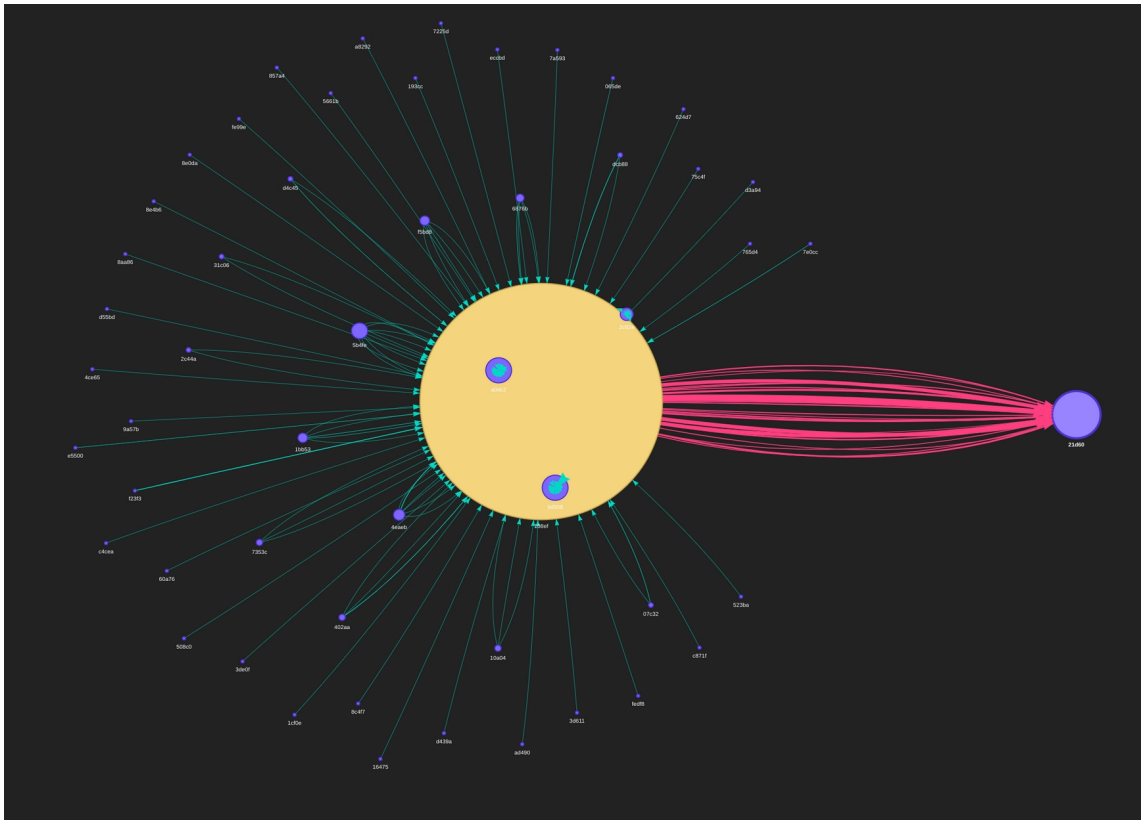

Figure 3: 108Ef - with Barnes Hut Physics

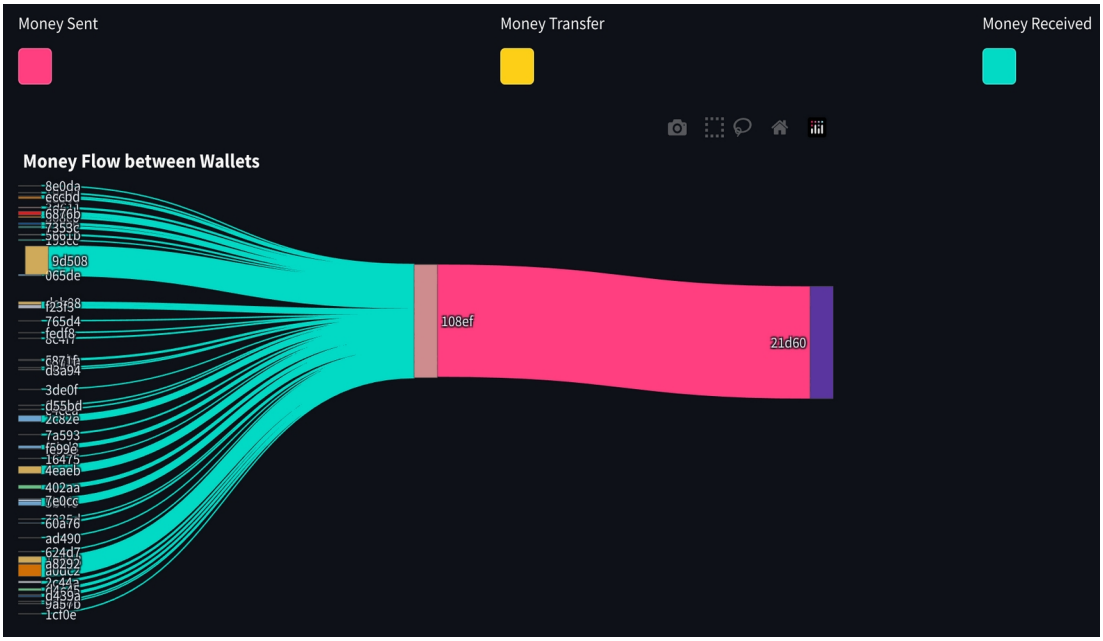

Figure 4: 108Ef - Sankey Diagram

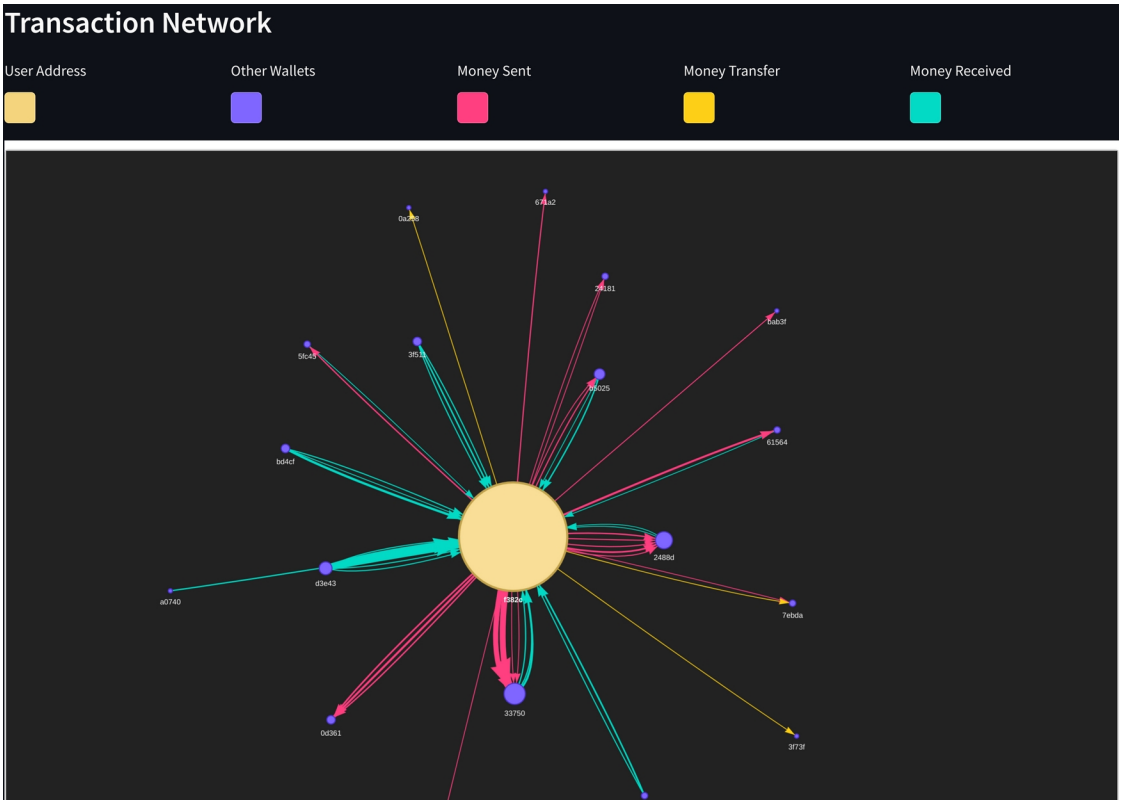

Figure 5: F382C - with Repulsion Physics

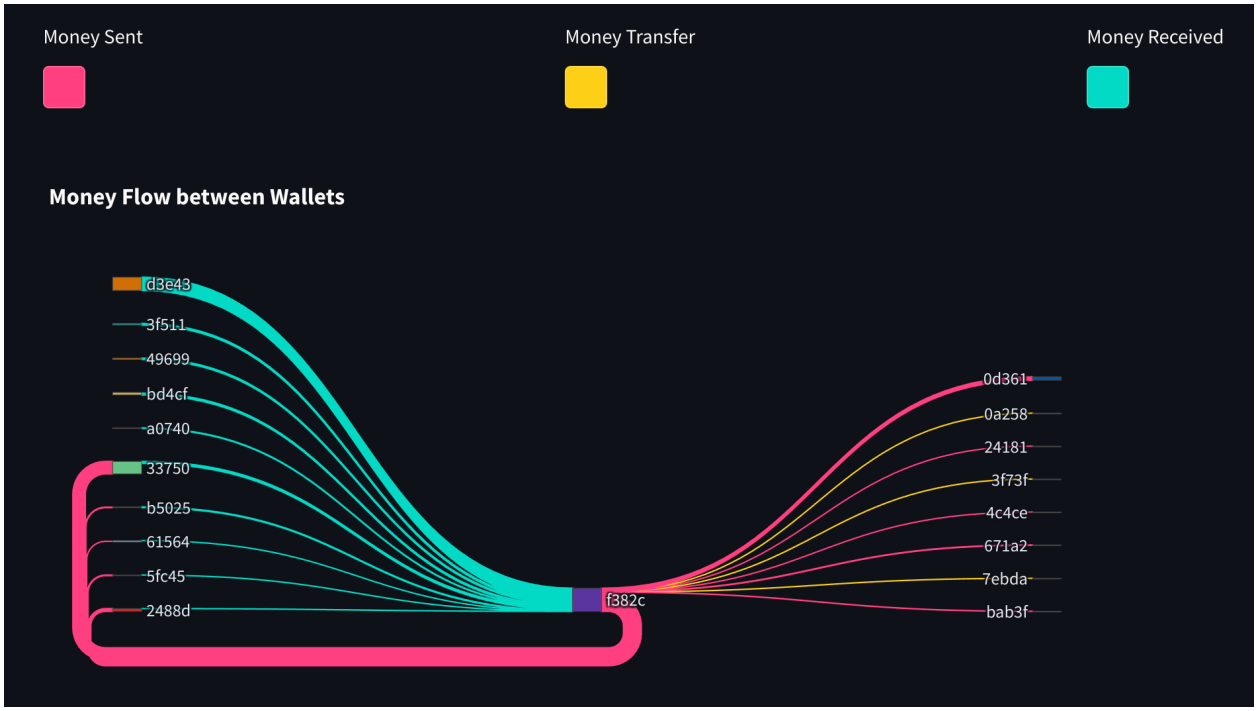

Figure 6: F382C - Sankey Diagram

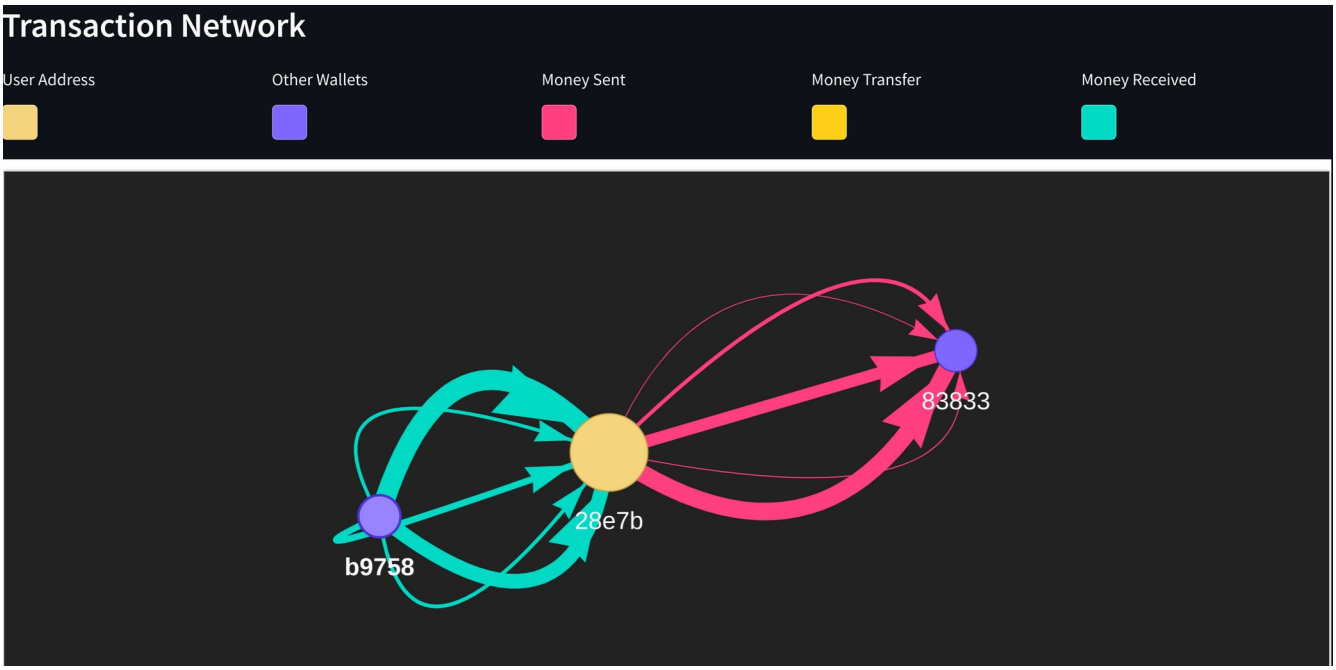

Figure 7: 28e7b - with Barnes Hut

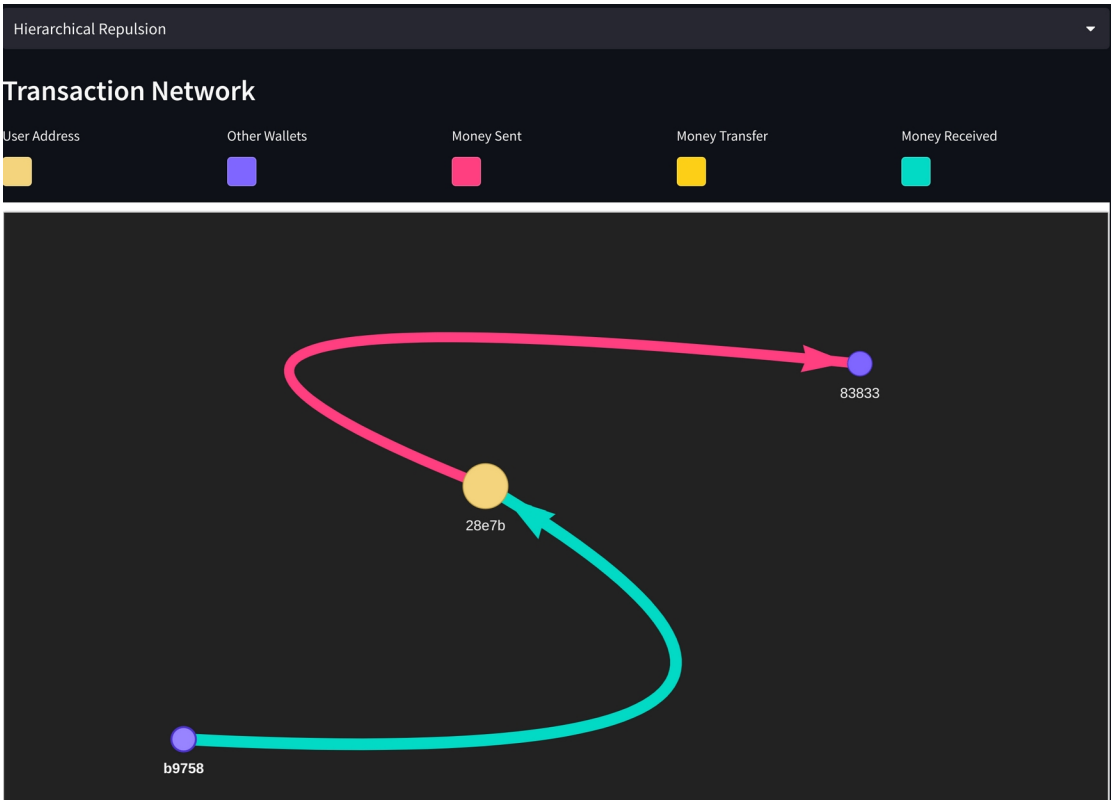

Figure 8: 28e7b - with Hierarchical Repulsion

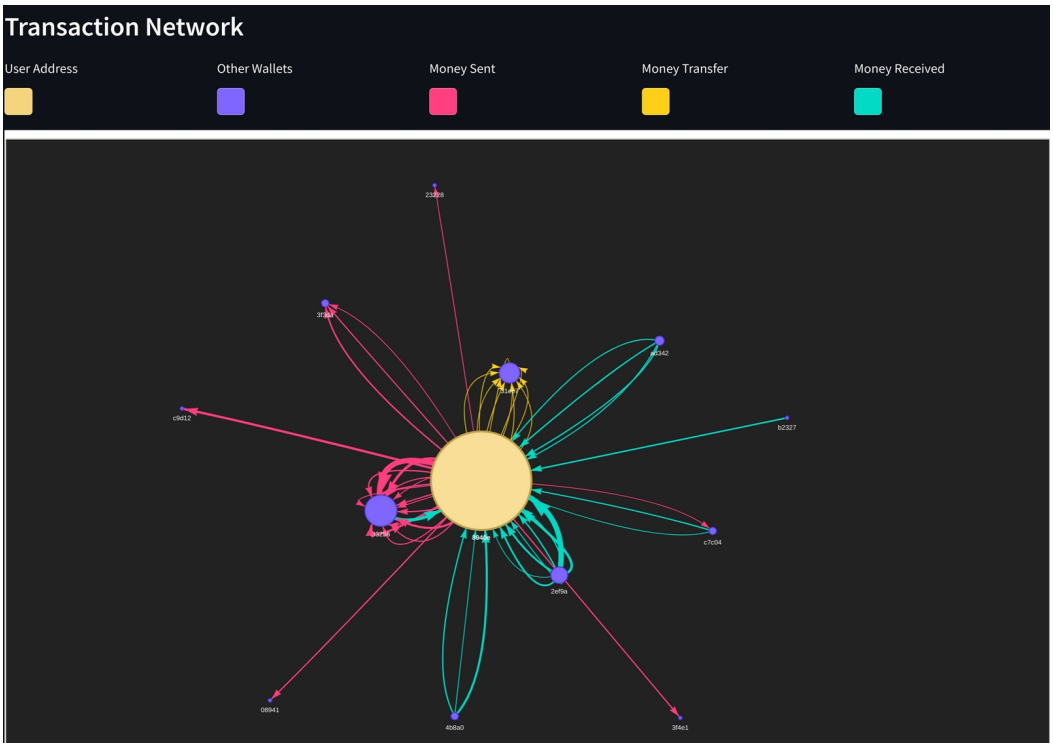

Figure 9: 3940e - with Barnes Hut

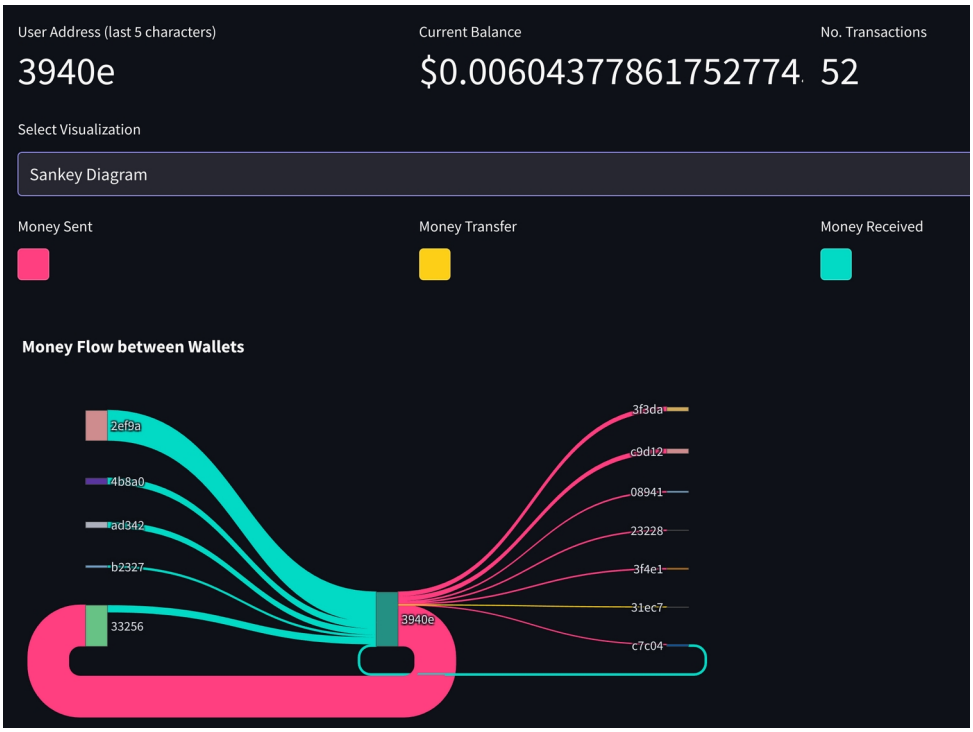

Figure 10: 3940e - Sankey Diagram

From figures 3 & 4 it appears that wallet 108Ef appears to be an intermediatary for a transaction, now this could be the case of users paying the wallet of an application and then that application send all the

transactions to the company. Figures 5,6, 9 and 10 are particulary interesting because in both of these wallets [F382C ,3940e], there multiple transactions going in and out but both of them have a significant backwards flow to one or a couple addresses in particular. One of the reasons that come to mind for this is *staking[3]*; which allows you to earn cryptocurrency as a reward for using your holdings to vouch for accuracy of transactions. As a result I believe that these wallets were able to hold until the end of the staking period in which they got paid out more than they initially deposited, hence why the there is a singular incoming and outgoing transaction loop. Figures 7, 8 are also really interesting because it depicted the usages of a singular wallet using an intermediary to facilitate a transaction to another wallet.

Conclusion

Normally visuallizing transactional processes on blockchain architectures can be pretty complicated and hard for a user. This is attributed to length hash values, multiple intermediaries, and underlying subprocesses. Through the usage of multi-directed network graphs and Sankey diagrams, it is easier to unravel what I going on within the transactions of each block. At present moment, this application can only unravel the transactions occuring within the ethereum network, however there are other processes that allow for users to perform transactions across different blockchains. Future research will focus on implementing this tracing application to trace across different blockchains in hopes of getting a clearer understanding.

References

[1] Buterin, Vitalik. A NEXT GENERATION SMART CONTRACT & DECENTRALIZED APPLICATION PLATFORM. 2015.

[2] Etherscan. “Accounts.” Etherscan, Etherscan , https://docs.etherscan.io/api-endpoints/accounts. Lasas, Karolis, et al. ‘Fraudulent Behaviour Identification in Ethereum Blockchain’. IVUS, 2020. Li, Zhen, et al. ‘Biteye: A System for Tracking Bitcoin Transactions’. 2020 Information Communication Technologies Conference (ICTC), 2020, pp. 318–322, https://doi.org10.1109/ICTC49638.2020.9123286.

[3] Shrimali, Bela, and Hiren B. Patel. ‘Blockchain State-of-the-Art: Architecture, Use Cases, Consensus, Challenges and Opportunities’. Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 9, 2022, pp. 6793–6807, https://doi.org10.1016/j.jksuci.2021.08.005.

[4] Yun, Youngju (2020) "The Influence of Blockchain Technology on Fraud and Fake Protection," OUR Journal: ODU Undergraduate Research Journal: Vol. 7, Article 8.

[5] Yousaf, Haaroon, et al. ‘Tracing Transactions across Cryptocurrency Ledgers’. Proceedings of the 28th USENIX Conference on Security Symposium, USENIX Association, 2019, pp. 837–850. SEC’19.