User Privacy in the Public Bitcoin Blockchain

Name Surname, Name Surname, Name Surname, *Member, IEEE,* Name Surname, and Name Surname, *Senior Member, IEEE*

***Abstract*—Bitcoin is a peer-to-peer electronic cash system that maintains a public ledger with all transactions. The public availability of this information has implications for the privacy of the users. The public ledger consists of transactions that transfer funds from a set of inputs to a set of output addresses. As long as those addresses cannot be linked to their owners, privacy is preserved. The linking of addresses to owners results in privacy leaks. The possibilities of linking addresses to owners are multiplied when addresses are used to receive funds more than once. In this work we describe privacy-leaking effects of address reuse and gather statistics of address reuse in the Bitcoin network.**

We also describe collaborative (CoinJoin) transactions that prevent the privacy attacks that have been published in the literature. Then we analyze the Blockchain to find transactions that could potentially be CoinJoin transactions.

***Index Terms*—Bitcoin, cryptocurrency, privacy, address reuse, CoinJoin**

I. Introduction

BITCOIN is a peer-to-peer electronic cash system [1] that maintains a public ledger with all transactions. All the transactions need to be available to the peer-to-peer network that guarantees the security of the system. Any full node of the network stores in a database all the transactions in the history of Bitcoin. This database is typically referred to as the *Blockchain*.

With traditional cash, there is no public record of all the transactions. And the traditional banking system keeps the transactions of their customers private. The public availability of the *Blockchain* represents a novelty and its privacy implications are worth studying. Ideally, any new payment system should offer privacy guarantees at least as good as traditional systems.

To receive a Bitcoin payment, the payee must provide the payer a Bitcon address to which the payment will be sent. Both the Bitcoin community and previous research studies agree that address reuse is in general a bad practice. It is recommended to generate a new address for each payment to be received. These addresses that are used only once are called disposable addresses.

In this paper we first review the basic elements of the Bitcoin system necessary for the subsequent discussion. Then we discuss the necessity of avoiding address re-use to prevent unnecessary privacy leakage. After that, we analyze the Blockchain to determine to which extent address reuse occurs in the Bitcoin community. Disposable addresses do not offer total protection against privacy leakage. We detail the remaining risks as well as possible solutions.

The authors

II. Address Reuse

A. Basic Bitcoin Elements

Bitcoin is a protocol, a network, and an Internet currency unit. We capitalize the word when we refer to either the protocol or the network.

Transactions are a fundamental element of Bicoin. Payments require transactions, and these transactions are shared with the network and securely stored in the Blockchain. Each transaction consumes some inputs and creates some outputs. The inputs and outputs are worth bitcoins, and for a regular transaction to be valid the total value of the outputs must not exceed the total amount of the inputs.

The outputs of one transaction can be used as inputs of other transactions. Critically, each output can be used only once. In this sense, available outputs are just like available money, which can also only be spent once. Another name for the available outputs is available coins. The network does not accept transactions that try to spend coins that have been spent before.

An example transaction is presented in listing 1. Long hexadecimal strings have been trimmed to save space. A transaction is identified by a hash, which is the first field. This particular transaction has a single input (“in”) which is the first output (“n”:0) of a previous transaction with a hash starting with a777.

The example transaction has two outputs worth 22 bitcoins and approximately 0.87 bitcoins. It is possible that the first output is a payment and the second one is a change. The “scriptSig” field in the input contains the signature required to spend the coins in the input. The “scriptPubKey” field in the outputs describes the signature required to spend those outputs.

Listing 1. Example Bitcoin Transaction

```
{
"hash":"1093[...]",
  "ver":1,
  "vin_sz":1,
  "vout_sz":2,
  "lock_time":0,
  "size":258,
  "in":[
    {
      "prev_out":{
        "hash":"a777[...]",
        "n":0
      },
      "scriptSig":"3045[...]"
    }
  ],
  "out":[
    {
      "value":"22.00000000",
```

```
      "scriptPubKey":"OP_DUP OP_HASH160 17ed[
          ...] OP_EQUALVERIFY OP_CHECKSIG"
    },
    {
      "value":"0.87213300",
      "scriptPubKey":"OP_DUP OP_HASH160 9319[
          ...] OP_EQUALVERIFY OP_CHECKSIG"
    }
  ]
}
```


In the example we can observe that another important element of Bitcoin are public/private asymmetric cryptographical keys. The public key is hashed and coded with some redundancy into base58 addresses. These addresses are alphanumeric chains that can be used to receive funds. The outputs of a transaction can be sent to an address, which is simply a convenient representation of a public key.

In order to spend an output, it is necessary to offer proof of ownership of the address and, consequently, of the output. This proof is the evidence of knowledge of the private key corresponding to the address. A user willing to spend an output sent to a given address must provide the public key that hashes to that address and a valid signature. The signature can be generated only by the owner of the private key.

There is no limit in Bitcoin regarding the number of transactions that can use a given address for the outputs. The owner of the address' private key will be able to spend each of those outputs once. Another particularity of bitcoin is that there is no practical limit of the number of addresses that can be generated. There are 2^{160} possible addresses because they are generated using RIPE-MD160. This makes it possible for users to generate new (disposable) addresses for each incoming payment.

A particular form of outputs which is relevant to the discussion later in this work are *change* outputs. A user willing to send a Bitcoin payment needs to combine in a transaction a number of inputs of value equal or larger than the desired payment. If the value of the inputs is larger than the output, it is likely that the sending part does not want to loose the difference, also called change. In order to keep the change, the payer creates a transaction with inputs exceeding the payment value and two outputs: the actual payment and the change. The payment is sent to the payee and the change is sent to an address controlled by the payer.

It is a common and recommended practice that the amount of the inputs is slightly higher than the value of the outputs. This small difference is called a *fee* and it is kept by the peers that contributed to the security of the network.

B. The Temptation

The temptation exists of using Bitcoin addresses just like regular bank accounts numbers. As addresses are used to receive payments, technically speaking only once is needed. We can use the same address to receive all our payments, just like we can use a single bank account to receive all our payments. Obviously, in this case, we will also use the same address for all the payments that we make, which is in agreement with the bank account parallelism.

The difference is that our bank account transactions and balance are kept private by our bank. In contrast, all the transactions and balances in Bitcoin are publicly available to anyone with Internet access. We could retain our privacy if we were able to keep our Bitcoin addresses private. Each time that we participate in a transaction, the other party may have the opportunity to link our address to our identity.

Address reuse is discouraged in the original Bitcoin paper [1].

C. Address Reuse in the Blockchain

Even though address reuse might have worrying consequences, the fact is that we can find address reuse in the Blockchain. We use Obelisk and Libbitcoin to download and query the Blockchain for evidence of address reuse. Both Obelisk and Libbitcoin are open source projects under heavy development. Our source code to generate the data and the plots presented in the paper is also available in github[1]. We sweep all the transactions in the Bitcoin history and for every address we count how many times it appears as an output of a transaction.

Some statistics of the distribution are presented in Table I. A histogram of the data is presented in Fig. 1. For readability reasons, we only show the first 100 positions of the histogram. The distribution has a long tail and some addresses are used over one million times.

D. Wikileaks Privacy Leakage

Wikileaks funding campaign is an example of address reuse. At the time of this writing, Wikileak's donation webpage offers by default a re-used address. It also offers the donors the possibility of generating a new (disposable) address by simply clicking a button. The public address makes it possible for everyone to inspect all the details of the transactions involving that address. A Blockchain explorer website (such as `blockchain.info`) can be used to browse all those details. At the time of this writing, Wikileak's public address has received over 3,854 bitcoins in 2216 transactions. The source addresses for each transaction are also public.

Wikileaks address reuse can unintendedly leak donors' privacy unless they are very cautious. If the output of a transaction is later sent to Wikileaks in a second transaction, the payer of the first transaction can learn that the payee of the first transaction is a Wikileaks donor by simply inspecting the Blockchain.

There are advanced techniques that we review later in this work that can result in increased privacy leakage. These techniques exploit the usual behaviour of Bitcoin *wallets* and *change addresses*.

E. Bitcoin Wallets

From the previous section it should be clear that address reuse has negative consequences to the user's privacy. A partial solution to Bitcoin's privacy weaknesses is the use

[1] https://github.com/jbarcelo/txfillstat

TABLE I
ADDRESS RE-USE STATISTICS

Mean	3.18
Min	1
25th perc.	1
50th perc.	1
75th perc.	1
Max	1,238,931
Number of addresses	12,963,199
Number of uses	41,244,997
Addresses used once	10,476,899
Addressed used twice	1,397,373
Used over 100 times	25,004

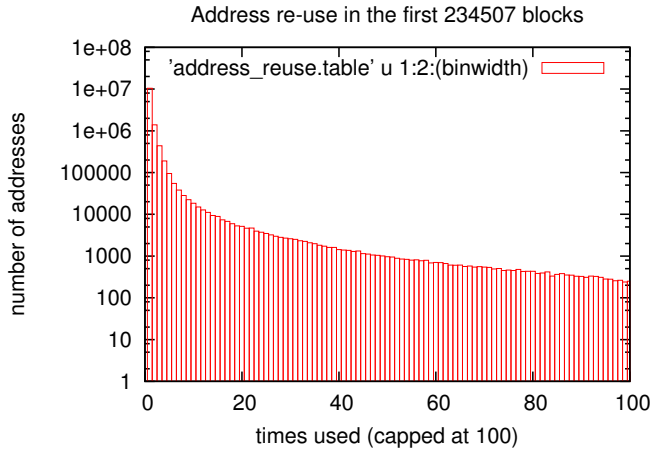

Fig. 1. Number of addresses for re-use factor from 1 to 100. Note the log scale.

of disposable addresses. Disposable addresses are used only once and therefore make it more difficult to link different transactions to the same user.

The reference Bitcoin peer implementation, *bitcoind* and many other software packages make it possible for the Bitcoin user to comfortably handle a large number of addresses. The software that takes care of Bitcoin addresses is called a Bitcoin wallet. This software generates as many addresses as needed. Some of these addresses are used for receiving payments while others are used to receive change outputs. The software does not reuse change addresses and therefore, if the user is cautious enough to avoid address reuse for payment reception, addresses are used a single time to receive funds.

The software wallet also computes the sum of the funds available in all the addresses to present a total balance to the user. The wallets also assists the users in the creation of transactions. Among all the outputs available to the user, the wallet picks those that are used as inputs of the new transaction. Change addresses are also handled transparently for the user.

The software keeps all the private keys of the user, and typically keeps them encrypted with a single password that the user needs to remember.

F. Wallet Privacy Leakage

Typical operation of Bitcoin wallets potentially leaks some information about its users. This leakage has been exploited in the past [2]. The first form of privacy leakage is when a wallet combines different outputs as inputs for a single transaction. This combination is necessary when the software creates a transaction for a payment value that exceeds that of the individual available coins. When the transaction is broadcast, an attacker can infer that all the inputs of the transaction and their associated addresses belong to the same user.

It is important to highlight at this point that the Bitcoin protocol does not require that all the inputs belong to the same wallet. This is simply a common practice in widespread software wallets. Therefore is not a protocol vulnerability, but an implementation vulnerability.

A second possible attack involves change addresses. There are techniques to try to infer which of the outputs of a particular transaction is a change output. For example, in a transaction which presents two outputs and one of them is smaller than all of the inputs, it can be assumed that the small output is a change output. Again, this is simply an assumption that relies on the way that popular wallets operate today and by no means is an inherent restriction of the protocol. If a change output of a transaction is identified and later used as an input in another transaction, an attacker may infer that all the inputs of the first and second transaction belong to the same wallet.

As most of the transactions have a change output and this output can be an input of another transaction, a number of transactions can be chained together in a privacy leaking chain.

Privacy invading techniques in the literature rely on heuristics that take advantage of *idioms of use*. It is noted in [2] that a change of current Bitcoin practices may invalidate current privacy attack methods.

To the best of the authors knowledge, techniques currently available in the literature cluster addresses belonging to the same wallet but do not attempt to link different wallets of the same user. Therefore, the use of different wallets may offer partial protection against published privacy attacks.

G. Enhanced Privacy Measures

The privacy attacks described in the previous subsection take advantage of the current behaviour of software wallets that create transactions in which all of the inputs belong to the same user. The attacks can be disrupted by introducing transactions in the Blockchain that intendedly violate the assumptions on which the privacy attacks rely. If different users collaborate in creating a transaction and the coins that are used as inputs belong to the different users, the attacks currently available in the literature will no longer be reliable. These transactions create an additional layer of confusion which results in additional protection for the privacy of the users.

The combination of inputs of different users has been termed *CoinJoin* by the Bitcoin community. In the simplest case, two users combine two inputs to make two payments

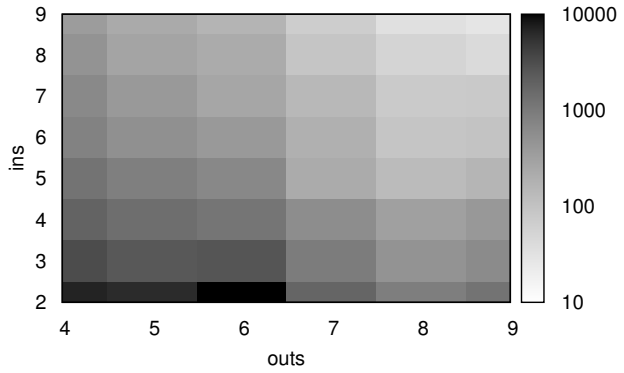

Fig. 2. Number of potential CoinJoin transactions for a given number of inputs and outputs.

in a single transaction. A simple CoinJoin transactions has two or more inputs from two different users, two outputs which are the two payments, and two change addresses for the two payers. If the inputs are of similar size, a payee cannot determine which inputs belong to the payer. Similarly, a Blockchain observer cannot link different inputs as belonging to the same user.

There is no limit in the number of users that can combine their payments in a single Bitcoin transaction. In fact, a larger number of collaborators makes it even more difficult for the attacker to extract information from the Blockchain. The only limitation is that to participate in a CoinJoin transaction, it is necessary to find other users that want to make a payment at the same time. At the time of this writing, mainstream wallets do not offer automated tools for finding collaborators for a CoinJoin transactions and therefore CoinJoin is seldomly used.

Nevertheless, even if only a fraction of users actually participates in CoinJoin transactions, all Bitcoin users benefit from increased privacy. The existence of some CoinJoin transactions undermine the applicability of published privacy attacks and therefore more sophisticated attacks that circumvent CoinJoin transactions are needed to reliably extract information from the Blockchain.

We scan the Blockchain for transactions with two or more inputs and four or more outputs as they are possible CoinJoin transactions. In Fig. 2 we present results for a number of inputs and outputs lower than 9. The total number of transactions in the Blockchain that could be CoinJoin transactions is XXXX.

ACKNOWLEDGMENT

The authors would like to thank ...

REFERENCES

[1] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008.
[2] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, “A fistful of bitcoins: characterizing payments among men with no names,” in *Proceedings of the 2013 conference on Internet measurement conference*. ACM, 2013, pp. 127–140.