Privacy and anonymity in Bitcoin: A survey on Bitcoins anonymity techniques

Gabriel Rocha Gomes, Ivan da Silva Sendin

Abstract Bitcoin offers a lot of new possibilities in regards to how decentralized money can work, however there are a lot of misconceptions about how secure Bitcoin really is, if it is truly anonymous or not. in this article we discuss the levels of anonymity in Bitcoin as well as methods to achieve anonymity and what they really represent in the network in terms of privacy and anonymity.

1 Introduction

Bitcoin is a decentralized digital currency based on mutual trust, the idea was proposed by Satoshi Nakamoto in 2008 (**11˙nakamoto2012bitcoin**), or at least that was a pseudonym of the person or group who proposed the idea, since after the publishing and early implementation of the Bitcoin system Nakamoto vanished from public life. The currency used in the Bitcoin network is called bitcoin(BTC) and can be transferred to other users like a normal currency. To send and receive bitcoins one must create a key pair, one private key that allows the user to access the funds in a wallet, and a public key derived from the private in a one way cryptographic function that identifies the wallet, the most common format of identifier is a base58 hash of the public key generating a string of numbers and characters that usually starts with '1'. Once the pair is created a user can broadcast a transaction to the Bitcoin network to be verified and attached to the Blockchain by a miner.

Bitcoin is fully distributed peer-to-peer system (**11˙antonopoulos2014mastering**), where each node of the network stores a ledger containing all transactions made by all users, reading from this ledger is quite simple as the information is public available on the distributed ledger, writing on the other hand is more complicated, the system is designed in such a way to prevent anyone to just write a transaction directly on the blockchain, for that Bitcoin uses a process called mining in witch people compete to validate transactions and bundle them in to the next block on the blockchain.

Gabriel Rocha Gomes
Faculdade de Computação – Universidade Federal de Uberlândia (UFU), Caixa Postal, 38.400-902, Minas Gerais, Brazil.
e-mail: gabriel.gomes1@ufu.br

Ivan da Silva Sendin
Faculdade de Computação – Universidade Federal de Uberlândia (UFU), Caixa Postal, 38.400-902, Minas Gerais, Brazil.
e-mail: sendin@ufu.br

Anais do XV Encontro Anual de Ciência da Computação (EnAComp 2020). ISSN: 2178-6992.
Catalão, Goiás, Brasil. 25 a 27 de Novembro de 2020.

Bitcoin uses a proof-of-work system regarding the transactions in which a number of transactions, after verified, are built into a block of transactions, then the miners dispute to find a specific number that makes the block hash reach a specific condition, this process takes a lot of computing and whoever finishes it first gets to claim an amount of recently generated bitcoins to himself, this is called a block reward and currently is 6.25 bitcoins, this amount will decrease by half every 4 years the next cut is schedule for 2024, the miner also gets to collect a transaction fee for his work, a small amount of every transaction to stimulate miners to keep mining, the protocol states that in 2140 the halving will stop and the block reward will cease to exist as the 21 million coins will have all been mined, and the only incentive for mining will be the transaction fees. Each transactions has inputs and outputs, the former refers to the previous transaction in which those bitcoins were a part of, these allow the members of the network to verify if the bitcoins were not spent, the transactions usually have two outputs one destined to whomever the user is trying to pay and the other usually goes back to the user, since in a transaction a group of coins must be ”spent” entirely.

The disconnection between the wallet address and a person gives the user a sentiment of anonymity, however all the transactions ever made are permanently saved in the Blockchain so once a address is linked to a person it's possible to know their entire purchase history so the user relies on the address not being linked to his true identity, furthermore an attacker can try to analyze the behavior of a wallet or group of wallets in order to gain information on their identity, effectively ending the users anonymity, there are a lot of recent papers discussing this topic and providing new approaches to the problem (**11˙Meiklejohn2013**), (**11˙Ober2013**). So what Bitcoin offers is a pseudo-anonymity, for that reason there are some services that helps the user to cover his tracks once a address is linked to their true identity, and to avoid that from happening. In this paper we analyze this services and protocols in order to classify them and list their strengths and weaknesses, as well as tell if they are providing privacy and anonymity.

2 Techniques

The anonymization techniques listed in this article aim to attain some requisites, they try to disconnect the ”spending”of bitcoins from the ”receiving”of those bitcoins, in other words the one using the technique or service cannot be linked with the one receiving the spent bitcoins by that transaction, most of the services do that by making the information of who spent the coins after the protocol of the service is finalized not reliable.

2.1 Stealth addresses

Stealth addresses(**11˙todd2014**) allow anyone wishing to make a bitcoin payment to generate a unique address derived from a publicly known G generator and a shared secret between payer and receiver in a similar way to Diffie-Hellman key sharing, Once the address is created the recipient can spend the amount of the address in question.

2.1.1 The Basic Stealth Address Protocol(BSAP)

Assuming Alice wants to send Bob bitcoins using this stealth address system (**11˚hckrnoonSA**), Bob must first share a public key $Q = d.G$ with G being a public generator known to Alice and Bob. Alice will do the same by creating a key $P = e.G$, Bob then shares Q with Alice who computes $c = H(e.Q)$ Alice then sends bitcoins to a new address $Q' = c.G$. To ensure that Bob can access P Alice shares P as the OP_RETURN of the transaction that sends the value in question to Q' and Bob calculates for every possible P $c = H(d.P)$ to spend the bitcoins in Q'. this is not desirable as both would have access to Q' Bitcoins

2.1.2 The Improved Stealth Address Protocol (ISAP)

The start of the process is the same but Alice this time sends the bitcoins to a wallet $Q' = Q + cG$ where c.G is a G-derived generator that only the parties involved in the transaction have, Alice for generating it and Bob for being able to calculate for each possible key P $c = H(d.P)$ to once again ensure that Bob can access P Alice shares P as the OP_RETURN of the transaction that sends the value in question to Q '. Since $Q' = Q + c.G$ and $Q = d.G$ we deduce that $Q' = d.G + c.G$ or $Q' = (d + c).G$ so to spend Q' Bob's funds only need to calculate $d' = d + c$ for every candidate P.

The Dual Key Stealth Address Protocol

This method aims to improve ISAP (**11˚hckrnoonSA**). In it the key used to generate the shared address is not the same as scanning the blockchain so Bob would have $S = s.G$ and $B = b.G$ and Alice would have $R = r.G$ the shared secret $c = H(s.r.G) = H(r.S) = H(s.R)$ is calculated and the funds are sent to $B' = B + c.G$ Bob then finds $b' = b + c$ so that a scan key and a spending key s, this causes the 's' key not to be shared while maintaining good practice involving private keys if someone is 'listening' to your connection or there is a proxy server that can collect your information.

2.1.3 How does this provide anonymity

This method gives Bob anonymity if he wants to receive from many different sources, because even though they can be normally traced in blockchain it is not possible to link two different payments made to Bob.

2.1.4 The problem with Stealth addresses

One problem is that the same unlikable effect can be achieved if Bob automatically generates multiple addresses and keeps their keys stored, although this does not produce the Diffie-Hellman shared keys effect. this does not matter in practice because internet security protocols can, normally, take care of it.

Another problem, as pointed out by user Nate Eldredge in (**11˚Eldredge2014**) , is that the use of stealth addresses is so low that a user linked to this kind of practice can be more easily iden-

tified, also this method becomes less efficient the more the number of possible P keys increases creating a contradiction because a larger number of anonymity set makes method less efficient.

2.2 CoinJoin

Proposed by G.Maxwell (**11˙maxwell˙2013**) this protocol aims to make a set of users anonymous by allowing them to ”merge”small transactions into big one and the redirect the bitcoins to chosen outputs, for example, say Alice wants to send 1 bitcoins to Bob and Carol wants to send 1 bitcoins to Daniel, what coin join does is it mixes carol's coins so effectively what happens is Alice sends Daniel 1 bitcoin and Carol sends Bob 1 bitcoin, in the end Bob gets his bitcoins and so does Daniel but to an outside observer what is supposedly Alice's behavior is actually Carol's. For a small group of users this system does not provide a lot of anonymity, that increases as the set of users gets large enough, many services use that strategy, they are called mixing services, examples are, coinShuffle, Mixcoin, and according to Möser(**11˙moser2017anonymous**) the most widely used is Shared Coin provided by wallet service BlockChain[1].

This method on it's own has a problem that could break the user's anonymity, the set of people wishing to participate can be grouped in subsets of people by the amount paid, reducing the size of the set and therefore anonymity. to circumvent this behavior some implementations of mixing services adopted variable mixing fees

One good example is Mixcoin (**11˙Bonneau2014**), supposing that a person whose wallet is linked to his identity wants to use this service, he would contact the provider and from an agreement some parameters would be established these parameters are:

- v the amount to be transmitted;
- t_1 period within which A must transfer the amount to the provider;
- t_2 period within which the provider must send the amount to the A;
- k_{out} wallet address where the provider must deposit the amount;
- ρ the mixing fee A has to pay;
- n number used to randomize mix rates;
- ω number of blocks required to confirm payment of A.

These parameters initiate a protocol that, if successful, returns to k_{out} the value v. According to the protocol, this wallet must be an address created for the purpose of mixing, the protocol works as follows, A requests the service, if accepted, the provider (B) sends an address k_{esc} and a guarantee containing all the parameters of A plus the wallet address k_{esc} signed by a long-term key from B, this ensures that A can post this guarantee in case of theft by B, this induces B to act in good faith to ensure its reputation. If A sends the amount to k_{esc} an auxiliary function generates an X value between 0 and 1, using the parameters of the transaction itself and the blockchain if X ¿ ρ the entire amount is retained as a mix rate, this policy is called all or nothing and was proposed as a way to decrease the predictability of these transactions by maintaining a fair percentage of the rate, this also means that A does not want to mix a very high amount but in fact several small amounts sequentially, if the value that is not retained is sent to k_{out} it is the responsibility of both parties to delete all records of the transaction by increasing A.'s guarantee of anonymity.

[1] `www.blockchain.info`

2.3 CoinSwap

Coin Swap(**11˙maxwell˙coinswap2013**) is a technique that allows someone to pay bitcoins without making a direct connection to the receiver by the means of a third party(Carol). Let's say Alice wants to pay Bob a certain amount of bitcoins, but doesn't want anyone tracing this transaction, in this case Alice could pay Carol that would pay Bob the same amount of coins. The problem is that do far this technique would require that Alice completely trusted Carol that could simply not pay Bob.

The Coin Swap protocol proposed at first by G.Maxwell in (**11˙maxwell˙coinswap2013**) this method works in a way that if one of the parts fails to fulfill the protocol the other can refund the bitcoin by using a 2-of-2 multisgnature output (**11˙antonopoulos2014mastering**) in a way that guaranties that everyone is satisfied or the anonymity is lost

To initiate the protocol Alice creates a 2-of-2 multisgnature output T0 that requires Alice's and Carol's keys to be spent, Carol does the same to Bob T1, to ensure that no one walks away with the money two time-locked refund transactions are created. Now Bob can select a random value x and calculate and send a Hash(x) to Alice and carol, Alice then creates a hash locked transaction as a guarantee that carol, with her signature, can spend T0 once she knows x. Carol does the same to T1 for Bob, this rather complicated set of transactions guarantee that the respective parties will receive their money, however both transactions rely on x and that would generate a link between Alice and Bob, if every party respects the deal Carol can simply make a separate transaction to Bob and Alice a separate transaction to Carol that way preserving the anonymity (**11˙moser2017anonymous**).

2.4 Fair Exchange

The idea of the fair exchange protocol (**11˙barber2012bitter**) is to exchange currencies between users, say that Alice exchanges currencies with Bob, that is, Alice generates a wallet with a certain amount of bitcoins and shares the secret with Bob, who in turn does the same and shares with Alice, this makes someone observing Alice's behavior starts to observe Bob's behavior. As long as Alice and Bob are Honest and are interested in carrying out the protocol correctly, the protocol ends and everyone is happy, but this requires trust between the two parties, the Fair Exchange ensures that the parties can enter the protocol without trusting one another.

To initiate the protocol both parties have to create a set of numbers a and b, they then engage in a cut-and-chose protocol to ensure correctness, Bob than creates a T1 that can be spent with Alice's signature plus Bob's signature or knowing the value of b, Alice than creates a T2 that can be spent With Bob's signature plus Alice's signature or the knowing of $(a + b)$, both parties sign refund transactions for T1 and T2 the transactions are published and the setup is complete, now Bob can spend Alice's coins by providing
$(a + b)$ and Alice can then calculate b by subtracting it from $(a + b)$, and spend Bob's coins. if Bob never posts $(a + b)$ Alice will get her funds back if Alice abandons the protocol early Bob will get his funds back (**11˙moser2017anonymous**). That way the protocol can be carried in a trustless manner.

3 Analyzing the privacy and anonymity provided by those methods

3.1 Defining privacy and anonymity

To understand how these methods can provide privacy and anonymity first we need to define those concepts. Thomas Wright (**11˙Wright2004**) defines anonymity within two main concepts, total anonymity and pseudonimity. Total anonymity "*means that the origin of communication is made totally untraceable*". pseudonimity "*is concealing a real identity by the use of an alias.*"knowing the behavior of Bitcoin we can safely say the it operates under pseudonimity as the wallets serve as an alias for a user

Wright (**11˙Wright2004**) also gives us some definitions for privacy two of witch, information privacy and privacy of communication, are important to us. Information privacy according to wright means "*data protection covering the collection and handling of personal information, such as medical, credit, residential information, but also government records etc*". privacy of communication "*covers the security of mail, telephones, e-mail and other forms of communication*", security meaning confidentiality, integrity and availability"

Taking in consideration the behavior of the Bitcoin protocol it cannot provide without modifications or a intermediate layer of operation, neither total anonymity nor privacy, total anonymity because the transactions can be traced back to the sender, information privacy because the details of every transaction are available to the public breaking their confidentiality and finally privacy of communication because every transaction is available to the public.

Defining metrics to compare the methods

In order to compare the methods we need to set a number of metrics those will help us define the purpose of the methods used, those metrics are:

Intermediaries the method requires any third party to act as an intermediary, this is important because the third party often a middle man will have the information of all parties involved in the exchange;

On / off blockchain the method is executed completely within the Bitcoin blockchain, not considering necessary information exchanged outside the blockchain, this will show us the intended use of the method because any technique that is done completely on the blockchain cannot provide full anonymity to the parties involved, methods that use an intermediary layer might be able to achieve that;

Communication the method requires the communication via any method outside the blockchain between the interested parties to be completed, this is important as the extra layer of communication can provide another medium to decrease anonymity;

Fees the method uses fixed fees that may allow grouping or probabilistic fees to avoid grouping, the CoinJoin methods that use probabilistic fees should theoretically make identifying these transactions harder;

Changes on Bitcoin protocol the proposed method requires any changes to the protocol, this is important because any technique that does not provide changes to the protocol won't be able to achieve total anonymity;

Implemented the method has implementations available for use, or it has to be executed by the interested parties themselves, techniques that have a readily usable implementation may be used more than techniques that require the user to have more knowledge on Bitcoin to execute the technique.

	Stealth Addresses	CoinJoin	CoinSwap	Fair Exchange
Intermediaries		✓	✓	
on blockchain	✓	✓	✓	✓
off blockchain				
require communication outside the blockchain	✓	✓	✓	✓
fees	no	static or probabilistic	no	no
require changes on bitcoin protocol				
implemented		✓		

Tabela 1: Characteristics of anonimitization techniques

The table 1 gives us important information about these techniques. First of all none of the methods can provide true anonymity or privacy because all of them work on the blockchain and don't require modifications to it, therefore they are limited by the protocols own limitations regarding anonymity and privacy. This leaves us with pseudonimity witch is the intended outcome of all this methods. this table also gives us information regarding the individual methods, for instance the only method with an actual implementation, as far as we know, is the CoinJoin, that implementation being the mixing services, witch is also the only method that require service fees outside of Bitcoin's normal transaction fee, also CoinJoins can have or not the implementation of probabilistic fees witch make the method more secure.

The use of an intermediary introduces a liability as the third party has all the information about the other participants, and motif to be dishonest, but the two methods in questions have workarounds to guarantee honesty as seen before. Ideally a method would not require third parties, would have a widely used and trusted implementation using the maximum amount of tools to to mask the transaction outputs from an attacker, note however that the table doesn't specify the intensity of the communication needed as some methods like stealth addresses require minimal communication. The methods studied in this article appear to have one or two of these characteristics but not all, however this just shows that there is room for improvement, using this methods as frameworks to build more complex and secure methods.

4 Related works

In (**11˙moser2017anonymous**) the authors discuss the same four techniques in this article, giving a general view of how the technique is executed and then doing several analysis in the Bitcoin's blockchain to count the number of transactions possibly related to this techniques.

To achieve this they proposed a few metrics for each technique as well as the expected behavior of the transaction related to that technique, then they analyze every transaction until June of 2016 to sort out which are related to the four techniques.

For the stealth addresses they search for transactions with raw public key in the OP_RETURN of the transaction, for a 3 months period. As this is required for the protocol to work therefore is a probable giveaway. The numbers of transactions per period of time found are relatively low, about 0 to 60 per period, the first transaction was situated in February of 2014, even though the adoption is generally low there is a spike of use in the period of April to June of 2015 that should be a interesting study topic.

For CoinJoin they define a set of rules, and separate transactions with at least four outputs: two for spending and two for change, and require all unique addresses to mach these condition and then eliminating some known false positives. The number of possible Coinjoin transactions are high relative to the other methods, in the paper the authors also count the origin of the payment and find that the biggest part(57%) comes from the Blockchain.info's Shared coin service.

For CoinSwap they take all transactions with a 2-of-2 multsignature requirement, and then define 5 criteria to estimate the potential number of transactions, after sanitizing the results they find that there is a large enough anonymity set for these type of technique but rule out most of the transactions as normal use.

For Fair Exchange they analyze Pay-to-script-hashes, after that they sanitize the set by extracting the top 100 non-standard scripts, but after analyzing the scripts they don't find any that match the requirements for a Fair Exchange and rule out the technique as not seen any pratical use.

This work produces a lot of interesting information about the techniques, the number of possible transactions related to them as well as the distribution of service in the case of CoinJoin, and the large set of anonymity for the CoinSwap method.

5 Conclusion

There is no possibility to achieve total anonymity or privacy in the Bitcoin blockchain without changing the protocol or using an external layer, the methods studied in this article seem to be trying to improve or recover pseudonimity, the use of CoinJoin can take funds from a compromised wallet to an untracked one, Coin Swaps and Fair Exchanges aim to mislead observers into watching the behavior of another individual and stealth addresses aim to detach different payments made to the same person making them unlinkable. instead of trying to improve on what the protocol can do these methods ensure that the level of anonymity provided by the protocol is preserved.

Referências

FERNANDES, Miguel A. et al. A framework for wireless sensor networks management for precision viticulture and agriculture based on IEEE 1451 standard. **Computers and Electronics in Agriculture**, 2013.

HIGUERA, J.; J., Polo; GASULLA, M. A ZigBee wireless sensor network compliant with the IEEE 1451 standard. In: PROCEEDINGS of the IEEE Sensors Applications Symposium (SAS' 2009). New Orleans, LA, USA: [s.n.], 2009. p. 309–313.

IEEE INSTRUMENTATION e MEASUREMENT SOCIETY. **Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats**. New York, 2007.

_____. **Network Capable Application Processor (NCAP) Information Mode**. New York, 1999.

_____. **Wireless Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats**. New York, 2007.

LEE, K. **IEEE 1451 and IEEE 1588 Standards**. [S.l.: s.n.], 2017. National Institute of Standard and Technology (NIST). Disponível em: https://www.nist.gov/sites/default/files/documents/el/isd/ieee/Information-on-1451_1588-V36.pdf. Acesso em: 20 de Dezembro de 2017.

LEE, Kang; SONG, Eugene. A Wireless Environmental Monitoring System Based on the IEEE 1451.1 Standards. **Instrumentation and Measurement Technology Conference**, 2006.

MANDA, S.; GURKAN, D. IEEE 1451.0 Compatible TEDS Creation Using .Net Framework. In: PROCEEDINGS of the IEEE Sensors Applications Symposium (SAS' 2009). New Orleans, LA, USA: [s.n.], 2009. p. 281–286.

RIBEIRO, J. M. **Desenvolvimento de uma ferramenta em sistema embarcado para o reconhecimento dos nós de rede sem fio baseado no padrão IEEE 1451**. Catalão, Goiás, 2015.

SONG, E. Y.; LEE, K. Understanding IEEE 1451-Networked smart transducer interface standard - What is a smart transducer? **IEEE Instrumentation Measurement Magazine**, v. 11, n. 2, p. 11–17, 2008. DOI: 10.1109/MIM.2008.4483728.