Forensic Science International: Digital Investigation 40 (2022) 301340

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi

DFRWS 2022 EU - Selected Papers of the Ninth Annual DFRWS Europe Conference

BlockQuery: Toward forensically sound cryptocurrency investigation

Tyler Thomas[*], Tiffanie Edwards, Ibrahim Baggili

Newhaven University, USA

ARTICLE INFO

Article history:

Keywords:
Bitcoin
Cryptocurrency forensics
Cryptocurrency investigations
Open source
Investigation confidentiality
Forensic completeness
Forensic integrity
Hierarchical deterministic (HD)
Derivation depth
Wallets
De-anonymization
Bitcoin address

ABSTRACT

Cryptocurrency transaction forensic examinations need to guarantee completeness, confidentiality, and information integrity. Our work presents BlockQuery as a proof of concept blockchain query system for Bitcoin. BlockQuery is capable of detecting transactions generated by Hierarchical Deterministic (HD) wallets that many publicly available tools cannot find due to failures in their address derivation methods. Moreover, BlockQuery does not use third party servers as data providers and operates on a local copy of the blockchain to prevent information disclosure. Compared to other Bitcoin query tools, BlockQuery was designed from a forensic standpoint and meets all four of the defined querying criteria of being open source, confidential, automatically converting key representations, and allowing the manual adjustment of derivation depth.

Published by Elsevier Ltd.

1. Introduction

The decentralized and public nature of the blockchain ledger presents several unique challenges to the validity of a cryptocurrency investigation. For a cryptocurrency examination to be forensically sound, we assert that the following criteria must be met:

- **Completeness:** Given a public key or wallet address, all transactions conducted using the key or wallet address are recovered.
- **Integrity:** The blockchain ledger being queried is identical to that which is currently accepted by the consensus network.
- **Confidentiality:** Information regarding which transactions are relevant to the examination are not being unintentionally disclosed.

The vast majority of publicly available tools for querying cryptocurrency blockchains fail to meet one or more of the aforementioned requirements. Many tools take the form of websites which query a server running a blockchain indexer. In the event that the indexing server is maintained by a third party, this may violate the confidentiality of the forensic examination by unnecessarily disclosing which accounts are subject to an ongoing investigation. An untrustworthy or compromised query service may trivially associate a law enforcement Internet Protocol (IP) address with the wallets they are querying to identify which individuals are under investigation.

Similarly, without control of the indexing server, an examiner cannot guarantee the integrity of the query responses. The results may be inaccurate, incomplete, or out of date due to programmer error or malice, and can only be confirmed by observing a local copy of the ledger. Cryptocurrencies, like Bitcoin, have many address derivation schemes and many of the publicly available tools we tested failed to account for these differences in address derivation.

It is for these reasons that a forensically sound cryptocurrency lookup platform must consist of a trusted full node running directly on the blockchain network. This allows forensic examiners to guarantee the integrity of the data they are searching through by maintaining a complete copy of the ledger that is updated in real time.

Additionally, a full node preserves the confidentiality of the investigation by eliminating the need for a third party to handle queries. Rather than broadcasting the specific wallet address and transaction IDs a practitioner is interested in, a full node blends into the network and passively collects blocks as they are broadcast.

[*] Corresponding author.
E-mail addresses: tthom10@unh.newhaven.edu (T. Thomas), tedwa4@unh.newhaven.edu (T. Edwards), baggili@gmail.com (I. Baggili).

https://doi.org/10.1016/j.fsidi.2022.301340
2666-2817/Published by Elsevier Ltd.

Our work contributes the following:

- We provide the primary discussion on what it means for a cryptocurrency investigation to be forensically sound.
- We present BlockQuery, an open source proof of concept blockchain query system for Bitcoin.
- We show that our approach is capable of detecting transactions generated by Hierarchical Deterministic (HD) wallets that many publicly available tools cannot find due to failures in their address derivation methods.

The remainder of this work is organized as follows: Section 2 elaborates on the differences in Bitcoin address derivation schemes. Past work is reviewed in Section 3. In Sections 4 and 5 we present the architecture of our proof of concept and compare it against existing publicly available tools. In Sections 6, 7, and 8 we discuss further implications, offer paths forward, and make closing remarks.

2. Background

In addition to potentially exposing which transactions are relevant to an ongoing investigation, existing online blockchain querying services using extended public keys fail to discover all transactions. This is the result of incomplete address derivation algorithms that do not take into account all valid public key representations detailed in Bitcoin Improvement Proposals (BIPs) 32,[1]49,[2] and84.[3]

A robust understanding of these standards is necessary to guarantee the complete discovery of possible transactions given an extended public key artifact. The principles described here are not only applicable to Bitcoin because HD wallets support multiple coin types. Other cryptocurrencies, such as Ethereum, have adopted compatible derivation.[4]

2.1. Hierarchical deterministic wallets

One of the most valuable artifacts in a cryptocurrency investigation is the extended public key. Extended keys are golden tickets for associating disparate transactions to a single point of origin. HD wallets use extended keys to compartmentalize addresses under logical “accounts”. Each account has an associated key pair, and accounts are organized hierarchically. This allows users to derive key pairs for multiple cryptocurrencies from the same master key pair. Further, multiple key pairs can then be derived from each cryptocurrency pair.

At the lowest level in the hierarchy, the keys are used to deterministically derive ephemeral wallet addresses. This scheme allows users to maintain relative anonymity across transactions by not reusing addresses while only having to maintain a single master key pair. Additionally, users may selectively publish compartments of their transaction history by sharing the extended keys used to derive those addresses.

Similarly, a forensic examiner may use extended public keys to de-anonymize portions of a subject's transaction history, assuming the key is not hardened. Hardened keys cannot be deterministically derived using the parent public key alone. The parent private key is required to ensure that the entire scheme is not de-anonymized if a public key from the top of the hierarchy is inadvertently exposed.

2.2. Bitcoin address types

At the time of writing, there were three valid Bitcoin address representations. The three address types are presented in Table 1. Each address type has a respective extended key representation used to derive addresses of that type; however, these extended key representations can readily be converted from one to another. This means that one can deterministically derive all possible wallet addresses given any non-hardened extended public key representation.

Some HD wallet implementations, such as Ledger Live, use all three Bitcoin address types. Memory forensic analysis of these applications has shown that in some cases it is only possible to recover one representation of the public key (Tyler et al., 2020). Given that most publicly available blockchain query tools do not account for this use case, it is possible that a pracitioner unfamiliar with the intricacies of Bitcoin address derivation may miss large portions of forensically relevant transaction history.

Suppose an investigator recovers an xpub and would like to find all transactions associated with it. They utilize a publicly available lookup service which correctly interprets it as a BIP44 extended key, and thus derives all relevant legacy addresses to search the blockchain for transactions. However, depending on the implementation of the wallet the key was pulled from, there may be transactions using Native SegWit or Nested SegWit addresses which will not be found.

To ensure recovery of all relevant transactions, a forensically sound blockchain lookup service cannot assume that all addresses should be derived using the same format in which their keys were presented.

3. Related work

3.1. Privacy decentralization and anonymity

Preserving the privacy of user data is not a new concept. For example, most online social networks involve users constantly sharing data over a centralized architecture. Privacy concerns are raised about user data because the central organization has access to all user data and can grant additional access to third-party entities. Social graphs, representing the interconnections, behaviours, and preferences between online social network users, contain private information that can help disclose the real identities of users (Schwittmann et al., 2014). To combat these privacy concerns, decentralized social networks were proposed. In 2013 (Schwittmann et al., 2014), examined different decentralized approaches to online social networks and compared their privacy levels. While the research concluded confidentiality can be achieved with end-to-end encryption, a persistent problem throughout the decentralized social networks was hiding the social graphs containing private user information from storage providers in distributed approaches.

In 2018 (Siddula et al., 2018), presented a survey about the privacy preservation of online social networks. The decentralized social networks PISCES and Lockr were discussed; however, PISCES was designed specifically for scalability and does not address the issue of link privacy, which is connecting one user back to another in a social graph.

3.2. Anonymity and transaction analysis

Directly related to link privacy and social graphs is the privacy and anonymity concerns of Bitcoin transactions. Since the rise of Bitcoin and other cryptocurrencies, the limitations of anonymity and privacy in digital currencies has been researched in many ways. Due to Bitcoin's popularity, substantial research has been

[1] https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki.
[2] https://github.com/bitcoin/bips/blob/master/bip-0049.mediawiki.
[3] https://github.com/bitcoin/bips/blob/master/bip-0084.mediawiki.
[4] https://eips.ethereum.org/EIPS/eip-2386.

Table 1
Bitcoin address representations.

Key Type	BIP	Prefix	Transaction Type	Example
Xpub	32/44	1	P2PKH (Legacy)	17 VZNX1SN5NtKa8UQFxwQbFeFc3iqRYhem
Ypub	49	3	P2SH (Nested SegWit)	3P8mzFSHtFXEPCYForFEkjpdnUHC3HqCnN
Zpub	84	bc1	P2WPKH (Native SegWit)	bc1qrde0awacdg266fvhj2fkyvuksystx2snsn4scv

conducted regarding de-anonymizing Bitcoin users. Interacting directly with Bitcoin users and posing as a buyer is one way to de-anonymize the identity of specific merchants, which was researched by (Meiklejohn et al., 2013) with the Silk Road and Mt. Gox merchants.

Another de-anonymization method is directly analyzing the Bitcoin transaction graph, similar to a social graph, to perform anonymity analysis. This was first studied by (Reid and Harrigan, 2012), who constructed two network representations to associate multiple public keys with each other and with other identifying user information. This method of mapping is also used by (Meiklejohn et al., 2013) when directly interacting with Bitcoin merchants by linking unknown addresses to known ones.

From a different standpoint (Baumann et al., 2014), suggested using IP addresses and geographical location data to have a more robust understanding of the structure of the Bitcoin transaction graph. With a similar approach (Koshy et al., 2014), set out to de-anonymize Bitcoin users at the IP level by leveraging the relay traffic in a Bitcoin network. The source IP address of the Bitcoin transactions are linked to geographical locations, which can then help link the IP addresses to real identities. The authors concluded mapping IP addresses to Bitcoin addresses is difficult with only network traffic analysis but possible to some degree.

With the use of publicly available data (FlederMichael, 2015), set out to de-anonymize Bitcoin user identities by first utilizing web scraping techniques to collect Bitcoin addresses from public forums, and then matching users to specific transactions with incomplete transaction data. The publicly available data and Bitcoin's transaction ledger were able to successfully help the researchers match user identities to certain transactions.

3.3. *Blockchain forensics*

In addition to examining ways of mapping Bitcoin and other cryptocurrency wallet addresses to user information, researchers turned to exploring data found at the blockchain level. Blockchain technologies have more use than just cryptocurrency. Blockchains act as a distributed peer-to-peer (P2P) pseudonymous network where anyone can view transactions between individuals (Ricci and BaggiliFrank, 2019). Although works such as (Cebe et al., 2018; Ryu et al., 2019; Li et al., 2019; Al-Khateeb et al., 2019) proposed using blockchain technologies to support other areas of digital forensics, research on the forensics of the blockchain itself has not been extensively studied.

(Zhang et al., 2019) studied attacks on distributed P2P storage networks and used StorJ, a blockchain-based cloud storage network, as an example. The results indicated the privacy of the data on such networks may be compromised after it is stored. The researchers also called for the need of future studies on other P2P cloud storage networks. Following that (Ricci and BaggiliFrank, 2019), explored the investigative potential of examining artifacts produced by StorJ. The research called for an in-depth study of all forensics artifacts produced by StorJ and more studies on blockchain-based storage platforms.

Our current work builds on past work, FORESHADOW, which extracted forensic data from the memory of cryptocurrency hardware wallets (Tyler et al., 2020). The extracted data could then be used to associate a hardware wallet with a computer and allow an observer to de-anonymize all past and future transactions due to hierarchical deterministic wallet address derivation.

4. Methodology

This work employed a constructive methodology to present an example implementation of a forensically sound cryptocurrency investigation platform. The following section details the design and development process.

4.1. *Software architecture*

Illustrated in Fig. 1, the architecture of our proof of concept consists of the following microservices packaged in Docker containers:

Bitcoin node: A standard Bitcoin JSON-RPC API server fully synced with the current state of the blockchain.

Indexer: This service processes and indexes the raw block data from the node for quick and easy querying.

Web application: The user interface for making queries and exploring discovered transactions.

4.1.1. *Architecture - bitcoin node*

The Bitcoin Core daemon, or *Bitcoind*, is a widely adopted implementation of the Bitcoin protocol. It is traditionally used as a backend service for mining programs and wallets. *Bitcoind* was chosen as the Bitcoin protocol implementation for this proof of concept because of its ease of use, customization, and integration into a variety of open source indexers. However, any Bitcoin protocol implementation which exposes the Bitcoin JSON-RPC API and can be configured to passively collect blocks may be used.

4.1.2. *Architecture - indexer*

An indexer is required because *Bitcoind* does not provide an API call to retrieve the complete list of transactions a given wallet address participated in. Additionally, a well developed indexer greatly increases the speed and efficiency of data retrieval. *Electrs*, a fast and storage efficient open source Rust implementation of the Electrum API was selected as the indexer.

Most critically, *electrs* was developed with privacy in mind and does not communicate with any third party servers. After syncing and indexing with a *Bitcoind* node, *electrs* is able to respond to serve queries of historical transaction data independently. Additionally, *electrs* will periodically poll the node to index new blocks as they are added to the ledger.

4.1.3. *Architecture - web application*

A custom web application was developed to accept user queries, compute address derivations, cache discovered transactions, and query *electrs*. The most important aspect of this service in terms of ensuring forensic validity is the address derivation. Many existing Bitcoin transaction lookup tools either do not support query by extended public key or fail to produce the same wallet addresses as those derived by HD wallets.

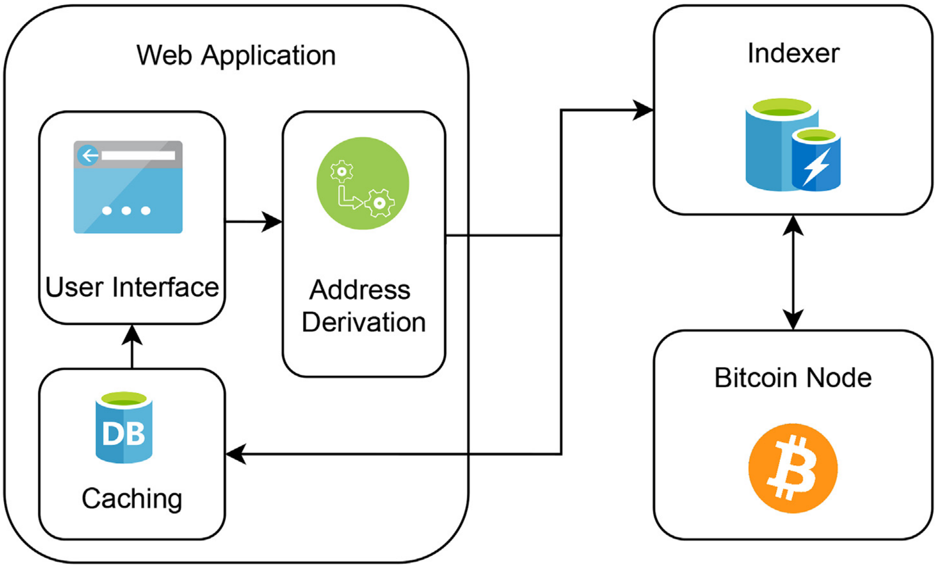


Fig. 1. Microservice architecture.

4.2. Address derivation

Algorithm 1 displays the address derivation technique used in our proof of concept. Two parameters are provided to the address derivation function and it returns the set of addresses derived. The two parameters are: the public key and the desired derivation depth.

HD wallets have an internal and external keychain. Addresses derived along the external keychain are intended for receiving transactions, while those derived along the internal keychain are intended to be used as change addresses or other wallet functions that are not meant to be public.

When sending Bitcoin from an address, all of the Bitcoin owned by that address must be sent. Therefore, if Alice controls 10 Bitcoins in address *x* and she wishes to send 7 Bitcoin to Bob at address *y*, she must send the residual 3 Bitcoins back to an address she controls if she wishes to retain ownership of it. Change addresses are used by Bitcoin wallets to attempt to obfuscate outgoing transactions by generating new addresses on a separate derivation path solely for the purpose of receiving this change. It is computationally infeasible to associate these change addresses to the sender address without access to the extended keys they were both derived from.

Consequently, in order for an examiner to have a complete understanding of how Bitcoin moved in and out of a HD wallet over the course of its transaction history, it is necessary to derive addresses along both the internal and external keychains.

After the set of possible child addresses is generated, *electrs* is queried with each address to retrieve the associated transaction history. This amounts to brute forcing as there is no other method of determining which derivation paths will be used for transactions without guessing and checking them incrementally. It becomes a challenge to know when to stop because the number of possible child addresses on a given derivation path is 231.

BIP 44 defines the “address gap limit” for HD wallets at 20. This means that when wallet software is attempting to find all addresses used by a given wallet, it assumes the in-use addresses have been exhausted after encountering 20 consecutive unused addresses.

While most HD wallet implementations follow this convention, it is theoretically possible for wallet applications to deviate from the standard and still deterministically derive valid addresses. Custom software attempting to perform data hiding may arbitrarily increment by some interval greater than 20 or start derivation at some very large constant. This approach may be employed as a form of anti-forensics by tech-savvy criminals.

Wallet software or query platforms that assume this standard is always followed can not find addresses derived this way, even if in possession of the master public/private key pair. To be absolutely certain that all possible addresses have been found, a query tool must enumerate all 231. Our proof of concept allows the user to specify the desired depth of each query.

Algorithm 1. Address Derivation Algorithm

```
$addresses \leftarrow \{\}$
for $format \in keyFormats$ do
    $key = format.convert(key)$
    if $key.level == ADDRESS$ then
        $addr = key.toAddress()$
        $addresses.append(addr)$
    else
        for $change \in changeCodes$ do
            if $key.level == ACCOUNT$ then
                $key = key.getChangeKey(change)$
            end if
            if $key.level == CHANGE$ then
                $i \leftarrow 0$
                while $i < depth$ do
                    $addr = format.getAddr(key, i)$
                    $addresses.append(addr)$
                    $i \leftarrow i + 1$
                end while
            end if
        end for
    end if
end for
return $addresses$
```

4.3. Evaluation

A variety of publicly available Bitcoin lookup platforms were

Table 2
Comparison of publicly available HD wallet lookup tools.

Name	Open source	Confidential	Automatic conversion	Adjustable depth
BlockQuery	✓	✓	✓	✓
LedgerHQ/xpub-scan	✓	X	✓	✓
dan-da/hd-wallet-addrs	✓	✓	X	X
mewald55/Blockpath	✓	✓	X	X
Blockchain.info	✓	X	X	X
Blockchainexplorer.one	X	X	X	X
Blockonomics.co	X	X	X	X
Blockchair.com	X	X	X	X

surveyed to be assessed against the standards of forensic soundness outlined in this work. Only services which allows users to search for transactions by extended public key were considered.

Bitcoin transactions were made using a Ledger Nano X cryptocurrency hardware wallet with the Ledger Live wallet software. Memory and file system forensics was then performed against the system to obtain forensic artifacts including extended public keys and addresses.

A second HD wallet was generated with python-hdwallet,[5] which allows users to manually set the index of the derivation path. Addresses were generated with address gaps of 100 in attempts to hide addresses from software assuming the default gap limit of 20.

The extended keys were passed into the HD wallet compatible lookup tools listed in Table 2. Each tool was then evaluated on its suitability for forensic use along four criteria. First, whether or not the tool was open source. As established by (Brian Carrier, 2002) and (MansonAnna et al., 2007), digital forensics tools should be open source so that their methods may be verifiable and in certain cases, legal. Second, if the tool queried a third party server and thereby compromised the confidentiality of the investigation. Next, if the tool automatically converted the key to every possible representation to cover the entire address space. Finally, if the tool allowed the user to manually adjust the address gap limit or derivation depth.

5. Findings

5.1. Evaluations results

Table 2 presents the results of the evaluation. Ledger's xpub-scan utility was the only tool capable of finding all of the transactions generated with the extended public keys provided in the query. This is because xpub-scan derives Legacy, Native SegWit, and Nested SegWit addresses regardless of the key format provided in the query.

Additionally, the command line interface allows the user to manually adjust the index range thereby allowing for the discovery of transactions outside the address gap limits defined in the standard. The application is also open source; however, it utilizes Ledger's servers to search the blockchain and while there is an option to use cryptoapis. io as a custom data provider, there is no way to prevent information disclosures.

None of the other tools were able to find any of the transactions made by Ledger Live or addresses with exceptionally high derivation gaps. Notably, Blockpath by GitHub user @mewald55 used an address gap limit of 150 rather than 20, which may aid in the detection of address islands.

5.2. Forensic suitability

It is important to note that none of the assessed tools were developed with forensics in mind. Not only did six out of the seven tools fail to automatically derive SegWit addresses when provided an xpub, several of them were not able to discover SegWit transactions even when provided the correct ypub or zpub. Similarly, only two out of the seven tools evaluated allowed users to query local instances of the blockchain rather than third party services.

The absence of a publicly available tool meeting the standards of forensic suitability illustrates the need for a dedicated open source solution. Our proof of concept successfully discovered all transactions while maintaining the confidentiality of the searches by not calling any third party APIs.

5.3. Tool usage

Figs. 2 and 3 demonstrate the Graphical User Interface (GUI) of BlockQuery. Fig. 2 shows the interface for making a query where the user can manually set the derivation depth and whether to derive along the internal chain, external chain, or both. Users can also force the application to query the indexer again for a cached public key to check for new transactions.

The remaining views render a relational database model allowing users to navigate along the relationships between keys, addresses, and transactions. Fig. 3 shows the view presented to the user when inspecting an individual transaction.

6. Discussion

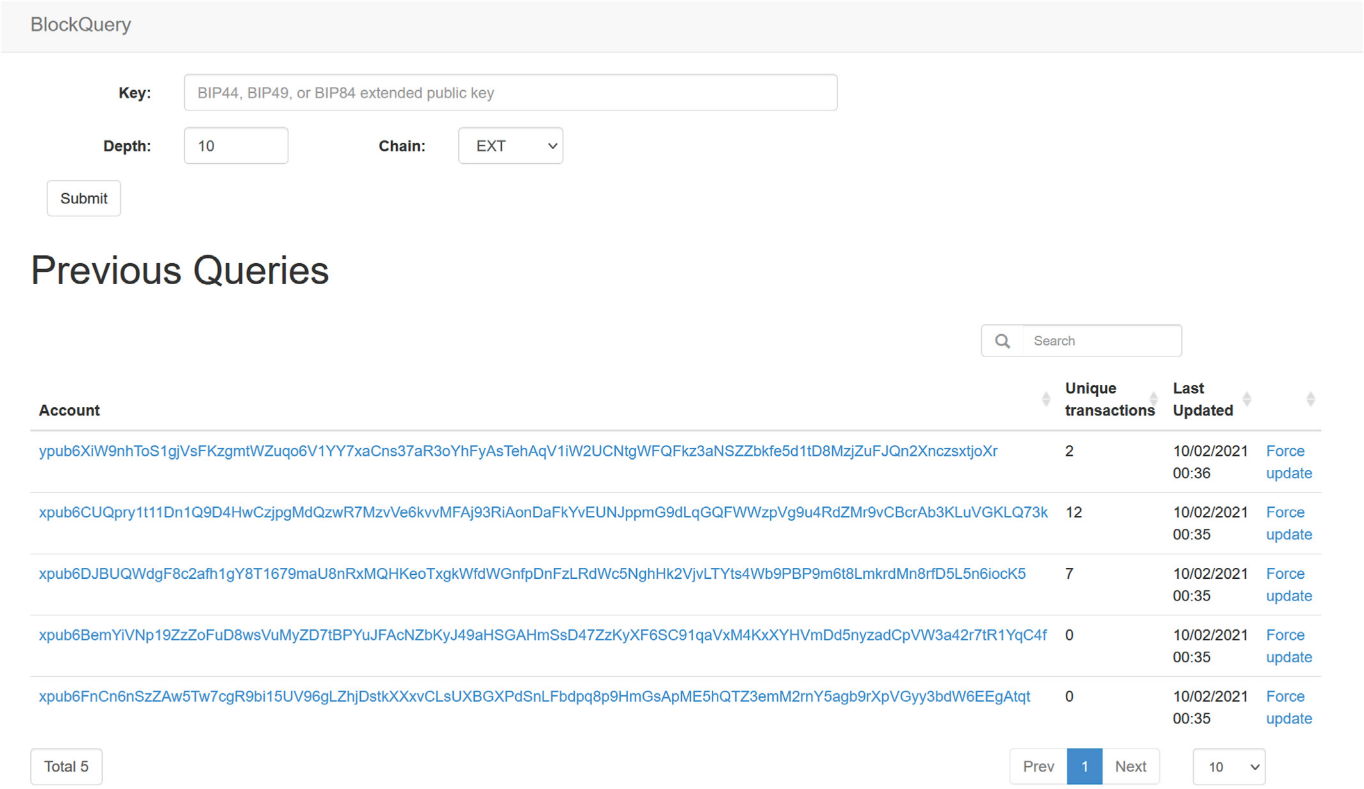
6.1. Limitations

Running a robust forensically sound cryptocurrency query platform requires a significant amount of resources. At the time of writing, the Bitcoin blockchain was approximately 350 GB and the Ethereum blockchain was approaching a terabyte. Similarly, computing all 231 possible addresses for a given extended key would require a machine with significant parallel computing power. A local law enforcement agency with limited resources may find this to be a prohibitive barrier.

In this case, trusted entities, such as state and federal law enforcement agencies or universities, may provide hosting of the service. This way, forensic investigators that lack the technical knowledge or resources to run their own node can query a third party without compromising the confidentiality of their investigation.

The microservice based architecture lends itself well to such a model as individual components can be run by different trusted entities.

[5] https://github.com/meherett/python-hdwallet.



BlockQuery

Key: BIP44, BIP49, or BIP84 extended public key

Depth: 10

Chain: EXT

Submit

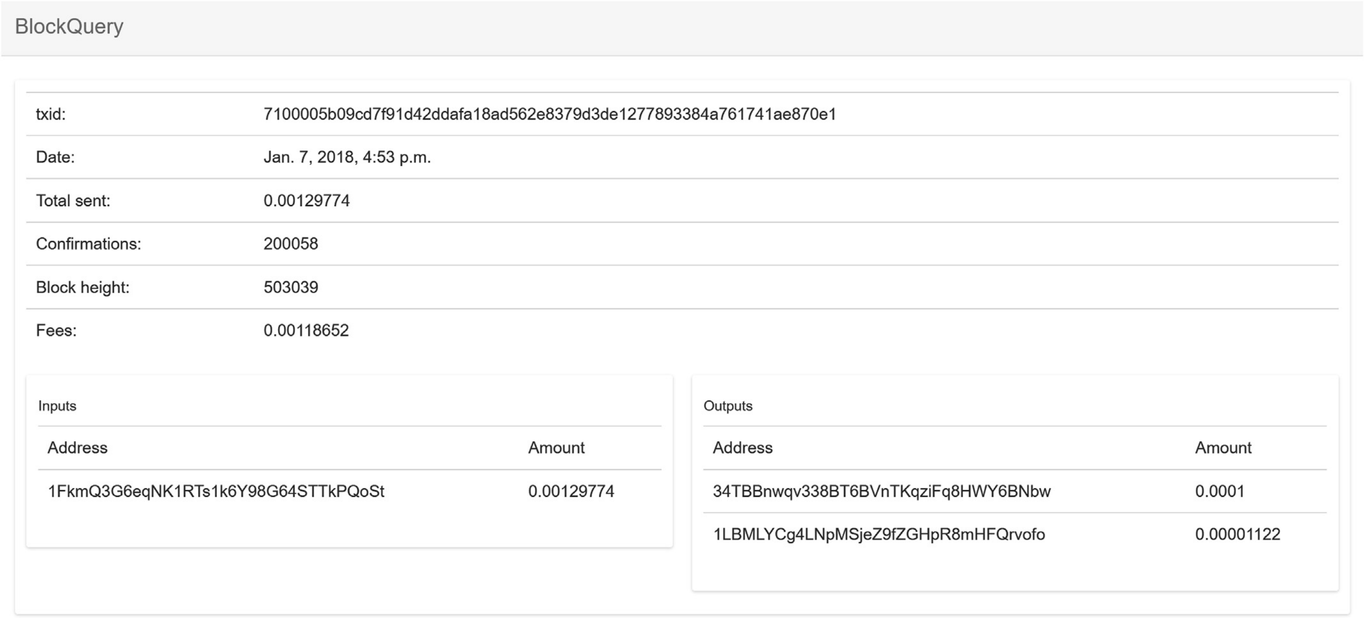
Previous Queries

Search

Account	Unique transactions	Last Updated	
ypub6XiW9nhToS1gjVsFKzgmtWZuqo6V1YY7xaCns37aR3oYhFyAsTehAqV1iW2UCNtgWFQFkz3aNSZZbkfe5d1tD8MzjZuFJQn2XnczsxtjoXr	2	10/02/2021 00:36	Force update
xpub6CUQpry1t11Dn1Q9D4HwCzjpgMdQzwR7MzvVe6kvvMFAj93RiAonDaFkYvEUNJppmG9dLqGQFWWzpVg9u4RdZMr9vCBcrAb3KLuVGKLQ73k	12	10/02/2021 00:35	Force update
xpub6DJBUQWdgF8c2afh1gY8T1679maU8nRxMQHKeoTxgkWfdWGnfpDnFzLRdWc5NghHk2VjvLTYts4Wb9PBP9m6t8LmkrdMn8rfD5L5n6iocK5	7	10/02/2021 00:35	Force update
xpub6BemYiVNp19ZzZoFuD8wsVuMyZD7tBPYuJFAcNZbKyJ49aHSGAHmSsD47ZzKyXF6SC91qaVxM4KxXYHVmDd5nyzadCpVW3a42r7tR1YqC4f	0	10/02/2021 00:35	Force update
xpub6FnCn6nSzZAw5Tw7cgR9bi15UV96gLZhjDstkXXxvCLsUXBGXPdSnLFbdpq8p9HmGsApME5hQTZ3emM2rnY5agb9rXpVGyy3bdW6EEgAtqt	0	10/02/2021 00:35	Force update

Total 5

Fig. 2. BlockQuery - Query view example.



BlockQuery

txid:	7100005b09cd7f91d42ddafa18ad562e8379d3de1277893384a761741ae870e1
Date:	Jan. 7, 2018, 4:53 p.m.
Total sent:	0.00129774
Confirmations:	200058
Block height:	503039
Fees:	0.00118652

Inputs

Address	Amount
1FkmQ3G6eqNK1RTs1k6Y98G64STTkPQoSt	0.00129774

Outputs

Address	Amount
34TBBnwqv338BT6BVnTKqziFq8HWY6BNbw	0.0001
1LBMLYCg4LNpMSjeZ9fZGHpR8mHFQrvofo	0.00001122

Fig. 3. BlockQuery - Transaction view example.

6.2. *Technical abstraction*

An important consideration when designing and developing digital forensics tools is the technical barrier a user must overcome to effectively use the tool. Not every investigator will be expected to understand the complexities of Bitcoin address derivation. Consequently, digital forensic toolkits should take into account common edge cases and liberally interpret the intentions of the user to ensure that all possible results are accessible.

It was this design mindset that motivated the development of BlockQuery. It is possible to obtain the same data with an extended public key by manually converting and deriving all possible address representations. However, this requires an in depth understanding of esoteric derivation standards and may be abstracted from the user. An effective solution would allow an investigator to simply enter an artifact they obtained from forensic analysis and receive the complete transaction history associated with that public key.

7. Future work

7.1. Extended key dataset

The lack of an open source extended key dataset was noted during the development of BlockQuery. Such a dataset would greatly facilitate the development of more forensic tools leveraging extended keys by streamlining testing and evaluation.

Likewise, an extended public key dataset could be used to conduct an analysis of the slight deviations from the standard in different HD wallet clients, similar to the idiosyncrasies of key representations in Ledger Live noted in this work.

7.2. Support for additional currencies

Many other cryptocurrencies implement address derivation schemes compatible with HD wallets. With this in mind, the formula outlined here can be applied to any such cryptocurrency. A cursory analysis conducted during the evaluation phase proved that there are very few services which allow users to search the blockchain by extended key for any currency other than Bitcoin.

7.3. Plugin integration

To further streamline the investigative process, BlockQuery may be integrated into existing plugins for retrieving forensic artifacts. For example, a Volatility plugin that pulls extended public keys from system memory images can call BlockQuery's API to automatically generate a detailed transaction history report. Similar plugins can be developed for file system forensics tools such as Autopsy.

8. Conclusion

Extended public keys are extremely valuable when conducting a forensic investigation. However, the improper use of tools not intended for forensic use may not yield a complete transaction history. Additionally, these tools may compromise the integrity of a forensic investigation by utilizing third party servers to query the blockchain for transactions or addresses specific to an ongoing investigation.

Our proof of concept, BlockQuery, attempts to address these issues and demonstrates a software architecture model for implementing a local extended key query service.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant Number 1921813 Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Al-Khateeb, Haider, Epiphaniou, Gregory, Daly, Herbert, 2019. Blockchain for modern digital forensics: the chain-of-custody as a distributed ledger. In: Blockchain and Clinical Trial. Springer, pp. 149–168.

Baumann, Annika, Fabian, Benjamin, Lischke, Matthias, 2014. Exploring the Bitcoin Network. WEBIST.

Brian Carrier, 2002. Open Source Digital Forensics Tools: the Legal Argument. Technical report, Citeseer.

Cebe, Mumin, Erdin, Enes, Akkaya, Kemal, Aksu, Hidayet, Uluagac, Selcuk, 2018. Block4forensic: an integrated lightweight blockchain framework for forensics applications of connected vehicles. IEEE Commun. Mag. 56 (10), 50–57.

Fleder, Michael, Michael, S., 2015. Kester, and Sudeep Pillai. Bitcoin transaction graph analysis.

Koshy, Philips, Koshy, Diana, Mcdaniel, Patrick, 2014. An analysis of anonymity in bitcoin using p2p network traffic. In: Financial Cryptography.

Li, Shancang, Qin, Tao, Min, Geyong, 2019. Blockchain-based digital forensics investigation framework in the internet of things and social systems. In: IEEE Transactions on Computational Social Systems, 6, pp. 1433–1441, 6.

Manson, Dan, Anna, Carlin, Steve Ramos, Gyger, Alain, Kaufman, Matthew, Treichelt, Jeremy, 2007. Is the open way a better way? digital forensics using open source tools. In: 2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07). IEEE, 266b–266b.

Meiklejohn, Sarah, Pomarole, Marjori, Jordan, Grant, Levchenko, Kirill, McCoy, Damon, Voelker, Geoffrey M., Savage, Stefan, 2013. A fistful of bitcoins: characterizing payments among men with no names. In: *Proceedings Of the 2013 Conference On Internet Measurement Conference*, IMC '13. Association for Computing Machinery, New York, NY, USA, ISBN 9781450319539, pp. 127–140. https://doi.org/10.1145/2504730.2504747.

Reid, Fergal, Harrigan, Martin, 2012. An Analysis of Anonymity in the Bitcoin System.

Ricci, Joseph, Baggili, Ibrahim, Frank, Breitinger, 2019. Blockchain-based distributed cloud storage digital forensics: where's the beef?. In: IEEE Security Privacy, 17, pp. 34–42. https://doi.org/10.1109/MSEC.2018.2875877, 1.

Ryu, Jung Hyun, Sharma, Pradip Kumar, Jo, Jeong Hoon, Park, Jong Hyuk, 2019. A blockchain-based decentralized efficient investigation framework for iot digital forensics. J. Supercomput. 75 (8), 4372–4387.

Schwittmann, Lorenz, Wander, Matthäus, Boelmann, Christopher, Weis, Torben, 2014. Privacy preservation in decentralized online social networks. In: IEEE Internet Computing, 18, pp. 16–23. https://doi.org/10.1109/MIC.2013.131, 2.

Siddula, Madhuri, Li, Lijie, Li, Yingshu, 2018. An empirical study on the privacy preservation of online social networks. In: IEEE Access, 6, pp. 19912–19922. https://doi.org/10.1109/ACCESS.2018.2822693.

Tyler, Thomas, Piscitelli, Mathew, Shavrov, Ilya, Baggili, Ibrahim, 2020. Memory foreshadow: memory forensics of hardware cryptocurrency wallets–a tool and visualization framework. Forensic Sci. Int.: Digit. Invest. 33, 301002.

Zhang, Xiaolu, Grannis, Justin, Baggili, Ibrahim, Lang Beebe, Nicole, 2019. Frameup: an incriminatory attack on storj: a peer to peer blockchain enabled distributed storage system. Digit. Invest. 29, 28–42.