TECH THROUGH
THE LENS OF
SECURITY

BlockQuery: Toward Forensically Sound Cryptocurrency Investigation

Tyler Thomas
Tiffanie Edwards
Ibrahim Baggili

University *of* New Haven
CONNECTICUT INSTITUTE OF TECHNOLOGY
UNHcFREG

University of New Haven

AUTHOR INFORMATION

Tyler Thomas

Tyler is a former graduate of the University of New Haven ad now holds a Master's degree in Cybersecurity & Networks. He focused on research in memory forensics.

Tiffanie Edwards

Tiffanie is a graduate student in Cybersecurity & Networks. She received her Bachelor's degree from Southern Connecticut University in Computer Science.

Ibrahim Baggili

Dr. Baggili is the founding director of the Connecticut Institute of Technology and the Elder Family Endowed Chair of Computer Science & Cybersecurity at the University of New Haven. Additionally, he leads the research team (UNHcFREG) on campus.

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.

Outline

- Introduction
- Motivation & Contributions
- Background
- Methodology
- Evaluation Metric & Findings
- Limitations & Discussion
- Future Work

Introduction

Forensic soundness of a cryptocurrency investigation:

- **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

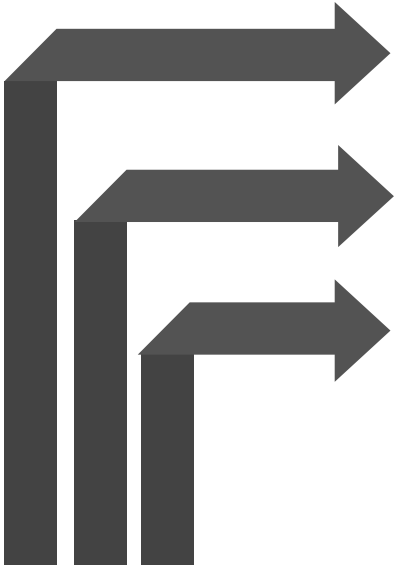
Introduction cont.

- Third-party blockchain indexer
- Integrity of query responses
- Address derivation schemes

Motivation

A forensically sound cryptocurrency lookup platform must consist of a trusted full node running directly on the blockchain network.

Contributions



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.

Background

Hierarchical Deterministic Wallets

- use extended keys to compartmentalize addresses under logical “accounts”
- Each account has an associated key pair, and accounts are organized hierarchically.
- At the lowest level in the hierarchy, the keys are used to deterministically derive ephemeral wallet addresses

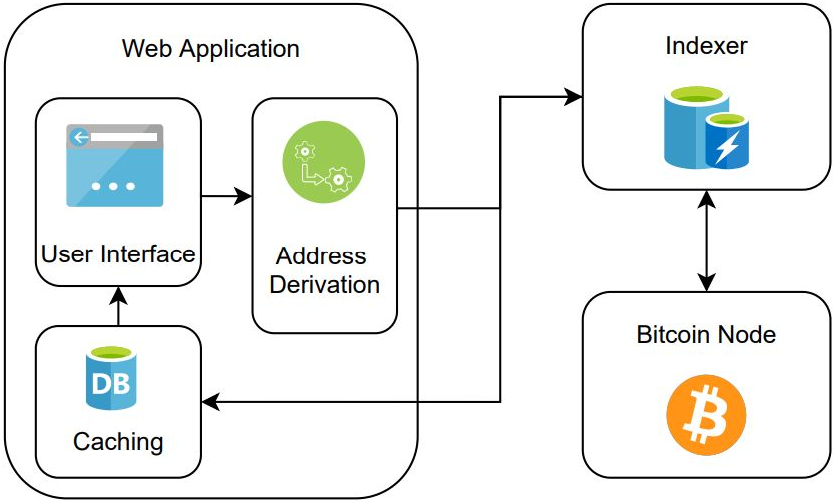
Bitcoin Address Types

- Three valid Bitcoin address representations.
- Each address type has a respective extended key representation used to derive addresses of that type
- Memory forensic analysis of applications using all three address types show in some cases it is only possible to recover one representation of the public key

Address Representations

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

Table 1: Bitcoin address representations



Methodology

- **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.

Algorithm 1 Address Derivation Algorithm

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

Methodology cont.

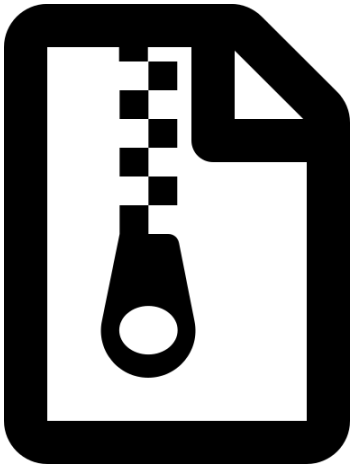
- Public key and desired deprivation depth are provided as parameters to the address deprivation function
- Set of derived addresses are returned
- Change addresses are used by Bitcoin wallets to attempt to obfuscate outgoing transactions by generating new addresses on a separate derivation path

Algorithm 1 Address Derivation Algorithm

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

Methodology cont.

- After the set of possible child addresses is generated, electrs is queried with each address to retrieve the associated transaction history Brute forcing 2^{31} possible child addresses on a derivation path
- Necessary to brute force all possible child addresses
- Our tool allows the desired depth of each query to be specified by the user



Evaluation & Findings

Evaluation Criteria

- Surveyed publicly available Bitcoin lookup platforms
  - Only considered services that allow users to search for transactions by extended public key
- Memory and filesystem forensics was performed against each 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

Evaluation Criteria

1. Whether or not tool was open-source
2. If the tool queried a third-party server and thereby compromised the confidentiality of the investigation
3. If the tool automatically converted the key to every possible representation to cover the entire address space.
4. If the tool allowed the user to manually adjust the address gap limit or derivation depth.

Findings

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

Limitations

- Computing all 231 possible addresses for a given extended key would require a machine with significant parallel computing power
  - Outsourcing this responsibility to larger agencies or trusted univiersities with the available resources is one way to overcome this

Future Work

- Extended key dataset would facilitate the development of more forensic tools leveraging extended keys by streamlining testing and evaluation
- Application of BlockQuery to other cryptocurrencies that implement address derivation schemes compatible with HD wallets
- Plugin integration of BlockQuery for certain software

Contacts

Tyler Thomas
tthom10@unh.newhaven.edu

Tiffanie Edwards
Tedwa4@unh.newhaven.edu

Ibrahim Baggili, Ph.D
Ibaggili@newhaven.edu