Privacy in Bitcoin Transactions: New Challenges from Blockchain Scalability Solutions

Jordi Herrera-Joancomartí(✉) and Cristina Pérez-Solà

Dept. d'Enginyeria de la Informació i les Comunicacions,
Universitat Autònoma de Barcelona, 08193 Bellaterra, Catalonia, Spain
jordi.herrera@uab.cat, cperez@deic.uab.cat

Abstract. Bitcoin has emerged as the most successful cryptocurrency since its appearance back in 2009. However, its main drawback to become a truly global payment system is its low capacity in transaction throughput. At present time, some ideas have been proposed to increase the transaction throughput, with different impact on the scalability of the system. Some of these ideas propose to decouple standard transactions from the blockchain core and to manage them through a parallel payment network, relegating the usage of the bitcoin blockchain only to transactions which consolidate multiple of those off-chain movements. Such mechanisms generate new actors in the bitcoin payment scenario, the Payment Service Providers, and new privacy issues arise regarding bitcoin users. In this paper, we provide a comprehensive description of the most relevant scalability solutions proposed for the bitcoin network and we outline its impact on users' privacy based on the early stage proposals published so far.

Keywords: Bitcoin · Scalability · Off-chain transactions · Lightning network · Duplex micropayment channels

1 Introduction

Bitcoin is an online virtual currency based on public key cryptography, proposed in 2008 in a paper authored by someone behind the Satoshi Nakamoto pseudonym [1]. It became fully functional on January 2009 and its broad adoption, facilitated by the availability of exchange markets allowing easy conversion with traditional currencies (EUR or USD), has brought it to be the most successful virtual currency.

The success of bitcoin has evidenced its weak design in terms of scalability since the number of transactions per second that the system may handle is orders of magnitude lower than standard globally used systems, like VISA.

In order to allow bitcoin to scale and to have a chance to be a global payment system, different solutions have been proposed. Although some of them are still in development, in this paper we point out the most relevant ones, that is, proposals that have a large acceptance degree in the community, focusing on those

V. Torra et al. (Eds.): MDAI 2016, LNAI 9880, pp. 26–44, 2016.
DOI: 10.1007/978-3-319-45656-0_3

that present an important shift in the bitcoin development: off-chain payment channels. Besides the general paradigm change that off-chain payment channels may suppose, we are mainly interested on how bitcoin users' privacy could be affected by such proposals.

The organization of the paper is as follows. In Sect. 2 we provide a general background of the bitcoin system and outline its scalability problems. Section 3 points out the main proposals to scale the bitcoin system, focusing in the off-chain payment channel solution. How bitcoin users' privacy will be affected by off-chain payment channels is discussed in Sect. 4, mainly analysing how actual techniques used to attack/protect users' privacy will be affected. Finally, Sect. 5 concludes the paper and gives some guidelines for further research in this field.

2 The Bitcoin System

In this section, we point out the main ideas to understand the basic functionality of the bitcoin cryptocurrency. Such background is needed to understand the scalability problems the system faces and the solutions that have been proposed. However, the complexity of bitcoin makes impossible to provide a full description of the system in this review, so interested readers can refer to Antonopoulos's book [2] for a detailed and more extended explanation on the bitcoin system.

Bitcoin is a cryptocurrency based on accounting entries. For that reason, it is not correct to look at bitcoins as digital tokens since bitcoins are represented as a balance in a bitcoin account. A **bitcoin account** is defined by an Elliptic Curve Cryptography key pair. The bitcoin account is publicly identified by its **bitcoin address**, obtained from its public key using a unidirectional function. Using this public information users can send bitcoins to that address[1]. Then, the corresponding private key is needed to spend the bitcoins of the account.

2.1 Bitcoin Payments

Payments in the bitcoin system are performed through transactions between bitcoin accounts. A **bitcoin transaction** indicates a bitcoin movement from source addresses to destination addresses. Source addresses are known as **input addresses** in a transaction and destination addresses are named **output addresses**. As it can be seen in Fig. 1, a single transaction can have one or multiple input addresses and one or multiple output addresses.

A transaction details the exact amount of bitcoins to be transferred from each input address. The same applies to the output addresses, indicating the total amount of bitcoins that would be transferred at each account. For consistency, the total amount of the input addresses (source of the money) must be greater or equal than the total amount of the output addresses (destination of the money). Furthermore, the bitcoin protocol forces that input addresses must spend the

[1] Notice that the terms public key, address or bitcoin account refer to the same concept.

Inputs

Previous output (index)	Amount	From address	Type	ScriptSig
e631567f352f...:1	3.02887912	1CGVyAgAx9gg1va5pGNVJtF6gdKpPUVTSf	Address	304402201700305a3d79a[....]2b985b15daa0ab9c50cd61449ca037dc9f0
c284ec14325f...:0	3.04042789	1GY84QPLfM9d4KqTjTbbHsb9BX9FF1kYQx	Address	3045022100e724004f2d3[....]91d95b56ad29f817f3e3259daffbd72f2a98
0fbec1d29b8e...:0	2.99934316	1CGVyAgAx9gg1va5pGNVJtF6gdKpPUVTSf	Address	304402200f6e9b4281cb0[....]2b985b15daa0ab9c50cd61449ca037dc9f0
232715b3c51a...:1	3.00515088	17ALqzZFPbSqXz9aQhzgK6ts9htZfV8Mwu	Address	304402207311495478c1d[....]8d4656bf7613d47dd4e6a5b062d9fb6a34

Outputs

Index	Amount	To address	Type	ScriptPubKey
0	0.51682435	1LUHXNTsHPUGVJJeefPdb2rpdxtWoHrcKv	Address	OP_DUP OP_HASH160 d5936a017660c48be2adaa9a77153eccfdb8b0b8 OP_EQUALVERIFY OP_CHECKSIG
1	11.5569767	1HzAb4E1kZH4pDKoxML4KXBLPPyUootw4s	Address	OP_DUP OP_HASH160 ba51b9aee7595c72a2cbc1d4e3e90e356f77804 OP_EQUALVERIFY OP_CHECKSIG

Fig. 1. Bitcoin transaction example: four input addresses and two output addresses (data from blockexplorer.com).

exact amount of a previously received transaction[2] and, for that reason, in a transaction each input must unambiguously indicate the transaction[3] and the index of the output from which the bitcoins were received (the field *Previous output (index)* in Fig. 1).

Finally, the owner of the input addresses should perform a digital signature using his private keys, proving that he is the real owner of such accounts.[4]

Before accepting a payment from a standard transaction, the receiver should:

- Validate that the bitcoins of the input addresses are not previously spent.
- Validate that the digital signature is correct.

The first validation prevents doublespending in the bitcoin system and it is performed through a ledger where all previous transactions are annotated. Before accepting the payment, the receiver needs to be sure that there is no other transaction already in the ledger that has an input with the same *Previous output (Index)*. For that reason, the integrity of the system is based on the fact that this ledger is not modifiable, although it should be possible to add new transactions. In the bitcoin system, this append-only ledger is called blockchain.[5] The second validation can be performed with the information included in the transaction itself (field *ScriptSig*) together with the information of the transaction identified in the *Previous output (Index)* (field *ScriptPubKey*).

[2] Notice that in Fig. 1, there are two input addresses that are exactly the same which indicates that bitcoins have arrived to this bitcoin account in two separate transactions.

[3] A transaction is identified in the bitcoin system by its hash value.

[4] Although this is the standard form of bitcoin verification for regular bitcoin transfer transactions, the verification of a transaction can be much more complex and is based on the execution of a stack-based scripting language (more details can be found in Chap. 5 of [2]).

[5] Note that the non-modifiable property of the blockchain implies that bitcoin payments are non reversible.

2.2 The Blockchain and the Mining Process

The **blockchain** is a general append-only ledger containing all bitcoin transactions performed since the system started to operate, back in 2009. Such approach implies that the size of the blockchain is constantly increasing and, for that reason, scalability is probably the biggest challenge that the system faces. The blockchain is freely replicated and stored in different nodes of the bitcoin network, making the bitcoin a completely distributed system.

Transactions are included in the blockchain at time intervals, rather than in a flow fashion, and such addition is performed by collecting all new transactions of the system, compiling them together in a data structure called block, and including the block at the top of the blockchain. Every time that a block containing a specific transaction is included in the blockchain such transaction is said to be a **confirmed transaction** since it has been already included in the blockchain and can be checked for doublespending prevention.

Blocks are data structures that mainly contain a set of transactions that have been performed in the system (see Fig. 2). To achieve the append-only property, the inclusion of a block in the blockchain is a hard problem, so adding blocks to the blockchain is time and work consuming. Furthermore, every block is indexed using its hash value and every new block contains the hash value of the previous one (see the field *Previous block* in Fig. 2). Such mechanism ensures that the modification of a block from the middle of the chain would imply to modify all remaining blocks of the chain from that point to the top in order to match all hash values.

Block 125552

Hash: 00000000000000001e8d6829a8a21adc5d38d0a473b144b6765798e61f98bd1d
Previous block: 00000000000008a3a41b85b8b29ad444def299fee21793cd8b9e567eab02cd81
Time: 2011-05-21 17:26:31
Difficulty: 244 112.487774
Transactions: 4
Total BTC: 84.52
Size: 1.496 kilobytes
Merkle root: 2b12fcf1b09288fcaff797d71e950e71ae42b91e8bdb2304758dfcffc2b620e3
Nonce: 2504433986

Transactions

Transaction	Fee	Size (kB)	From (amount)	To (amount)
51d37bdd87...	0	0.135	Generation: 50 + 0.01 total fees	15nNvBTUdMaiZ6d3GWCeXFu2MagXL3XM1q: 50.01
60c25dda8d...	0	0.259	1HuppjXz7dPrt2a67LqacDW5T4VanFrpqC: 29.5	1B8vkT58i8KUPVJvvyQfrbc8Wjwu3vEarQ: 0.5 1BQbxzgRSLEsmv1JNc8MG76wdUgMwbsaww: 29
01f314cdd8...	0.01	0.617	1NdzSE6sHubscXJrv7jJn2gd4fL9L3ai6E: 0.03 1Ijv9m5VrRUE7VoktCsj18KUSqkqchhbum: 0.02 1HsYJJPqTn34DEjMnTb3VfKckX7ZcWPibm: 4.82	175FNxcLc1YrTwwG6TcsywcsHYdVqyhbwC: 0.01 1MueNMRJmcqVQeqE7v4dqogpNbhyxqq8R6: 4.85
b519286a10...	0	0.404	12DCoCVvDCkQShZ5RTh9bysgCkmkRMNQbT: 0.14 13CJwnnXJPwkzY4Xnaoqf8dnyNBwrHG9fe: 0.01	1Mos7p8fqJKBcYNRG1TdT5hBRxdMP6YHPy: 0.15

Fig. 2. Example of a bitcoin block (data from blockexplorer.com).

Adding a block to the blockchain is known as the **mining process**, a process that is also distributed and that can be performed by any user of the bitcoin network using specific-purpose software (and hardware). The mining process uses a hashcash proof-of-work system, first proposed by Adam Back as an anti-spam mechanism [3]. The proof-of-work consists in finding a hash of the new block with a value lower than a predefined target[6]. This process is performed by brute force varying the nonce value of the block. Once the value has been found, the new block becomes the top block of the blockchain and all miners discard their work on that block and move to the next one.

Mining new blocks is a structural task in the bitcoin system since it helps to confirm the transactions of the system. For that reason, and also assuming that mining implies a hard work, miners have to be properly rewarded. In the bitcoin system, miners are rewarded with two mechanisms. The first one provides them with newly created bitcoins. Every new block includes a special transaction, called **generation transaction** or coinbase transaction (see the first transaction in Fig. 2), in which it does not appear any input address and the output address is determined by the miner who creates the block, who obviously indicates one of its own addresses.[7] The second rewarding mechanism is the fees that each transaction pays to the miner. The fee for each transaction is calculated by computing the difference between the total input amount and the total output amount of the transaction (notice that in example block of Fig. 2 the first transaction does not provide any fee while the second one generates a 0.01 fee). All fees collected from transactions in a block are included in the generation transaction.

2.3 The Bitcoin Network

The bitcoin system needs to disseminate different kinds of information, essentially, transactions and blocks. Since both data are generated in a distributed way, the system transmits such information over the Internet through a distributed peer to peer (P2P) network. Such distributed network is created by bitcoin users in a dynamic way, and nodes of the bitcoin P2P network [4] are computers running the software of the bitcoin network node. This software is included by default into bitcoin's full-client wallets, but it is not usually incorporated in light wallet versions, such as those running in mobile devices. It is important to stress such distinction in case to perform network analysis, because when discovering nodes in the P2P bitcoin network, depending on the scanning techniques, not all bitcoin users are identified, but only those running a full-client and those running a special purpose bitcoin P2P node. Furthermore, online bitcoin accounts,

[6] Notice that the value of the target determines the difficulty of the mining process. Bitcoin adjusts the target value depending on the hash power of the miners in order to set the throughput of new blocks to 1 every 10 min (in mean).

[7] The amount of a generation transaction is not constant and it is determined by the bitcoin system. Such value, started in 50 bitcoins, is halved every four years, fixing asymptotically to 21 millions the total number of bitcoins that will ever be created.

provided by major bitcoin Internet sites, can also be considered as light weight clients, so they do not represent a full bitcoin P2P node neither.

2.4 Bitcoin Scalability Issues

Recently, the increase of both the popularity and the usage of the bitcoin system has shown its bounds regarding its ability to scale with the number of users. It is obvious that a system with a unique (although replicated) register containing all system transactions (i.e. the blockchain) may present a bottleneck.

Scalability issues can be measured in different ways as it is pointed out in [5]. From latency (the time for a transaction to be confirmed) to bootstrap time (the time it takes a new node to download and process the history necessary to validate payments) through cost per confirmed transactions, different measures can be used to evaluate the efficiency of a payment system. Croman et al. [5] give approximations of all of these metrics for the Bitcoin network. However, probably the easiest measure to compare Bitcoin with existing global payment systems and the one that has a direct impact on scalability is the system transaction throughput. The transaction throughput can be measured by the maximum number of transactions per second that a system may deal with and it is often chosen to evaluate systems because it is objective, easy to compute, and can be used to compare different payment systems easily. For instance, Visa reported to allow around 2,000 transactions per second in normal situation [5] while reaching a peak of 56,000 in a stress test [6]. Paypal manages lower values, providing 136 transactions per second as mean throughput on his payment network.[8]

Bitcoin throughput can be measured taking into account different parameters, from network communication latency to processing power of the nodes performing transaction validation. However, the restriction that limits the most the throughput of the system is the maximum size of blocks. Currently (June 2016), block size is fixed at a maximum value of 1 MB.[9] Yet this limit has not always been in place: the initial release of the code in February 2009 did not explicitly contain a block size limit and it was not until late 2010 when the 1 MB limit was enforced. The procedure for activating the block size limit was gradual: first, the core was changed so that no large blocks were mined;[10] second, the consensus rules were updated to reject blocks larger than 1 MB;[11] finally, the new rules started to be enforced on block height higher than 79,400 (which was reached[12] in September 12^{nd}, 2010). From that moment on, the block size limit has been kept to 1 MB.

[8] PayPal Q1 2016 Results [7] reported handling 1.41B payment transactions, which leads to an estimated $1.41B/4/30/24/60/60 = 136$ transactions per second.

[9] https://github.com/bitcoin/bitcoin/blob/a6a860796a44a2805a58391a009ba22752f64e32/src/consensus/consensus.h#L9.

[10] https://github.com/bitcoin/bitcoin/commit/a30b56ebe76ffff9f9cc8a6667186179413cc6349.

[11] https://github.com/bitcoin/bitcoin/commit/8c9479c6bbbc38b897dc97de9d04e4d5a5a36730#diff-118fcbaaba162ba17933c7893247df3aR1421.

[12] https://blockchain.info/block-height/79400.

Limiting the size of blocks to 1 MB implies a maximum throughput of 7 transactions per second [5]. The 7 transactions per second limit is an approximation obtained by dividing the maximum size of blocks by the average size of Bitcoin transactions (250 bytes) and the inter-block time (10 min). Therefore, a block of maximum size may contain 1,000,000/250 = 100 average sized transactions, thus giving a throughput of 100/600 = 6.6 transactions per second. Notice that such value is very far from the numbers that other payment systems, like Visa or PayPal, may deal with.

3 Bitcoin Scalability Proposals

Modification proposals in the Bitcoin core protocol, even those of utter importance like the ones affecting the scalability of the system, are often difficult to tackle since they have to be deployed with extreme precaution and maximum consensus. Furthermore, if changes affect the consensus mechanisms of the protocol, their implications may cause a blockchain fork and that could have a big impact in a cryptocurrency with a market capitalization of more than 11.5 billion dollars.[13] Moreover, the collateral implications of changes need to be also considered beforehand to prevent unexpected consequences, specially those related to security and decentralization.

Changes in the Bitcoin consensus rules may be introduced by soft (protocol) forks or hard (protocol) forks. A soft fork is produced when the protocol rules are changed so that the new rules are more strict than the old rules. In this case, all blocks accepted by the new rules will also be recognized as valid by the old rules. On the contrary, hard forks make the protocol rules less strict. Therefore, all blocks accepted by the old rules will also be valid by the new rules but there may be blocks that are valid with the new rules that were invalid with the older rules. Soft forks are preferred for updating rules because they do not break compatibility with previous versions and they do no require all participants to upgrade [8].

The effects of hard and soft forks on the network are also different. As an example, let's consider the case where 95 % of the mining power of the network upgrades to a new set of rules. In a hard fork, the upgraded 95 % will eventually create a block which is valid under the new rules but invalid following the old rules. From that moment on, a (blockchain) fork will remain in the network: the upgraded clients will consider the block valid and will keep mining on top of it, whereas the non-upgraded 5 % will recognize the block as invalid and discard it together with all subsequent blocks. The non-upgraded 5 % will always consider the block invalid, and thus will create an alternative branch of the chain and will remain in that branch, making the (blockchain) fork persistent. On the contrary, in a soft fork, most of the blocks will be mined by the upgraded nodes (since they have 95 % of the mining power) and they will be accepted by all nodes (regardless of their upgrading status). Sooner or later, one of the non-upgraded miners will create a new block which will be seen as valid by the 5 % of the

[13] Information from http://coinmarketcap.com/ on June 17th, 2016.

miners but invalid for the rest. As a consequence, 5 % of the mining power will start mining on top of that new block, and the 95 % left will keep mining at the same height. Since the upgraded nodes have the majority of the mining power, their branch will soon be longer than the branch created by the non-upgraded miner. Seeing that this branch is longer and valid, the non upgraded nodes will change to the upgraded branch, and thus all the network will be mining again on the same branch.

In the next subsections, we review the techniques that have been proposed to boost the scalability of bitcoin. It is worth mention that since in this paper we are focused on the bitcoin system, we do not consider those scalability proposals that are envisaged for general decentralized blockchain systems, like the ones proposed Croman et al. [5], but could not be applied to the bitcoin system due to the impractical solutions to redefine some primitives, like modify the proof-of-work protocol.

3.1 Tuning Bitcoin Protocol Parameters

Tuning protocol parameters may allow Bitcoin to improve its scalability, although previous studies have concluded that high scalability in the longer term requires a protocol redesign [5].

The parameter that has been most discussed by the Bitcoin community in order to improve system scalability is the block size limit. Some proposals suggest to increase the limit following different strategies or even propose to remove the limit. Jeff Garzik's BIP 100 [9] proposed to change the 1 MB fixed limit to a new floating block size limit, where miners may increase the block size by consensus. Gavin Andresen BIP 101 [10] proposal (currently withdrawn) consisted in initially increasing block size to 8 MB and doubling the size every two years for 20 years, after which the block size remains fixed. Jeff Garzik's BIP 102 [11] proposes to simply increase block size to 2 MB. Pieter Wuille's BIP 103 [12] proposed to increase the maximum block size by 4.4 % every 97 days until 2063, implying a 17.7 % block size increase per year. Gavin Andresen's BIP 109 [13] propose a fixed block size increase to 2 MB with a new limit on the amount of data that can be hashed to compute signature hashes and a change on how the maximum number of signatures is counted. All of these proposals have to be deployed via a hard fork, since all blocks bigger than 1 MB will be seen as invalid by the current version nodes.

However, the increase on the block size limit can not be done arbitrarily. Recent studies [5] argued that with the current 10 min average block interval and taking into account block propagation times in the network, block size limit should not be increased to more than 4 MB.

Segregated Witness [14] is another proposal that does not increase the block size limit, but reduces the amount of information stored per transaction, thus effectively allowing more transactions per block. Additionally, segregated witness solves the malleability problem (refer to Sect. 3.2 for a description) and introduces many other benefits.

3.2 Off-Chain Payment Channels

It is unclear whether tuning the protocol parameters alone will provide enough scaling benefits to satisfy bitcoin needs in the future. For that reason, one of the proposals that has been broadly accepted in the bitcoin community as a relevant bitcoin scalability solution is an improvement that has been enumerated in the previous section: the segregated witness approach. As a single proposal, segregated witness only provides, in the best case, a 4x increase in throughput of the bitcoin network, falling in the buy-time-now solution for the bitcoin scalability problem. But the segregated witness ability to resolve the transaction malleability problem allows to develop new mechanisms that could provide a much more powerful tool for bitcoin scalability issues: off-chain payment channels.

Transaction Malleability Problem. As we pointed out in Sect. 2.1 Bitcoin transactions are identified with its hash value, a value computed using a double SHA256 function over the raw data that defines the transaction. However, for space considerations, this identifier is not stored in the blockchain. Since signatures are not performed over all transaction data, after its creation a transaction can be modified adding some irrelevant data, resulting in a slightly different transaction but with a completely different identifier. Notice that, in this case, we will have two different valid transactions with different identifiers, and only after the transaction is included in the blockchain, the final identifier of the transaction will become unique. It is important to mention that such a modification, that provides malleability, does not affect the ability of an attacker to spend/steal the bitcoins present in the transaction inputs (since the attacker cannot perform the digital signature of the owner). The attacker is only able to modify the identifier of the transaction in a value that differs from the one its real owner has established. For that reason, although transaction malleability is known back from 2011, it has never been considered as a security issue.

Nonetheless, transaction malleability supposes a problem for smart contracts when a child transaction wants to spend a parent output before the parent transaction appears on the blockchain. In case that a malleabled parent transaction is finally included in the blockchain, then all pre-signed child transactions would be invalid.

Basic Off-Chain Payment Channel Ideas. Off-chain Payment Channels are mechanisms that allow payments between two parties, A and B, payments that can be performed without including a transaction for the payment itself in the blockchain.

The first proposal of such a mechanism was first targeted at micropayments from one payer to one payee. Its main goal was to avoid the fees that transactions in the blockchain imply and that are not affordable for micropayment transactions [15]. To set up the payment channel, a transaction is included in the blockchain as a deposit of the money that will be used in the payment channel. A refund transaction is also created, allowing the payer to recover the

deposited funds if the payee does not cooperate. The refund transaction can not be included in the blockchain until a certain point in the future, and thus the channel may remain open until that moment arrives. Between the set up and the closing of the payment channel, the payer can perform multiple payments to the payee through transactions that, although formatted in standard bitcoin format, would be transferred privately between A and B without using the standard bitcoin P2P network. Furthermore, the individual payment transactions will not appear in the blockchain: only the set up transaction that opens the channel and the last transaction that closes the channel will be broadcast through the bitcoin P2P network and will be included in the blockchain.

The channel can be closed at any time by B by signing and broadcasting the last transaction received from A. If A has never sent a transaction to B or B does not cooperate, A can get back her funds using the refund transaction, but she will have to wait until the transaction is valid as specified by the time lock. Moreover, if all the funds deposited in the channel by A have already been transferred to B, the channel is exhausted and can no longer be used. In that case, B can sign and broadcast the last transaction received from A, which transfers the whole amount of the channel to B and closes the channel.

In order to create the described unidirectional micropayment channel two bitcoin features are used: multisignature outputs and transactions with lock time.

Multisignature outputs are transaction outputs that may require more than one signature to unlock. For instance, two signatures may be required to unlock a single output. Multisignature outputs are used in the set up transaction of the basic micropayment channel explained above in order to lock the funds that are being used by the channel. In the set up transaction, the payer deposits a certain amount of bitcoins in the channel by sending that amount of bitcoins to a multisignature output controlled by both the payer and the payee.

Bitcoin transactions may have a time lock specifying either a unix timestamp or a blockchain block height. Transactions with a time lock can not be included in any block until the specified time has arrived. Time locks are used in micropayment channels as a mechanism that allows to replace transactions. A certain transaction with a time lock can be replaced by creating a new transaction with a smaller time lock spending the same outputs (or some of those outputs). The new transaction can then be broadcast sooner (because it has a smaller time lock) and thus replaces the old transaction. Note that, in order to effectively replace the transaction, the interested party must broadcast the new transaction to the network before the older one becomes valid. In the basic micropayment channel described above, a time lock is placed on the refund transaction to ensure that it can not be used to return all the money to the payer before the channel extinguishes. Payment transactions spend the same outputs and do not have a time lock, so they can replace the refund transaction.

Such basic approach is restricted to a unidirectional channel between A and B, allowing A to perform off-chain payments to B but without the ability for B to pay to A. The straightforward approach to generate bidirectional channels is

to create two unidirectional channels, one from A to B and another from B to A. The problem with such approach is that both channels are independent and if one of the channels runs out of money (suppose the channel $A \rightarrow B$) no more payments can be performed from A to B even if in the other payment channel ($B \rightarrow A$) A has a positive balance with B.

To construct bidirectional off-chain payment channels without this restriction, two different schemes have been proposed: duplex micropayment channels and lightning channels. In the following paragraphs, we provide a high level overview of both proposals.

Duplex Micropayment Channels. Duplex micropayment channels (DMC) are proposed by Decker and Wattenhofer [16]. DMC are able to provide bidirectional payments between two entities within a finite time frame. The main idea behind DMC is indeed to create two unidirectional channels between the two parties A and B as described before, but using a technique that allows to reset the channels when needed and thus effectively overcoming the problem of exhausting the funds in one of the channels while having a positive amount of bitcoins in the other. Therefore, the main contribution of the proposal is the technique that allows to reset the unidirectional channels: the invalidation tree.

The invalidation tree is a tree structure of depth d made of transactions with multisignature outputs. Each transaction in the tree has a time lock such that any two transactions spending coins from the same output have different time locks and the time lock of a children transaction is at least the same of the time lock of the parent transaction. A branch of the tree is thus a set of d transactions of (non-strictly) increasing time lock. At any given moment, only one branch of the tree is valid, while the other branches are effectively replaced because of the time lock.

The unidirectional channels are then build on top of a leaf of the invalidation tree and are operated in the same way than the basic channels explained previously in this section. User A pays to user B using the $A \rightarrow B$ channel while user B pays to A using the $B \rightarrow A$ channel independently. However, when one of the channels is exhausted and the sender has funds in the other channel, a reset of the channels can be triggered so that these funds become available to spend. In order to do so, A and B create a new branch of the invalidation tree that replaces the currently active branch, and new unidirectional channels are appended to the leaf of the new branch.

DMC have a finite time frame defined by the time locks used in the invalidation tree. Moreover, DMC also have a finite number of available channel resets, which can be freely determined by the depth of the invalidation tree and the time locks used.

Lightning Channels. Lightning channels are proposed by Poon and Dryja [17] and are able to create bidirectional channels without any time limitations, that is, channels that can remain open until any of the parties decides to close them.

Unlike DMC, lightning channels do not create unidirectional channels: they create a unique channel that allows to send bitcoins in both directions.

In order to do so, for each new payment the two users agree on the current balances of the channel and create two transactions that represent these new balances (both transactions represent the same balances). One transaction is then kept by each user, allowing her to sign it and broadcast it to the network and thus closing the channel in its current state. These transactions have some particularities. The transaction that A can broadcast to the network sends immediately to B his amount, and prevents A from getting her share until after some blocks have been created on top of the block including the transaction. During this time frame, B can also claim A's amount if he reveals a secret that only A knows. Similarly, the transaction that B keeps sends bitcoins immediately to A and prevents B from getting his amount until some blocks, and during this time A can claim B's amount if she reveals a secret than only B knows.

Whenever a new payment has to be made in the channel, both users update the balances and agree to the new state of the channel: they create two transactions with the new balances, keeping again each one of the transactions. Now, both users can broadcast their transaction to the network and thus secure their balances. However, at this point there is nothing preventing the users from broadcasting their transactions from the previous state of the channel. In order to ensure that none of the users cheat by broadcasting old transactions, at every new transaction the users exchange their respective secrets for the previous transaction. Now, if one of the users tries to cheat by broadcasting an old transaction, the other party can claim all the funds of the channel by revealing the secret.

Off-Chain Payment Networks. Duplex micropayment channels and lightning networks as described above provide a mechanism to stablish bidirectional channels between two different users. However, it is impractical that users will open a new off-chain payment channel with a counterpart unless the number of payments between both parties is high. To overcome such problem, both proposals allow an improvement by which two-side off-chain payment channels can be somehow concatenated in order to allow users to perform payment through multiple established off-chain channels without the parts needing to trust the intermediary ones. The idea to implement this feature uses the ability to spend a transaction once a secret value is known. In Fig. 3, a single hop example is showed. C, who receives the payment, generates a random value, x, and computes its hash, $h(x)$, that will send to A. A creates a transaction $Tx_1(B, h(x))$ and sends it to B. B can charge the transaction providing the value x, that he does not have. To obtain x, B creates another transaction $Tx_2(C, h(x))$ that can be charged by C when he provides x. Since C does indeed know the value x, when he reveals it he can charge the second transaction and B can charge the first one.

Notice that with this scenario, the payment between users A and C can be performed not only in case A and B have a direct payment channel but also when there is a path through multiple payment channels that link them. Based on this

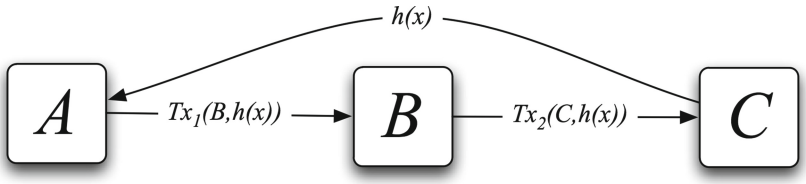

Fig. 3. One hop off-chain payment channel.

principle, it is straight forward to envisage the appearance of such intermediary nodes, the payment service providers, nodes that will create a highly connected network that will route and perform such off-chain payments.

4 Privacy Implications of Scalability Solutions

Anonymity is probably one of the properties that has contributed to the success of bitcoin deployment. Anonymity in the bitcoin network is based on the fact that users can create any number of anonymous bitcoin addresses that will be used in their bitcoin transactions. This basic approach is a good starting point, but the underlaying non-anonymous Internet infrastructure, together with the availability of all bitcoin transactions in the blockchain, has proven to be an anonymity threat. In [18], research performed on bitcoin privacy is categorized in three main areas: Blockchain analysis, traffic analysis and mixing networks. Next, we review how the different ideas proposed in those areas would be affected by the implementation and adoption of off-chain payment channels.

4.1 Blockchain Analysis

A direct approach to analyze the anonymity offered by the bitcoin system is to dig information out of the blockchain. A simple analysis provides information about the movements of bitcoins: from which bitcoin addresses the money comes and to which bitcoin addresses it goes. However, such basic approach has two main drawbacks:

- Users with multiple addresses: since users in the bitcoin system can create any number of addresses, in order to obtain insightful information from the blockchain researchers try to cluster all addresses that belong to the same user. As we will see, authors apply different techniques to perform such clustering.
- Blockchain data volume: at present time, the size of the blockchain data is 72 GB. However, such data only include the raw blockchain data, which is not a database and it is not suitable for data queries. Storing such information in a searchable database expands the data size and produces a much larger database which is more difficult to deal with. For that reason, research papers on privacy issues regarding blockchain analysis have drastically decreased in the last years.

Furthermore, a basic assumption is made on the blockchain information when doing such analysis: the blockchain includes all transactions of the system. But, in the light of off-chain payment channels, such assumption is no longer valid since only a fraction of transactions are finally stored in the blockchain.

Address clustering has been one of the blockchain analysis techniques used to deanonymize users in the bitcoin networks. The idea is to cluster different bitcoin addresses belonging to the same user in order to trace his economic movements. Different heuristics have been used to perform such clustering. A common assumption is to consider that all addresses included as inputs in a transaction belong to the same user [19–21]. Another technique to cluster addresses is to consider that in a transaction with multiple outputs, in case that one of them goes to a not previously used addresses (and the others use addresses already appeared in the blockchain) the new address can be clustered with all input addresses of such transaction [20,22].

Notice that those techniques cannot be effectively used when off-chain payments channels would become common use. First of all, the assumption that all inputs from a single transaction belong to the same user no longer holds since set up transactions for payments channels include two inputs from exactly two different users. Furthermore, a portion of transactions in the blockchain with exactly two outputs would come from closing payment channels, for which no assumption can be performed between input and output addresses. At most, transactions closing payment channels can be linked with set up transactions since both share the same address, but since input addresses from set up channels will be different from output addresses from closing channels it will not be possible to infer how much money each address have been spent/earned as a consolidate balance of the payment channel.

On the other hand, with the adoption of payment channels, the size of the address cluster for a typical user will hardly be reduced. Notice that without payment channels, users are free to use a new address for every single operation performed (paying, cashing or taking the change). Such amount of addresses hardens the possibility to obtain a single cluster for each user. However, once a payment channel is opened the user performs all payments through such channel without involving any new address. Nevertheless, all those payments are performed off-chain so they cannot be traced by blockchain analysis techniques.

User anonymity has been also analyzed using k-anonymity measures. Ober *et al.* [23] indicate that to estimate the level of k-anonymity provided by bitcoin it is necessary to estimate the number of active entities since, for instance, dormant coins (those included in an address not active for a long time) reduce the anonymity set. Furthermore, they also indicate that to better estimate the k-anonymity at a certain point of time, active entities should be defined based on a window time around this period (hours, days, weeks, ...). Then, an active entity is the one that has performed a payment within this time window. With off-chain payment channels, entities activity is very hard to estimate since once a channel is opened, it cannot be determined by blockchain analysis if such channel is an active one or not for the obvious mechanism of off-chain communication.

Finally, tools like BitIodine [24] are based on mining the blockchain information assuming that all transactions performed by the system are included. With the off-chain payment channels, such assumption is no longer valid and, depending on the degree of its adoption, such a tool will not be able to provide significant information.

4.2 Traffic Analysis

As we already mentioned, the anonymity degree of users in the bitcoin system is also bounded by the underlying technologies used. Transactions in the bitcoin system are transmitted through a P2P network, so the TCP/IP information obtained from that network can be used to reduce the anonymity of the system, as it is pointed out in [19]. Although it is true that most wallets are able to work over TOR anonymous network,[14] a high number of bitcoin users do not use such services, and then, there is still room for network analysis. Moreover, using TOR to obtain anonymity while using bitcoin may not be the best choice [25].

Koshy *et al.* [26] performed an anonymity study based on real-time transaction traffic collected during 5 months. For that purpose, authors develop CoinSeer, a bitcoin client designed exclusively for data collection. For more than 5 million transactions, they collected information on the IP address from where the CoinSeer received such transaction and, in the general case, they assigned as the IP corresponding to the transaction the one that broadcast the transaction for the first time. In order to perform a pure network analysis, authors do not apply any address clustering process, so only single input transactions (almost four million) are taken into account in the analyzed data set. Then, to match an IP with a bitcoin address, they consider a vote on the link between IP_i and $address_j$ if a transaction first broadcasted form an IP_i contains the bitcoin $address_j$ as input address. Although Koshy *et al.* could not provide positive results for deanonymizing users, the techniques they propose were interesting and could be a threat to user privacy when more data is available for the analysis. Note that there already exist some actors in the current bitcoin scenario that are able to collect and process this kind of data. For instance, some blockchain explorers keep and publicly show information on the IP address from which they first receive transactions and blocks.

However, new off-chain payment channels present a new scenario since, as we already explained in Sect. 3.2, off-chain payments use a separate network. Although nodes of the channel need to be connected to the bitcoin network for channel set up and also to monitor the correctness of channel counter-party, communication of the channel will pass through a different network, the one that connects users from different off-chain payment channels through their payment service providers. Users will access the off-chain payment networks through a single node[15] for which the user will maintain an open payment channel. So a

[14] https://www.torproject.org/.

[15] It is difficult to predict at present time whether users will maintain multiple payment channels with multiple payment service providers but multiple channels could be not viable depending on the fees needed to open and close those channels.

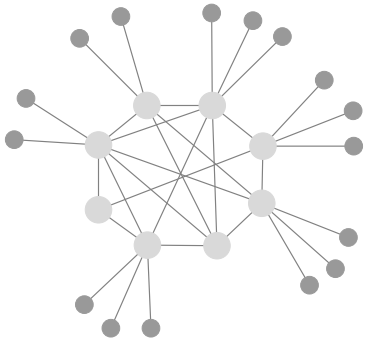

Fig. 4. Possible off-chain payment channel topology.

possible network topology for the network would be similar to the one showed in Fig. 4, with a highly connected component which include the payment service provider nodes and final users connected to those nodes.

This architecture differs from the distributed topology of the bitcoin network, where every user is able to maintain multiple simultaneous network connections with different nodes of the network to obtain some security features.

Notice that with this type of topology, transaction anonymity is lost when the payment between two users is performed through off-chain payment channels that only include a single payment service provider, since the entity providing such off-chain channels may know both, source and destination of the payment.

4.3 Mixing

In the bitcoin environment, mixing is used to anonymize bitcoins. Mix services shuffle the inputs and outputs of a transaction in order to hinder the relation between them. The goal is to allow bitcoin users to send bitcoins from one address to a mix service and receive from the mix service the bitcoins to another address that can not be linked with the original one. Therefore, the process detaches the link between a source (the input address) and a destination (the output address). Although such mixing service can be implemented straightforward using a central authority which receives payments and pays back to different addresses, the trusted level of such central authority would be too high. For that reason, different proposals that avoid or reduce the trusted role of the central authority have been presented.

A basic mix service can be implemented using a multiple-input and multiple-output transaction, as it is described in CoinJoin [27]. The idea is that multiple users can jointly create a transaction with multiple input addresses and multiple output addresses. To be a valid transaction, the transaction should be signed by all users participating in the mixing. One of the problems of this proposal (and to some extent of the majority of mixing proposals) is that one of the anonymous users of the mix service can perform a DoS attack. Since the final

valid transactions should be signed by all users that include bitcoins in the transactions, each mixing transaction never becomes valid in case the attacker simply does not sign any transaction in which he takes part.

In [28], Bonneau *et al.* present Mixcoin, a more sophisticated centralized mixing system that relies on accountability. Users of the system obtain, prior the mixing phase, a signed warranty that can be used to prove, in case of the event, that the mixer entity has misbehaved. Authors point out that such public verifiable proof of misbehavior would discourage malicious mixing. Furthermore, to reduce the possibility that the mixer could deanonymize users using his stored information, the authors propose a concatenation of several mixer services, thus reducing the strategy of a malicious mixer to a collusion with the other mixers.

Mixing services as described so far can be applied to standard bitcoin addresses and transactions but when off-chain payment channels are used, such mixing techniques cannot be applied in the same form. The reason is that in the standard bitcoin model, payments can bee seen as one hope transactions that are visible by all participants (they appear in the blockchain). Fortunately, in this scenario every user can create multiple addresses without any cost. For that reason, standard mixing services use multiple new addresses to hinder identities since there is no way to allow payments through secret multiple hops, because each hope (a transaction, in fact) must be recorded publicly in the blockchain. Conversely, in off-chain payments, on one hand, users may be more restricted on the number of payment channels that they create (due to fee costs) but on the other hand, payments are processed with multiple hops through different payment service providers, and such hops could remain secret since they take place in the off-chain payment network and there is no need to store them. So the natural idea to detach the link between source and destination in this scenario is to perform payments through secret multiple hop routes.

At that point it is worth notice that, in off-chain payment networks, payment anonymity highly resembles standard communication anonymity where a path between source and destination is hidden to protect communication identification. For that reason, common onion routing techniques [29] could be applied to allow anonymous payments in a similar way than TOR network provides anonymous browsing. Nevertheless, in the same way that single TOR utilization does not guarantee that you are browsing the www anonymously (since, for instance, the browser configuration may reveal some details about your identity), details on the protocol for multi-hop off-chain payment channels will have to be carefully analyzed (when they are available) in order not to disclose the link between the source and the destination.

5 Conclusions

Bitcoin scalability is one of the relevant topics in the broad crytocurrency field since some limitations Bitcoin faces are common to all blockchain based cryptocurrencies. Different ideas have been proposed so far, being the segregated witness approach the one that most support has received. Segregated witness

has the potential to solve transaction malleability and once solved, bitcoin will have the ability to work with unsigned transactions and off-chain payment channels will be able to start working in practice. From that moment on, payment networks will be able to grow and flourish. Moreover, payment networks will have a life of its own, being able to operate on top of the existing Bitcoin protocol but with the freedom that off-chain transacting will offer. This is potentially one of the biggest changes the bitcoin ecosystem has ever seen and, as such, will have an impact on many bitcoin properties, among which users' privacy is included.

Once off-chain payment networks become main use, some Bitcoin premises on decentralization and openness of its payment routing, multiple address generation and full transaction disclosure through the blockchain may be modified. Research performed so far has proven that the way the system uses payment addresses may unveil some information from their owners, when all transactions performed by the system were freely available in the blockchain for analysis and transactions were published through open P2P networks. However, if most of the transactions occur off-chain, this kind of analysis will no longer be as effective as before. Depending on the final payment channels implementations, some of these techniques may be able to adapt to extract information about the existing channels. Nonetheless, we will have to wait until the payment networks are deployed to evaluate to what extent this kind of analysis remains feasible.

Acknowledgments. This work was partially supported by the Spanish Ministerio funds under grant MINECO/TIN2014-55243-P and FPU-AP2010-0078, and through the Catalan Government funded project AGAUR/2014-SGR-691.

References

1. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2008)
2. Antonopoulos, A.M.: Mastering Bitcoins. O'Reilly, Media (2014)
3. Back, A.: A partial hash collision based postage scheme (1997). http://www.hashcash.org/papers/announce.txt. Accessed June 2016
4. Donet Donet, J.A., Pérez-Solà, C., Herrera-Joancomartí, J.: The bitcoin P2P network. In: Böhme, R., Brenner, M., Moore, T., Smith, M. (eds.) FC 2014 Workshops. LNCS, vol. 8438, pp. 87–102. Springer, Heidelberg (2014)
5. Croman, K., Decker, C., Eyal, I., Gencer, A.E., Juels, A., Kosba, A., Miller, A., Saxena, P., Shi, E., Gün, E.: On scaling decentralized blockchains. In: Proceedings of 3rd Workshop on Bitcoin and Blockchain Research (2016)
6. Visa: 56,582 transaction messages per second!, July 2014. http://visatechmatters.tumblr.com/post/108952718025/56582-transaction-messages-per-second. Accessed June 2016
7. Paypal: Paypal q1 2016 fast facts, June 2016. Accessed June 2016
8. Core, B.: Bitcoin core statement, January 2016. https://bitcoincore.org/en/2016/01/07/statement/. Accessed June 2016
9. Garzik, J.: BIP 100: making decentralized economic policy (2015). Accessed June 2016
10. Andresen, G.: BIP 101: increase maximum block size (2015). Accessed June 2016
11. Garzik, J.: BIP 102: block size increase to 2MB (2015). Accessed June 2016

12. Wuille, P.: BIP 103: block size following technological growth (2015). Accessed June 2016
13. Andresen, G.: BIP 109: two million byte size limit with sigop and sighash limits (2016). Accessed June 2016
14. Lombrozo, E., Lau, J., Wuille, P.: BIP 141: segregated witness (consensus layer) (2015). Accessed June 2016
15. Hearn, M., Spilman, J.: Bitcoin contracts. https://en.bitcoin.it/wiki/Contract. Accessed June 2016
16. Decker, C., Wattenhofer, R.: A fast and scalable payment network with bitcoin duplex micropayment channels. In: Pelc, A., Schwarzmann, A.A. (eds.) SSS 2015. LNCS, vol. 9212, pp. 3–18. Springer, Heidelberg (2015)
17. Poon, J., Dryja, T.: The bitcoin lightning network: scalable off-chain instant payments. Technical report (draft). https://lightning.network (2015)
18. Herrera-Joancomartí, J.: Research and challenges on bitcoin anonymity. In: Garcia-Alfaro, J., Herrera-Joancomartí, J., Lupu, E., Posegga, J., Aldini, A., Martinelli, F., Suri, N. (eds.) DPM/SETOP/QASA 2014. LNCS, vol. 8872, pp. 3–16. Springer, Heidelberg (2015)
19. Reid, F., Harrigan, M.: An analysis of anonymity in the bitcoin system. In: Altshuler, Y., Elovici, Y., Cremers, A.B., Aharony, N., Pentland, A. (eds.) Security and Privacy in Social Networks, pp. 197–223. Springer, New York (2013)
20. Androulaki, E., Karame, G.O., Roeschlin, M., Scherer, T., Capkun, S.: Evaluating user privacy in bitcoin. In: Sadeghi, A.-R. (ed.) FC 2013. LNCS, vol. 7859, pp. 34–51. Springer, Heidelberg (2013)
21. Ron, D., Shamir, A.: Quantitative analysis of the full bitcoin transaction graph. In: Sadeghi, A.-R. (ed.) FC 2013. LNCS, vol. 7859, pp. 6–24. Springer, Heidelberg (2013)
22. Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G.M., Savage, S.: A fistful of bitcoins: characterizing payments among men with no names. In: Proceedings of the 2013 Conference on Internet Measurement Conference, IMC 2013, pp. 127–140. ACM, New York (2013)
23. Ober, M., Katzenbeisser, S., Hamacher, K.: Structure and anonymity of the bitcoin transaction graph. Future Internet **5**(2), 237–250 (2013)
24. Spagnuolo, M., Maggi, F., Zanero, S.: BitIodine: extracting intelligence from the bitcoin network. In: Christin, N., Safavi-Naini, R. (eds.) FC 2014. LNCS, vol. 8437, pp. 457–468. Springer, Heidelberg (2014)
25. Biryukov, A., Pustogarov, I.: Bitcoin over tor isn't a good idea. In: 2015 IEEE Symposium on Security and Privacy (SP), pp. 122–134. IEEE (2015)
26. Koshy, P., Koshy, D., McDaniel, P.: An analysis of anonymity in bitcoin using P2P network traffic. In: Christin, N., Safavi-Naini, R. (eds.) FC 2014. LNCS, vol. 8437, pp. 469–485. Springer, Heidelberg (2014)
27. Maxwell, G.: Coinjoin: bitcoin privacy for the real world. Post on Bitcoin Forum
28. Bonneau, J., Narayanan, A., Miller, A., Clark, J., Kroll, J.A., Felten, E.W.: Mixcoin: anonymity for bitcoin with accountable mixes. In: Christin, N., Safavi-Naini, R. (eds.) FC 2014. LNCS, vol. 8437, pp. 486–504. Springer, Heidelberg (2014)
29. Chaum, D.L.: Untraceable electronic mail, return addresses, and digital pseudonyms. Commun. ACM **24**(2), 84–90 (1981)