Bitcoin Address Clustering and Anticlustering

Ferran Bernabé Puigmarti

02/07/2023

Abstract– En l'ecosistema de Bitcoin, hi ha dos aspectes fonamentals que expliquen el seu èxit, que són la privacitat i la seguretat. D'aquesta manera, teòricament, aquests aspectes permeten als usuaris disposar d'una confidencialitat financera, en mantenir les transaccions i saldos en l'anonimat, d'una protecció de la identitat i d'una prevenció d'activitats fraudulentes, ja que no es poden rastrejar les transaccions d'un usuari. A la pràctica, la naturalesa pública de la cadena de blocs, permet el seguiment de transaccions i l'associació de direccions de Bitcoin amb identitats reals. En aquest estudi, s'analitzaran les tècniques per a poder realitzar un seguiment de les transaccions d'un usuari, conegudes com a tècniques d'Address Clustering o heurístiques d'Address Clustering i mecanismes per a transformar la transacció i inutilitzar les tècniques d'Address Clustering, coneguts com a mecanismes Anticlustering o antiheurístiques d'Address Clustering. A més a més, es programaran les heurístiques i antiheurístiques en Python, comprovant l'efectivitat de les heurístiques en bases de dades de transaccions i calculant el cost afegit per a defensar les transaccions utilitzant antiheurístiques.

Paraules clau– Bitcoin, Blockchain, UTXO, Anonimat, Pseudoanonimat, Unlinkability, Heurístiques, Antiheurístiques, Clustering, Fees.

Abstract– In the Bitcoin ecosystem, there are two fundamental aspects that explain its success, which are the privacy and the security. This way, theoretically, this aspect allows users to have a financial confidentiality, to keep the transactions and the balances anonymously, to have an identity protection and a fraudulent activity prevention, because the user's transactions can't be tracked. In the practice, the public nature of the Blockchain, allow the tracking of transactions and the associating of Bitcoin addresses and real identity. In this study, I'll analyze the techniques to do a tracking of a user's transactions, known as Address Clustering techniques or Address Clustering Heuristics, and mechanisms to transform the transactions that disable the Address Clustering techniques, known as Anticlustering mechanisms or Address Clustering Antiheuristics. Furthermore, I'll program the heuristics and antiheuristics in Python, verifying the heuristics effectiveness in transaction databases, calculating the added cost in order to defend the transactions using antiheuristics.

Keywords– Bitcoin, Blockchain, UTXO, Anonymity, Pseudoanonymity, Unlinkability, Heuristics, Antiheuristics, Clustering, Fees.

1 Introducció - Context del treball

BITCOIN és una infraestructura que ha permès la creació d'una criptomoneda amb un control descentralitzat, és a dir que no hi ha una entitat central que tingui el control de la infraestructura en qüestió. Aquesta descentralització s'aconsegueix a través de la cadena de blocs de Bitcoin, que actua com una base de dades distribuïda on s'emmagatzema tota la informació necessària per a fer funcionar el protocol [1]. La informació mínima que s'inclou en la cadena són els blocs, els quals contenen

• E-mail de contacte: ferribernabepuigmarti@gmail.com
• Mención realitzada: Enginyeria de Tecnologies de la Informació
• Treball tutoritzat per: Jordi Herrera Joancomartí (Enginyeria de la Informació i les Comunicacions)
• Curs 2022/23
• Link github: https://github.com/Ferri-bernabe/TFG

transaccions, i dintre de les transaccions trobem les adreces, juntament amb més camps, que permeten fer pagaments entre usuaris. Aquesta informació de la cadena de blocs és totalment pública i qualsevol pot accedir-hi utilitzant qualsevol explorador de blocs de Bitcoin o mitjançant un node de Bitcoin, però aquesta forma és més tediosa i costosa per un usuari comú.

Com s'ha esmentat anteriorment, la xarxa de Bitcoin permet fer pagaments entre usuaris, però té una característica pròpia, que no utilitza el model bancari tradicional on un usuari té un compte amb una quantitat de diners, sinó que utilitza el model UTXO (Unspent Transaction Output). Aquest model es basa en el fet que per a poder fer un pagament s'han de gastar els outputs, que són la part d'una transacció on s'especifica la quantitat de Bitcoins i els usuaris a qui van dirigits, els quals venen amb una condició per a poder ser gastats. A l'hora de fer un pagament en forma de transacció, la xarxa de Bitcoin comprova que en el pagament s'està fent servir una UTXO vàlida, és a dir, que no s'ha gastat anteriorment i que l'usuari qui vol fer el pagament és el propietari de l'output. Per a demostrar que l'usuari és el propietari, hi ha diverses maneres de fer-ho, però la manera general és demostrant que aquest usuari coneix la clau privada a la qual li correspon el hash de la clau pública. D'aquesta manera, el saldo total que té una persona és el valor total del conjunt d'UTXO de les quals pot verificar que és el propietari. Una persona pot tenir diverses adreces de Bitcoin amb diferents saldos associats, però, tot i que aquestes adreces són visibles a través de la cadena de blocs, públicament les propietàries d'aquestes no són conegudes de manera simple, fet pel qual es denomina que Bitcoin és pseudoanònim [2].

Perquè un sistema sigui reconegut com a anònim complet, ha de complir les propietats de pseudoanonimat, és a dir, que una interacció d'un usuari amb el sistema no es pugui atribuir a la persona real, i no-enllaçament, és a dir, que no es pugui saber que dues interaccions independents del mateix usuari siguin del mateix usuari. Aquesta propietat de no-enllaçament és extremadament important perquè hi ha serveis de Bitcoin que demanen identificació real de l'usuari, i si es poden enllaçar les transaccions d'aquest usuari, es podria saber la quantitat de Bitcoins i com mou aquesta quantitat. Aquesta propietat de no-enllaçament sembla contrària a un sistema amb un identificador únic públic per usuari, donat que en observar aquest identificador en el sistema, es pot atribuir la interacció que ha realitzat amb l'usuari en qüestió. Per exemple, fòrums com Reddit, utilitzen un identificador per usuari, de manera que cada interacció d'aquest amb el sistema serà reconeguda pels observadors, de manera que no compleix la propietat de no-enllaçament. En canvi, altres fòrums com 4chan no demanen un identificador a l'usuari, fent que a priori, sembli que dues interaccions d'un mateix usuari no seran enllaçables entre elles [3]. En el cas de Bitcoin sembla evident que utilitzar una única adreça com a identificador de l'usuari no complirà amb la propietat de no-enllaçament i, per tant, les transaccions que fes l'usuari serien conegudes pels observadors, ja que aquestes transaccions acaben a la cadena de blocs, que és informació pública. A causa d'això, una de les pràctiques més esteses i que el mateix protocol recomana, és la generació contínua d'adreces per rebre i pagar altres usuaris a la cadena de blocs, i per tant evitar reutilitzar adreces. Així i tot, a causa de disposar dels historials de les transaccions que els mateixos usuaris produeixen per poder interaccionar amb el protocol, és possible en certs casos recuperar informació sobre certes adreces, podent agrupar-les en clústers (agrupacions), formant grups d'adreces controlades pel mateix usuari.

En aquest treball, faré una recerca sobre les tècniques i heurístiques principals que s'utilitzen per obtenir informació sobre els propietaris de les adreces, i també sobre quines possibles solucions/tècniques es poden aplicar per intentar evitar que es puguin identificar les adreces. Més enllà de l'estudi tèoric, el treball inclourà una part pràctica programant les heurístiques sobre unes transaccions sintètiques i el disseny de mecanismes anticlustering per inutilitzar aquestes heurístiques.

2 ESTAT DE L'ART

L'anàlisi de mètodes de clustering i anticlustering en el camp de la seguretat de Bitcoin ha estat un tema d'investigació recurrent en els últims anys. En relació als mètodes de clustering, hi ha una àmplia varietat de tècniques utilitzades per fer un seguiment de les transaccions d'un usuari. Durant la meva revisió de l'estat de l'art, he trobat articles científics de gran importància que aborden aquest tema, com ara "Resurrecting Address Clustering in Bitcoin"[4], "Bitcoin address clustering method based on multiple heuristic conditions"[5] i "Heuristic-Based Address Clustering in Bitcoin"[1]. També he pogut observar que la pàgina "bitcoin.it – Wiki"recull la majoria de les tècniques de clustering estudiades a l'apartat de "Privacy"[6]. En aquesta última pàgina, també es recullen tècniques de clustering a través de comportaments inusuals de la xarxa de Bitcoin, però en aquest estudi no s'analitzaran els mètodes de clustering en l'àmbit de xarxa i únicament s'estudiarà l'anàlisis de les transaccions mitjançant les heurístiques. Les heurístiques utilitzades en aquests mètodes de clustering es basen principalment en els comportaments dels moneders de Bitcoin en quant a la generació de les transaccions. Aprofitant aquestes empremtes que es deixen en les transaccions, és possible identificar l'adreça de canvi, amb una gran efectivitat, i, per tant, establir una relació amb l'usuari en qüestió. L'objectiu del meu estudi és recopilar aquestes heurístiques ja conegudes, programar-les i provar-les en bases de dades de transaccions per aprofundir en la comprensió d'aquest àmbit. En quant a l'anticlustering, l'enfocament més habitual és l'ús de tècniques de "mixing"o realitzar pagaments a diverses adreces [7]. En el meu treball, proposo desenvolupar mecanismes anticlustering específics per a cada una de les heurístiques de clustering. Això implica la creació d'una antiheurística per a cada heurística existent. Amb aquest enfocament, pretenc modificar les transaccions per a inutilitzar les heurístiques estudiades. En modificar la transacció d'aquesta manera, s'eliminarà la traça natural que deixen els diferents moneders i, per tant, es dificultarà la identificació de l'adreça de canvi. Això suposarà un desafiament per a qualsevol intent de vincular aquesta adreça amb l'usuari o fins i tot pot donar lloc a la vinculació amb una adreça incorrecta.

3 OBJECTIUS

L'objectiu principal del TFG és analitzar l'anonimat de la Blockchain, analitzant la propietat de no-enllaçament comentada anteriorment. D'aquesta manera, mitjançant les heurístiques mencionades, s'analitzaran quines funcionen millor i amb quines es pot rastrejar les transaccions que realitza un usuari, creant així clústers per a tots els usuaris amb les adreces que corresponen a ells mateixos. Per a realitzar l'objectiu principal, s'han definit uns quants subobjectius:

1. Implementar els mecanismes de clustering (heurístiques).
2. Provar els mecanismes de clustering en dades sintètiques.
3. Provar els mecanismes de clustering en el groundtruth proporcionat en l'estat de l'art [4].
4. Observar les diferències de resultats en les diferents dades.
5. Dissenyar i implementar mecanismes anticlustering.
6. Calcular el cost de la privacitat dels mecanismes anticlustering.

Com es pot observar, a partir de la programació dels mecanismes de clustering, es provaran en dades sintètiques i en el groundtruth proporcionat [4], tot observant les diferències i el perquè d'aquestes. A continuació, un cop comprovades les heurístiques en els dos grups de dades, es crearan mecanismes anticlustering per neutralitzar les heurístiques. Aquests mecanismes augmentaran el cost de les transaccions en fer-les més grans, motiu pel qual es calcularà la diferència de cost, per a saber quant de costós és obtenir la privacitat.

4 METODOLOGIA I DESENVOLUPAMENT

Pel que fa a la metodologia que s'ha fet servir per a programar tant les heurístiques com els mecanismes anticlustering, ha estat el mètode de cascada [8]. És una de les metodologies més simples i que funciona bé per a projectes senzills, sense necessitat de fer iteracions ni dissenys innecessaris que comportarien més feina però no un millor resultat per al meu cas. Entrant en detalls, per a fer la programació de les heurístiques i dels mecanismes anticlustering s'ha fet servir el mateix sistema de treball. Un cop recollit els requisits, s'ha fet un disseny en forma de casos d'ús i la consegüent implementació de cadascuna de les heurístiques, comprovant que cadascuna d'aquestes funcioni correctament. Posteriorment, s'ha fet una fase de manteniment on s'han realitzat millores, correccions d'errors i optimitzacions del codi programat.

Pel que fa al procés per a poder complir amb tots els objectius es divideix en dues parts ben diferenciades. La primera part consisteix en la implementació d'heurístiques i la segona part en la implementació d'antiheurístiques. Pel que fa a la primera part, primer s'ha fet una recerca de les heurístiques estudiades per a poder detectar l'adreça de retorn de les transaccions. Un cop estudiades les principals heurístiques, s'han programat i s'han provat en bases de dades de transaccions sintètiques, és a dir, generades pel tutor, i en la base de dades pública mencionada anteriorment [4]. Respecte a la segona part, primer s'ha fet un estudi de quan les heurístiques fallen o no troben l'adreça de retorn i s'ha implementat una funció que neutralitzi cadascuna de les heurístiques de forma individual. Per acabar amb aquesta segona part, s'han provat les antiheurístiques en un entorn de dades sintètiques, generades amb "Bitcoin Core"[9].

5 IMPLEMENTACIÓ DE LES HEURÍSTIQUES

En la recerca d'heurístiques[10][11], es va determinar que per les bases de dades que teníem disponibles, les que es podien desenvolupar i provar són les següents:

1. Múltiples entrades per usuari

   Donada una adreça d'un usuari, busca aquesta adreça en tots els inputs i recopila les adreces amb les quals comparteix input (i que, per tant, són del mateix usuari). També es busquen les adreces que comparteixen inputs amb les noves adreces trobades, recorrent les transaccions fins que no hi hagi més adreces associades. D'aquesta heurística, que és l'única que no consisteix a trobar l'adreça de retorn, sorgeix Múltiples entrades per tothom, que aplica el Múltiples entrades per usuari per a cada adreça no associada a un clúster. Un exemple de com aquesta heurística generaria un clúster d'inputs que hagin estat utilitzats en les mateixes transaccions seria la següent: imaginem que volem trobar un clúster per a l'adreça "1EhQpkTWB94zFK9c8xsk8JMbJ8a74SdDsz". L'heurística buscaria aquesta adreça en totes les transaccions de la Blockchain i quan la detectés, generaria un clúster amb les adreces que s'hagin conjuntat amb aquesta. Per exemple, l'heurística trobaria que aquesta adreça s'ha utilitzat en la transacció *818a5877a...c5de354*, trobada utilitzant l'explorador de Blockchain.com [12]:

1EhQpkTWB94zFK9c8xsk8JMbJ8a74SdDsz 0.00110288 BTC	399sXgidemyjCPWHM4AtKWbrHywGZ2GRzF 0.03675769 BTC
1E7ehYMrzbuhyRWMooD5X4yYQEXbkxWpDw 0.00492758 BTC	1BhVMg1dZ5QDvNwaucQmbaQZ5JS4KaFiKK 0.00029239 BTC
18QexQsrr5PGz4k6NtFyMnXHkieP8Jk8wR 0.03120162 BTC	

   D'aquesta manera, el clúster temporal quedaria de la següent manera:

   {1EhQpkTWB94zFK9c8xsk8JMbJ8a74SdDsz,
   1E7ehYMrzbuhyRWMooD5X4yYQEXbkxWpDw,
   18QexQsrr5PGz4k6NtFyMnXHkieP8Jk8wR}

   Suposant que l'adreça 1EhQpkTWB94zFK9c8xsk8JMbJ8a74SdDsz, no s'utilitza en cap altra transacció, ara s'aplicaria el mateix mecanisme per les dues adreçes restants del clúster. Aquest cicle es repeteix fins que totes les adreçes del clúster han estat analitzades i ja no hi ha

més adreces conjuntades amb qualsevol del clúster en la Blockchain.

2. Tipus d'adreça per Transacció

   Donada una transacció, retorna l'adreça de canvi segons el tipus d'adreça. D'aquesta manera, es comprova que totes les adreces dels inputs siguin del mateix tipus, i que només hi hagi un output amb aquest tipus d'adreça, resultant així l'adreça de canvi. Cal destacar que Bitcoin disposa de 4 tipus d'adreces principals[13]:

   - Legacy (P2PKH) - comencen amb un 1
   - Nested SegWit (P2SH) - comencen amb un 3
   - Native SegWit (bech32) - comencen amb bc1q
   - Taproot (P2TR) - comencen amb bc1p

   Un exemple de l'adreça de canvi que detectaria aquesta heurística, la podem trobar en la transacció *4af2eb9f...f62ac00*:

bc1pq3h3mwygharvgm6f30dj98lmueg085frwkw6mvgr0qmven20cf9suz6g8t 0.01726280 BTC	bc1px3pv8jz2p8lgmmfrk9y6qdgpuc8ayus753telany2kzez3l7zdysyts00u 0.01171683 BTC
	1CVMDiKoGSLua8CCqAn2FHta41po9sqv7p 0.00551332 BTC

   En aquest cas, podem observer que l'adreça de l'input és del tipus Taproot i trobem dues adreces de diferent tipus en els outputs, una de tipus Legacy i una del tipus Taproot. Com que generalment els wallets utilitzen només 1 tipus d'adreça, aquesta heurística ens diu que les adreces Taproot d'aquesta transacció pertanyen a l'usuari i que per tant, l'output 0 és l'adreça de retorn i l'output 1 és l'adreça de pagament.

3. Detecció utilitzant decimals

   Donada una transacció, retorna l'adreça de canvi segons els decimals que tinguin els outputs. Partint del fet que els humans utilitzem números amb pocs decimals, si només un dels outputs té menys de dos decimals, aquest serà l'output de pagament, fent que l'adreça de canvi sigui l'única amb més de dos decimals. Un exemple de l'adreça de canvi que detectaria aquesta heurística, la podem trobar en la transacció *0f0a0b68...8cb019b478a*:

1BGMoN5Kz7amtyNshRomWT4ihhsYYQvRR3 0.01469490 BTC	bc1qdzsutzcnptsqvxgnecrvewgudxrpj8z909vnpc3efekh6sq6ga5q6c29ha 0.01000000 BTC
	18uhzy546Qz7CxRNkHohg4W9VSkfTkbSvY 0.00428230 BTC

   Com es pot observar, l'output 0 té dos decimals (0,01), mentre que l'output 1 en té vuit. Aquest exemple podria ser el típic d'una botiga que posa el preu directament en BTC i no en una moneda centralitzada, fent que l'heurística detecti l'adreça de pagament com la que té menys de dos decimals, i l'adreça de canvi, la que en té més, en aquest cas, l'output 1.

4. Pagament de quantitat exacta

   Donada una transacció, si només hi ha un output, aquest output serà inclòs al clúster de l'usuari, ja que hi ha una baixa probabilitat que pel pagament que es vol efectuar, hi hagi algun/s inputs que siguin exactes i que, per tant, no es necessiti una adreça de retorn. Aquest és el típic moviment que realitza un usuari al canviar de wallet. Un exemple d'aquest tipus de transaccions és la transacció *a9ee22...0208*:

1Pbf6vv7WLCUEbFypcn3AAsDTdFEWx7TKr 0.00200000 BTC	3KbGRaaWkYfsVZPsPHe1FMceskX4ViUdKZ 0.00160000 BTC

   Com que només hi ha un output, l'heurística detectaria que aquesta adreça de l'output 0 pertany al mateix usuari.

5. Reutilització d'adreça

   Donada una transacció, observa quins dels outputs han estat reutilitzats en transaccions anteriors. D'aquesta manera, l'output que no hagi estat utilitzat en una transacció prèvia serà l'adreça de canvi, ja que els wallets normalment generen noves adreces de canvi per a cada pagament nou que s'ha d'efectuar. Posem com a exemple la transacció *33efd7...45f0*:

bc1pg429pjck4t0dduknf5elhfv8tunlpzeg4svuk8zs0003e9579w5s7mahhm 0.00079535 BTC	bc1prhyalk738z43jwvq6fte66yjql6rxns976qc5mtrmrp23ejmd5msrc65y4 0.00044479 BTC
	bc1pg429pjck4t0dduknf5elhfv8tunlpzeg4svuk8zs0003e9579w5s7mahhm 0.00010000 BTC
	bc1p5e5j2zkhzdac3mz4p4zxmrf5ujcl5ycxwf6vxcn5d85gt6g8g0jq86x3fp 0.00001000 BTC

   Si els outputs 1 i 2 ja s'han utilitzat anteriorment en la cadena de blocs, aleshores, l'heurística, detectarà l'adreça de canvi com l'output 0, ja que serà la nova adreça generada pel wallet.

6. Canvi òptim

   Donada una transacció, comprova que hi hagi dos o més inputs i que hi hagi un output que sigui més petit que algun dels inputs. Si es compleixen les condicions, l'output que és més petit que algun dels inputs, serà l'output de canvi. Aquest fet és potenciat pel comportament dels wallets d'optimitzar al mínim el nombre d'inputs, abaratint la transacció. Per tant, si hi ha un output que és més petit que algun input, en cas que el pagament fos per aquell import, no caldrien més inputs per a realitzar el pagament. Un exemple de l'adreça de canvi que detectaria aquesta heurística, la podem trobar en la transacció *584c48...c47fbb*:

bc1qnm90x3wutuml58mz akned2ezur5qluwcqmgrrf 0.01304882 BTC	bc1q6dlg2sd5tcn653kqn md403h7mvvath78g90yw 8 0.00008000 BTC
bc1q96vxunrjdvxlj2ygkm a60dfu5r8h8uv83f0cyg 0.02709290 BTC	322YDab3eDrcM9aSiazE Dy97KQn1vop2qF 0.03964172 BTC

En aquesta transacció, l'heurística detectaria com a adreça de canvi l'output 0, ja que si l'adreça de pagament fos l'output 1, es necessitarien els dos inputs per arribar a la quantitat necessària.

7. BIP 69

   BIP són les sigles de l'anglès "Bitcoin Improvement Proposal", que són documents on es presenta una proposta de millora tècnica, organitzativa o de qualsevol altre tipus pel desenvolupament de Bitcoin [14][15]. Abans d'aquesta proposta els inputs i outputs de les transaccions de Bitcoin no tenien un ordre en concret, creant així una debilitat a l'hora de poder detectar l'adreça de canvi en els outputs. Aquesta debilitat sorgeix en què la seguretat a l'hora de detectar l'adreça de canvi depenia del funcionament del wallet, fent que si es podia detectar el funcionament d'aquest, es podrien saber totes les adreces de canvi per totes les transaccions del wallet. Un exemple d'un mal funcionament que tenien alguns wallets, és que es posava l'adreça de canvi en l'últim output, fent que no s'haguessin d'aplicar més heurístiques, ja que podries detectar sempre l'output de retorn. Per evitar aquesta debilitat es va introduir el BIP 69, que obliga les transaccions a tenir un ordre en concret. L'heurística es basa a identificar el patró de comportament del BIP 69 per a poder saber si la transacció compleix amb aquest estàndard. El funcionament és simple, els inputs s'ordenen comparant les adreces de forma lexicogràfica i en el cas dels outputs, primer es comparen els valors de Bitcoins i en cas d'empat, les adreces de forma lexicogràfica [16]. Un exemple de transacció on no es compleix el BIP 69 és la *93a5e8...88f4c*:

bc1qkmwjjahtlpqsylsszsu vtp276yuhuqezu47aj5 0.00098527 BTC	3JSC5Wrcxm4RyU767pn evNCzrx39VbKtFD 0.00629174 BTC
bc1q567klmz9mfgm0xcq ak07kj4clynjfkv9pjnxnl 0.00610858 BTC	bc1qkzdk5qw8wrv48cpxg 9lkmjuamcqxdawdq5f26a 0.00068865 BTC

Com es pot observar, en el cas dels inputs, l'ordre hauria de ser al revés, ja que en el caràcter 5, k és més gran que 5. En el cas dels outputs, el valor de l'output 0 és més gran que el de l'output 1 i, per tant, també hauria d'anar al revés. En aquest cas, l'heurística detectaria que no s'està complint el BIP 69.

Finalment, trobem que es poden aplicar heurístiques en conjunt i, mitjançant un llindar, determinar l'adreça de retorn. Aquesta tècnica és coneguda com a votació per llindar i donada una transacció, s'apliquen les heurístiques segons una distribució d'importància determinada. Aquesta distribució vindrà donada pel nombre d'heurístiques en conjunt que s'utilitzin. D'aquesta manera, cada heurística tindrà un valor d'importància X sobre la votació, on la suma de Xi serà 1. Un exemple de distribució d'importància per les heurístiques, podria ser la següent:

- Tipus d'adreça per transacció - 0.25
- Detecció utilitzant decimals - 0.25
- Pagament de quantitat exacta o canvi òptim - 0.25
- Reutilització d'adreça 0.25

Si la suma del resultat de les següents heurístiques per a la mateixa adreça resultant és major que un llindar determinat, l'adreça en qüestió serà considerada com l'adreça de canvi. Agafarem d'exemple la transacció *407aca...c71f*:

3B1PRR77C2iERrShqzQ BbdMNPgm9rmYH8P 0.01765333 BTC	3HyR3fdnQ6dBBTpTYZ wu4vnxPQMWjh7FNT 0.01488006 BTC
	19Dcg7tjTeK4JDcj5sPVd nzFSHv7rfUuA6 0.00230221 BTC

Ara, doncs, analitzem cada heurística segons la distribució anterior:

- Tipus d'adreça per transacció - Podem veure que l'heurística d'Address Type detectaria l'output 0 com l'adreça de canvi, per tant, output 0 += 0.25.
- Per la resta d'heurístiques podem observar que no detectarien cap output de canvi, per tant, cap dels outputs sumaria més punts (address reuse hauríem d'analitzar la cadena de blocs sencera, però suposem que no detecta cap adreça nova).

En aquesta situació, l'output 0 tindria 0.25 punts i l'output 1 0 punts. Si suposem que el llindar és de 0.2 punts, aleshores, l'heurística detectaria l'output 0 com a l'output de canvi. D'aquesta manera, observem que aquesta heurística es veu molt afectada per la distribució de puntuacions que es faci i el threshold que se seleccioni.

6 Execucions de les heurístiques

Un cop explicades i programades les heurístiques que s'han utilitzat en aquest treball, s'ha comprovat la seva efectivitat en dues bases de dades de transaccions, la primera sintètica, creada per mitjà de fer transaccions entre diferents wallets amb el "Bitcoin Core", i la segona, en base a transaccions reals de la blockchain prenent com a groundtruth les dades de [4]. En primer lloc, s'han executat totes les heurístiques sobre les dades sintètiques i les proporcionades en [4] per a poder observar quantes adreces de canvi podia detectar cadascuna de les heurístiques. En la primera taula es pot observar els resultats per ambdós conjunts de dades:

Heurística	Adreces detectades en 8845 transaccions sintètiques	Adreces detectades en 10000 transaccions de [4]
Tipus d'adreça per transacció	81 - 0,92%	4435 - 44,35%
Detecció utilitzant decimals	1931 - 21,83%	515 - 5,15%
Pagament de quantitat exacta	0 - 0,00%	0 - 0,00%
Canvi òptim	2881 - 32,57%	5658 - 56,58%
Reutilització d'adreça	0 - 0,00%	3510 - 35,1%
Votació per llindar (llindar = 0.2)	4241 - 47,95%	6858 - 68,58%
Compleix BIP 69	2968 - 33,56%	3196 - 31,96%

6.1 Anàlisi dels resultats de les heurístiques

Com es pot observar, pel tipus d'adreça per transacció, hi ha una gran diferència entre els resultats. Aquesta diferència és provocada pel tipus de dades i com han estat generades. En el cas de les transaccions sintètiques, s'ha utilitzat el Bitcoin Core com a wallet i aquest wallet genera el mateix tipus de direccions, fent que l'efectivitat sigui molt baixa. D'altra banda, les transaccions de [4] són extretes de la cadena de blocs, fent que hi hagi molts tipus de wallets, avivant la seva efectivitat.

En segon lloc, podem observar que la detecció utilitzant decimals detecta més adreces de canvi per les dades sintètiques que per les de [4]. Aquest fet és causat perquè en les transaccions sintètiques s'han utilitzat en alguns casos valors rodons, sense decimals, i en restar-li les fees provoca que, en ser utilitzat l'output en un futur, generi decimals per la direcció de canvi, en contraposició als pagaments rodons, fent que l'efectivitat de l'heurística no sigui excessivament baixa. El baix nombre de transaccions detectades en el cas de les dades de [4] s'explica en el fet que normalment, els pagaments no es fan posant un preu en BTC sinó que es posa el preu en una moneda central i després es fa la conversió a BTC, que en ser un preu volàtil fa que es generin decimals i l'heurística quedi inutilitzada.

En el cas de l'heurística de pagament de quantitat exacta, l'efectivitat en els dos casos és del 0,00 per cent, ja que els dos groundtruth no disposen de cap transacció que només tingui un input i un output, per tant, l'heurística queda inutilitzada.

Pel cas de l'heurística de canvi òptim, es pot observar que l'efectivitat és alta comparada amb la resta d'heurístiques pels dos grups de dades. La diferència és donada pel fet que en el cas de les dades sintètiques, les transaccions són les primeres per a la cadena de blocs, provocant que les primeres transaccions no tinguin inputs suficients per a fer que l'heurística funcioni.

On sí que es pot observar una gran diferència és en l'heurística de reutilització d'adreça, on l'efectivitat en les dades sintètiques és nul·la, ja que ja que en les dades sintètiques no s'ha fet reutilització de cap adreça. Podem veure que el comportament de la cadena de blocs real, però, aviva l'efectivitat d'aquesta heurística, ja que normalment els outputs de les transaccions són reutilitzats, exceptuant l'output de canvi.

També es pot observar que la votació per llindar funciona amb una alta efectivitat tenint un llindar baix del 0,2. Com és lògic, l'efectivitat de les heurístiques era més alta per les dades de [4], per tant, aquesta també ho serà.

Finalment, cal destacar que s'han detectat un nombre similar de transaccions que compleixen amb el BIP 69 pels dos grups de dades.

En segon lloc, el tutor va proporcionar el groundtruth pels dos grups de dades, per tant, es poden aplicar les heurístiques de "tipus d'adreça per transacció, Detecció utilitzant decimals i canvi òptim" i comparar els resultats obtinguts amb el groundtruth, observant així quantes adreces de les detectades han estat encertades. Els resultats de les comparatives es poden observar en el següent requadre:

Heurística	Efectivitat dades sintètiques	Efectivitat dades [4]
Tipus d'adreça per transacció	46 encerts / 81 intents = 56,79%	2968 encerts / 4435 intents = 66,92%
Detecció utilitzant decimals	1775 encerts / 1931 intents = 91,92%	515 encerts / 515 intents = 100,00%
Canvi òptim	2696 encerts / 2881 intents = 93,58%	5289 encerts / 5658 intents = 93,48%
Reutilització d'adreça	0 intents = 0,00%	3228 encerts / 3510 intents = 91,97%
Votació per llindar (llindar = 0.2)	3911 encerts / 4241 intents = 92,22%	6222 encerts / 6858 intents = 90,73%

Cal destacar que els encerts són els casos en què les heurístiques han endevinat l'adreça de retorn, i els intents són els casos en què l'heurística ha donat un resultat. D'aquesta manera, es podrien treure els casos en què l'heurística ha fallat fent intents – encerts.

En primer lloc, trobem l'heurística de tipus d'adreça per transacció, que es pot observar que té una efectivitat relativament baixa. En el cas de [4] pot ser donat pels canvis dels wallets a l'hora de generar l'adreça de canvi de diversos tipus. Pel que fa a les altres heurístiques, es pot observar una efectivitat molt alta, encertant més del 90 per cent d'adreces de canvi. D'aquestes cal destacar que per la reutilització d'adreça en el cas de les dades sintètiques, no es pot calcular l'efectivitat, ja que no s'ha detectat cap adreça de canvi a causa de la forma de crear les dades pel wallet utilitzat.Per últim, cal destacar que l'efectivitat pel pagament de quantitat exacta tampoc es pot calcular pel mateix motiu i que pel BIP 69 no es pot calcular l'efectivitat, ja que aquesta heurística no dona una adreça de canvi sinó que indica algunes empremtes.

7 IMPLEMENTACIÓ DE LES ANTI-HEURÍSTIQUES

Un cop ja finalitzades les heurístiques així com les agrupacions en clústers de les que es podia fer, s'han dissenyat i programat les antiheurístiques. Com el nom indica, una antiheurística és una tècnica que modifica la transacció per a evitar que una heurística detecti l'adreça de canvi. D'aquesta manera, cada heurística tindrà una antiheurística que la inutilitzarà, fent que no es pugui detectar l'adreça de canvi o que es detecti una d'errònia. D'antiheurístiques n'hi ha de dos tipus diferenciats, on el destinatari de la transacció ajuda (donant una altra adreça on pagar-li) o on el destinatari no ajuda i, per tant, l'usuari que és la font de la transacció haurà de manipular els inputs per a poder fer l'antiheurística. D'aquesta manera, el primer cas s'identificarà com a tipus de defensa conjunta i el segon cas, com a tipus de defensa unilateral. Així doncs, l'usuari podrà fer un pagament segons el tipus de defensa que s'utilitzi. L'estratègia a seguir és clara, s'agafarà una transacció generada automàticament pel Bitcoin Core (creada mitjançant "fundrawtransaction" [17][18][19]) i se li passaran totes les heurístiques. Si alguna heurística encerta l'adreça de canvi, es passarà l'antiheurística corresponent per a inutilitzar-la, segons el tipus de defensa triat. Per exemple, si l'heurística de canvi òptim detecta l'adreça de canvi correctament, s'aplicarà l'antiheurística de canvi òptim corresponent, per a fer que no es pugui detectar l'adreça de canvi correctament. Un cop aclarit l'estratègia a seguir, explicaré quines antiheurístiques s'han dissenyat. Per al tipus de defensa unilateral, trobem:

- Tipus d'adreça per transacció - La premissa d'aquesta heurística és que tots els inputs han de ser del mateix tipus, per tant, l'antiheurística que s'ha pensat és afegir un input que sigui de diferent tipus dels que hi ha. Cal destacar que si no hi ha un input de diferent tipus, aquesta heurística serà impossible de defensar. Com que la transacció es fa més gran en afegir un input més, es pagaran més fees. Recuperant la transacció *4af2eb9f...f62ac00*:

bc1pq3h3mwygharvgm6f30dj98lmueg085frwkw6mvgr0qmven20cf9suz6g8t 0.01726280 BTC	bc1px3pv8jz2p8lgmmfrk9y6qdgpuc8ayus753telany2kzez3l7zdysyts00u 0.01171683 BTC
	1CVMDiKoGSLua8CCqAn2FHta41po9sqv7p 0.00551332 BTC

Defensada amb l'antiheurística quedaria:

bc1pq3h3mwygharvgm6f30dj98lmueg085frwkw6mvgr0qmven20cf9suz6g8t 0.01726280 BTC	bc1px3pv8jz2p8lgmmfrk9y6qdgpuc8ayus753telany2kzez3l7zdysyts00u 1.01171683 BTC
3… 1.00000000 BTC	1CVMDiKoGSLua8CCqAn2FHta41po9sqv7p 0.00551332 BTC

Com es pot observar, s'ha afegit una adreça de diferent tipus que l'input ja existent, inutilitzant així l'heurística.

- Detecció utilitzant decimals - La idea de l'heurística és trobar una única adreça amb més de dos decimals, per tant, l'antiheurística pensada es basa a pagar 0,00000001 BTC (1 Satoshi) més a l'adreça on es fa el pagament, fent així que no només hi hagi l'adreça de retorn amb més de dos decimals. Com que la transacció no es fa més gran, només es pagarà 1 Satoshi extra per a defensar la transacció. Recuperant la transacció *0f0a0b68...8cb019b478a*:

1BGMoN5Kz7amtyNshRomWT4ihhsYYQvRR3 0.01469490 BTC	bc1qdzsutzcnptsqvxgnecrvewgudxrpj8z909vnpc3efekh6sq6ga5q6c29ha 0.01000000 BTC
	18uhzy546Qz7CxRNkHohg4W9VSkfTkbSvY 0.00428230 BTC

Defensada amb aquesta antiheurística quedaria:

1BGMoN5Kz7amtyNshRomWT4ihhsYYQvRR3 0.01469490 BTC	bc1qdzsutzcnptsqvxgnecrvewgudxrpj8z909vnpc3efekh6sq6ga5q6c29ha 0.01000001 BTC
	18uhzy546Qz7CxRNkHohg4W9VSkfTkbSvY 0.00428230 BTC

Com es pot observar, s'ha afegit un Satoshi a l'adreça de pagament, fent que ara, els dos outputs tinguin més de dos decimals.

- Pagament de quantitat exacta - Aquesta potser és l'antiheurística més evident. Donat que només s'observa si hi ha un únic output, aquest output seria considerat de retorn. D'aquesta forma, la idea de l'antiheurística és afegir un input extra i un output extra. En fer més gran la transacció, es pagarà en forma de comissions. Recuperant la transacció vista en les heurístiques *a9ee22...0208*:

1Pbf6vv7WLCUEbFypcn3AAsDTdFEWx7TKr 0.00200000 BTC	3KbGRaaWkYfsVZPsPHe1FMceskX4ViUdKZ 0.00160000 BTC

Defensada amb aquesta heurística, quedaria:

1Pbf6vv7WLCUEbFypcn3AAsDTdFEWx7TKr 0.00200000 BTC	3KbGRaaWkYfsVZPsPHe1FMceskX4ViUdKZ 0.00160000 BTC
1… 1 BTC	1… 1 BTC

En tenir més d'un input i un output, l'heurística quedaria inutilitzada.

- Canvi òptim - En l'heurística s'observa que el valor de retorn sigui més petit que algun dels inputs. La idea

d'aquesta antiheurística és afegir inputs fins que el valor de retorn sigui més gran que tots els inputs. La clau en aquest procés per a trobar-lo és buscar quin és l'output que és més petit que algun dels inputs, és a dir, l'output de retorn, i anar afegint inputs fins que aquest output de retorn sigui més gran que l'input més gran. En fer la transacció més gran, es pagaran més comissions. Recuperant la transacció vista en les heurístiques *584c48...c47fbb*:

bc1qnm90x3wutuml58mzakned2ezur5qluwcqmgrrf 0.01304882 BTC	bc1q6dlg2sd5tcn653kqnmd403h7mvvath78g90yw8 0.00008000 BTC
bc1q96vxunrjdvxlj2ygkma60dfu5r8h8uv83f0cyg 0.02709290 BTC	322YDab3eDrcM9aSiazEDy97KQn1vop2qF 0.03964172 BTC

La transacció final defensada seria:

bc1qnm90x3wutuml58mzakned2ezur5qluwcqmgrrf 0.01304882 BTC	bc1q6dlg2sd5tcn653kqnmd403h7mvvath78g90yw8 1.00008000 BTC
bc1q96vxunrjdvxlj2ygkma60dfu5r8h8uv83f0cyg 0.02709290 BTC	322YDab3eDrcM9aSiazEDy97KQn1vop2qF 0.03964172 BTC
bc1q... 1 BTC	

Arribat a aquest punt, si es tornés a aplicar l'heurística de canvi òptim, identificaria l'adreça de canvi com l'output u, per tant, identificaria incorrectament aquesta adreça de canvi.

Pel tipus de defensa conjunta, trobem:

- Tipus d'adreça per transacció - Com que l'heurística es basa en el fet que l'adreça de retorn serà l'única del mateix tipus que els inputs, l'usuari destinatari ens donarà una altra adreça del mateix tipus que els inputs per a fer el pagament, fent que l'adreça de retorn no sigui l'única del mateix tipus. Com que la transacció es fa més gran en afegir un output més, es pagaran més fees. Una altra opció, si no s'hagués volgut mantenir l'essència de la transacció inicial, hagués estat utilitzar només 1 output de pagament però del mateix tipus que els inputs, sense necessitat d'afegir-ne un altre output. Així doncs, agafem la transacció *4af2eb9f...f62ac00*:

bc1pq3h3mwygharvgm6f30dj98lmueg085frwkw6mvgr0qmven20cf9suz6g8t 0.01726280 BTC	bc1px3pv8jz2p8lgmmfrk9y6qdgpuc8ayus753telany2kzez3l7zdysyts00u 0.01171683 BTC
	1CVMDiKoGSLua8CCqAn2FHta41po9sqv7p 0.00551332 BTC

Aplicant aquesta antiheurística, quedaria:

bc1pq3h3mwygharvgm6f30dj98lmueg085frwkw6mvgr0qmven20cf9suz6g8t 0.01726280 BTC	bc1px3pv8jz2p8lgmmfrk9y6qdgpuc8ayus753telany2kzez3l7zdysyts00u 0.01171683 BTC
	1CVMDiKoGSLua8CCqAn2FHta41po9sqv7p 0.00275666 BTC
	Bc1p... 0.00275666 BTC

- Detecció utilitzant decimals - La idea és que l'usuari donarà una altra adreça per a fer el pagament, fent que hi hagi dues adreces de pagament i, per tant, els hi pertany la meitat del valor total a cada adreça. Si aquest nou valor genera més de 2 decimals, no farà falta defensar la transacció, ja que l'adreça de retorn no serà l'única amb més de 2 decimals. En cas que el nou valor no generi més de 2 decimals, s'haurà de pagar 1 Satoshi extra a alguna de les adreces de pagament, traient-li aquest Satoshi a l'altra adreça. En qualsevol dels casos, només es pagaran les comissions extres de fer més gran la transacció (1 output extra), ja que en el cas d'haver d'afegir 1 Satoshi a una adreça de pagament, se li traurà a l'altra, fent que no hi hagi rebostos extres. Recuperant la transacció utilitzada anteriorment *0f0a0b68...8cb019b478a*:

1BGMoN5Kz7amtyNshRomWT4ihhsYYQvRR3 0.01469490 BTC	bc1qdzsutzcnptsqvxgnecrvewgudxrpj8z909vnpc3efekh6sq6ga5q6c29ha 0.01000000 BTC
	18uhzy546Qz7CxRNkHohg4W9VSkfTkbSvY 0.00428230 BTC

La transacció defensada quedaria així:

1BGMoN5Kz7amtyNshRomWT4ihhsYYQvRR3 0.01469490 BTC	bc1qdzsutzcnptsqvxgnecrvewgudxrpj8z909vnpc3efekh6sq6ga5q6c29ha 0.00500000 BTC
	18uhzy546Qz7CxRNkHohg4W9VSkfTkbSvY 0.00428230 BTC
	bc1q... 0.00500000 BTC

En haver-se generat més de dos decimals en els dos outputs de pagaments, no farà falta afegir-ne un Satoshi en una adreça i treure-li a l'altra.

- Pagament de quantitat exacta - La idea principal és que l'usuari destinatari donarà dues adreces per a fer el pagament, inutilitzant així l'heurística. En fer més gran la transacció, es pagarà en forma de fees. Recuperant la transacció *a9ee22...0208*:

1Pbf6vv7WLCUEbFypcn3AAsDTdFEWx7TKr 0.00200000 BTC	3KbGRaaWkYfsVZPsPHe1FMceskX4ViUdKZ 0.00160000 BTC

La transacció defensada seria la següent:

1Pbf6vv7WLCUEbFypcn3AAsDTdFEWx7TKr 0.00200000 BTC	3KbGRaaWkYfsVZPsPHe1FMceskX4ViUdKZ 0.00080000 BTC
	3… 0.00080000 BTC

- Canvi òptim - La idea principal és fer un pagament en 2 adreces, fent que alguna d'aquestes adreces sigui més petit que algun input. Per a no fer un pagament d'1 Satoshi en una adreça i en l'altra, la resta del pagament (això no ajudaria al destinatari), s'agafa un input qualsevol i se li resta un Satoshi, fent que aquesta quantitat anirà a parar en la nova adreça de pagament, i la resta del pagament a l'altra adreça. D'aquesta forma ens assegurem que almenys hi haurà dues adreces amb un valor més petit que almenys un input. El fet de fer la transacció més gran, es pagaran més comissions. Observant la transacció vista anteriorment *584c48...c47fbb*:

bc1qnm90x3wutuml58mzakned2ezur5qluwcqmgrrf 0.01304882 BTC	bc1q6dlg2sd5tcn653kqnmd403h7mvvath78g90yw8 0.00008000 BTC
bc1q96vxunrjdvxlj2ygkma60dfu5r8h8uv83f0cyg 0.02709290 BTC	322YDab3eDrcM9aSiazEDy97KQn1vop2qF 0.03964172 BTC

La transacció després d'aplicar l'antiheurística seria:

bc1qnm90x3wutuml58mzakned2ezur5qluwcqmgrrf 0.01304882 BTC	bc1q6dlg2sd5tcn653kqnmd403h7mvvath78g90yw8 0.00008000 BTC
bc1q96vxunrjdvxlj2ygkma60dfu5r8h8uv83f0cyg 0.02709290 BTC	322YDab3eDrcM9aSiazEDy97KQn1vop2qF 0.02659291 BTC
	3… 0.01304881 BTC

Amb aquesta transacció, almenys una de les adreces és més petita que algun dels inputs.

L'última antiheurística que s'ha dissenyat, és independent del tipus de defensa i és la del BIP69. Aquesta antiheurística es fa per a qualsevol tipus de defensa i consisteix a ordenar la transacció segons les indicacions del BIP 69, assegurant-se de no deixar així emprentes de la posició de l'adreça de retorn. Cal destacar que aquesta defensa depèn en gran part de si la resta de wallets utilitzen o no el BIP 69, ja que si s'ha rastrejat un output de canvi i la transacció a avaluar utilitza el BIP 69, mentre que altres wallets no l'utilitzen, seria una empremta en sí mateix. Així doncs, recuperant la transacció *93a5e8...88f4c*:

bc1qkmwjjahtlpqsylsszsuvtp276yuhuqezu47aj5 0.00098527 BTC	3JSC5Wrcxm4RyU767pnevNCzrx39VbKtFD 0.00629174 BTC
bc1q567klmz9mfgm0xcqak07kj4clynjfkv9pjnxnl 0.00610858 BTC	bc1qkzdk5qw8wrv48cpxg9lkmjuamcqxdawdq5f26a 0.00068865 BTC

La transacció defensada per l'antiheurística seria:

bc1q567klmz9mfgm0xcqak07kj4clynjfkv9pjnxnl 0.00610858 BTC	bc1qkzdk5qw8wrv48cpxg9lkmjuamcqxdawdq5f26a 0.00068865 BTC
bc1qkmwjjahtlpqsylsszsuvtp276yuhuqezu47aj5 0.00098527 BTC	3JSC5Wrcxm4RyU767pnevNCzrx39VbKtFD 0.00629174 BTC

Com es pot observar, s'han reordenat els inputs i els outputs segons el BIP 69.

8 Cost afegit per les anti-heurístiques

Com s'ha comentat anteriorment, per a defensar una transacció, en la majoria dels casos, comportarà haver de pagar quelcom extra en forma de comissions, o alguna heurística en concret, en forma d'1 Satoshi. L'augment de les comissions és causat pel fet que la mida de la transacció serà major i, per tant, tenint el mateix "fee rate", les comissions recomanades seran majors. Cal tenir en compte que per calcular les comissions només amb la informació del volum del bloc es faria de la següent manera: Comissions = vsize (Bytes) * fee rate (Sat/Byte). D'aquesta manera, es pot calcular quant de més s'ha hagut de pagar per a defensar una transacció. Cal destacar que abans de passar les antiheurístiques es crea una transacció normal. En aquest punt es pot calcular les comissions inicials segons el volum d'aquesta primera transacció. Després de passar les antiheurístiques, hi haurà una transacció defensada, i en aquest punt es podrà calcular la diferència de comissions que hi ha hagut. Sembla evident, que per a cada ronda d'execucions, les comissions seran diferents, ja que cada transacció tindrà unes heurístiques a defensar diferents i, per tant, actuaran unes antiheurístiques que modificaran o no el volum de la transacció i els valors de pagament.

8.1 Simulació del cost de les antiheurístiques

Per a comprovar que les antiheurístiques funcionen correctament i per a calcular les comissions en mitjana que s'han de pagar extra per a poder defensar la transacció, s'ha produït un entorn aleatoritzat que s'executarà cent cops per a extreure un resultat mitjà.

L'entorn és el següent:

1. Es crea un wallet nou i es minen 200 blocs per a tenir outputs per a gastar.
2. Es crea un altre wallet amb nom aleatori (número entre 1000 i 9999), el qual, si no existeix en la llista de wallets, se li farà un "complex payment" (pagament defensat amb antiheurístiques) des del primer wallet.
3. En aquest punt el nou wallet (o ja existent), tindrà com a mínim un output i, per tant, farà un "complex payment" a un wallet aleatori que pertanyi a la llista de wallets.

Aquests passos es repetiran un total de 7 cops produint-se de 6 a 12 "complex payments" (sent 12 el nombre de pagaments més probable). A aquest conjunt de pasos se li anomena l'l'execució d'un entorn". Aquest entorn es repetirà 100 cops, fent que en total s'hauran fet aproximadament

100 simulacions entorn * 12 complexPayments/simulació = 1200 complex payments. Per a mesurar les comissions en cada transacció creada pel çomplex payment"es calcula la diferència de les comissions inicials, amb les comissions finals on la transacció ja ha estat defensada. Per a cada entorn, es fa la mitjana de tots els çomplex payments"que s'hagin produït. D'aquesta manera, executant cent cops l'entorn, obtindrem cent mitjanes de les comissions que s'han hagut de pagar de forma afegida en cada entorn. Es pot observar aquest resultat en la següent llista:

[0.00000195, 0.00000371, 0.00000677, 0.00000941, 0.00001142, 0.00001417, 0.00001662, 0.00001912, 0.00002169, 0.00002302, 0.00002572, 0.00002846, 0.00003046, 0.00003137, 0.00003317, 0.00003653, 0.00003950, 0.00004103, 0.00004289, 0.00004375, 0.00004627, 0.00004713, 0.00004960, 0.00005046, 0.00005165, 0.00005331, 0.00005636, 0.00005927, 0.00006043, 0.00006257, 0.00006383, 0.00006687, 0.00006956, 0.00007174, 0.00007412, 0.00007535, 0.00007677, 0.00007805, 0.00008023, 0.00008192, 0.00008467, 0.00008603, 0.00008844, 0.00009047, 0.00009424, 0.00009692, 0.00010031, 0.00010265, 0.00010581, 0.00010801, 0.00010991, 0.00011231, 0.00011471, 0.00011720, 0.00011976, 0.00012182, 0.00012500, 0.00012619, 0.00012904, 0.00013016, 0.00013147, 0.00013292, 0.00013511, 0.00013541, 0.00013609, 0.00013661, 0.00013805, 0.00013945, 0.00014015, 0.00014032, 0.00014097, 0.00014183, 0.00014253, 0.00014424, 0.00014527, 0.00014629, 0.00014895, 0.00014924, 0.00015083, 0.00015257, 0.00015325, 0.00015341, 0.00015471, 0.00015651, 0.00015786, 0.00015870, 0.00016012, 0.00016135, 0.00016184, 0.00016232, 0.00016414, 0.00016503, 0.00016645, 0.00016685, 0.00016851, 0.00016916, 0.00016932, 0.00017092, 0.00017212, 0.00017297]

Un cop fetes les cent simulacions de l'entorn podem observar que la mitjana de comissions extra que s'ha de pagar per a defensar les transaccions és de 0,00009570 BTC per a "fee rates" d'entre 5 i 20 Sat/Byte. Es pot observar que pels primers entorns, les comissions afegides són més baixes perquè els UTXO a gastar són més grans, fent que per defensar les transaccions es necessiti utilitzar pocs UTXO. Els últims entorns, en tenir UTXO de valors més baixos, necessiten ajuntar més inputs per a defensar la transacció, fent així que la transacció sigui més gran i que, per tant, s'hagin de pagar més comissions.

9 Conclusions

Aquesta investigació m'ha permès obtenir un bon coneixement sobre el funcionament de la privacitat en les transaccions de Bitcoin. A causa de la naturalesa pública de la cadena de blocs, s'ha tingut l'oportunitat d'analitzar com el funcionament dels moneders deixa una empremta que permet vincular l'adreça de retorn d'una transacció amb l'usuari corresponent, tot tenint una alta efectivitat. Aquest fet, permet formar un clúster d'adreces associades a un usuari i, per tant, poder rastrejar els moviments que aquest realitza. D'altra banda, s'ha aprofundit en l'estudi de com eliminar les empremtes deixades pels moneders a través de la modificació de les transaccions, anomenant aquesta tècnica com a antiheurístiques. L'objectiu d'aquestes antiheurístiques és inutilitzar les heurístiques existents, buscant els casos on aquestes heurístiques fallen o donen resultats erronis. S'han identificat dos tipus de tècniques antiheurístiques, aquelles que només permeten la modificació dels inputs, sense l'ajuda externa del receptor, i aquelles que permeten la modificació dels outputs, que és quan el destinatari permet efectuar pagaments a diverses adreces. A través d'aquesta investigació, s'ha observat que pel primer tipus no sempre és possible eliminar completament l'empremta del moneder, cosa que implica que hi haurà transaccions on no es podrà protegir l'anonimat. És important destacar que l'ús d'aquestes antiheurístiques comporta un cost addicional, ja sigui en forma de comissions més altes o pagant 1 Satoshi extra al destinatari. En l'entorn simulat, s'ha constatat que aquest cost addicional no és excessiu i permet assegurar la privacitat de l'usuari en relació amb la divulgació de l'adreça de retorn. Així doncs, aquest treball m'ha permès comprendre el funcionament de la privacitat en les transaccions de Bitcoin i m'hha reptat a dissenyar i programar eines per abordar els reptes associats a aquest àmbit.

Agraïments

El principal agraïment que vull fer és pel meu tutor, Jordi Herrera, ja que m 'ha ajudat molt sobretot en ser una guia pel treball. Destacant dues parts, en l 'apartat d 'heurístiques l 'ajuda en les bases de dades de les transaccions, i en l 'apartat d 'antiheurístiques, l 'ajuda que em va proporcionar per donar exemples de crides RPC cap al "Bitcoin Core". A més a més d'haver fet que m'apasioni pel món de la tecnologia Blockchain i que hagi decidit aquest TFG.

Referències

[1] Y. Zhang, J. Wang and J. Luo, "Heuristic-Based Address Clustering in Bitcoin,ïn IEEE Access, vol. 8, pp. 210582-210591, 2020, doi: 10.1109/ACCESS.2020.3039570 – https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9265226

[2] Automatic Bitcoin Address Clustering: Dimitry Ermilov, Maxim Panov and Yury Yanovich. (2017) - https://bitfury.com/content/downloads/clustering_whitepaper.pdf

[3] Recuperat el 01/03/2023, de people.engr.tamu.edu, Bitcoin and Anonymity - https://people.engr.tamu.edu/bettati/Courses/489CryptoCurrencies/2017A/Slides/BitcoinAndAnonymity.pdf

[4] Resurrecting Address Clustering in Bitcoin: Malte Möser and Arvind Narayanan, booktitle: Financial Cryptography and Data Security, year: 2022, publisher: Springer International Publishing, pages: 386- 403, isbn: 978-3-031-18283-9 - https://arxiv.org/abs/2107.05749v2

[5] He Xi, Ketai He, Shenwen Lin, Jinglin Yang, Hongliang Mao, "Bitcoin address clustering method based on multiple heuristic conditions", in IET, 2022 - https://doi.org/10.1049/blc2.12014

[6] (2019). Recuperat el 01/02/2023, de Bitcoin.it, Privacy - https://en.bitcoin.it/wiki/Privacy

[7] (2019). Recuperat el 01/02/2023, de Bitcoin.it, Privacy, Example – Privacy altcoin mixing - https://en.bitcoin.it/wiki/Privacy#Example_-_Privacy_altcoin_mixing

[8] (2019). Recuperat el 22/02/2023, de Ionos.es, DigitalGuide, el-modelo-en-cascada - https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/

[9] (2023 actualitzat). Recuperat el 20/04/2023, de Bitcoin.org, Bitcoin Core - https://bitcoin.org/en/bitcoin-core/

[10] (2019). Recuperat el 01/02/2023, de Bitcoin.it, Privacy – Change Address Detection - https://en.bitcoin.it/wiki/Privacy#Change_address_detection

[11] (2019). Recuperat el 01/02/2023 de Medium.com, Cryptoquant, Introduction to Bitcoin Heuristics - https://medium.com/cryptoquant/introduction-to-bitcoin-heuristics-487c298fb95b

[12] (2023 actualitzat). Recuperat el 04/06/2023, de Blockchain.com, explorer, assets, BTC - https://www.blockchain.com/es/explorer/assets/BTC

[13] (2023 actualitzat). Recuperat el 01/02/2023, de Exodus.com - https://www.exodus.com/support/article/1918-preguntas-frecuentes-sobre-bitcoin-aprende-mas-acerca-de-btc

[14] (2019). Recuperat el 03/04/2023, de Academy.bit2me.com, Que es bip - https://academy.bit2me.com/que-es-bip-bitcoin/

[15] Recuperat el 03/04/2023, de Criptonoticias.com, Que es bip - https://www.criptonoticias.com/criptopedia/que-es-bip-propuestas-mejorar-bitcoin/

[16] (2019). Recuperat el 03/04/2023, de Bitcoin.it, BIP 0069 – https://en.bitcoin.it/wiki/BIP_0069

[17] (2021). Recuperat el 25/04/2023, de bitcoin.stackexchange, Fundrawtransaction – What is it - https://bitcoin.stackexchange.com/questions/105811/fundrawtransaction-what-is-it

[18] Recuperat el 25/04/2023, de developer.bitcoin, Fundrawtransaction - https://developer.bitcoin.org/reference/rpc/fundrawtransaction.html

[19] (2022). Recuperat el 25/04/2023, de bitcoin.stackexchange, Is there a way to avoid sendtoaddress from sweeping – https://bitcoin.stackexchange.com/questions/106967/is-there-a-way-to-avoid-sendtoaddress-from-sweeping