Bitcoin Forum

simple machines forum

April 20, 2020, 09:59:52 AM

Welcome, **Guest**. Please login or register.

News: Latest Bitcoin Core release: 0.19.1 [Torrent]

Search

HOME HELP SEARCH LOGIN REGISTER MORE

Bitcoin Forum > Bitcoin > Development & Technical Discussion > **CoinJoin: Bitcoin privacy for the real world**

« previous topic
next topic »
print

Pages: [**1**] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 »

Author Topic: CoinJoin: Bitcoin privacy for the real world (Read 291894 times)

gmaxwell
Moderator
Legendary

Activity: 3024
Merit: 3468

CoinJoin: Bitcoin privacy for the real world

August 22, 2013, 02:32:31 AM

#1

Merited by ETFbitcoin (17), monbux (10), fillippone (8), ebliever (5), Husna QA (2), OgNasty (1), smooth (1), morvillz7z (1), Financisto (1)

Bitcoin is often promoted as a tool for privacy but the only privacy that exists in Bitcoin comes from pseudonymous addresses which are fragile and easily compromised through reuse, "taint" analysis, tracking payments, IP address monitoring nodes, web-spidering, and many other mechanisms. Once broken this privacy is difficult and sometimes costly to recover.

Traditional banking provides a fair amount of privacy by default. Your inlaws don't see that you're buying birth control that deprives them of grand children, your employer doesn't learn about the non-profits you support with money from your paycheck, and thieves don't see your latest purchases or how wealthy you are to help them target and scam you. Poor privacy in Bitcoin can be a major practical disadvantage for both individuals and businesses.

Even when a user ends address reuse by switching to BIP 32 address chains, they still have privacy loss from their old coins and the joining of past payments when they make larger transactions.

Privacy errors can also create externalized costs: You might have good practices but when you trade with people who don't (say ones using "green addresses") you and everyone you trade with loses some privacy. A loss of privacy also presents a grave systemic risk for Bitcoin: If degraded privacy allows people to assemble centralized lists of good and bad coins you may find Bitcoin's fungibility destroyed when your honestly accepted coin is later not honored by others, and its decentralization along with it when people feel forced to enforce popular blacklists on their own coin.

As I write this people with unknown motivations are raining down tiny little payments on old addresses, presumably in an effort to get wallets to consume them and create evidence of common address ownership.

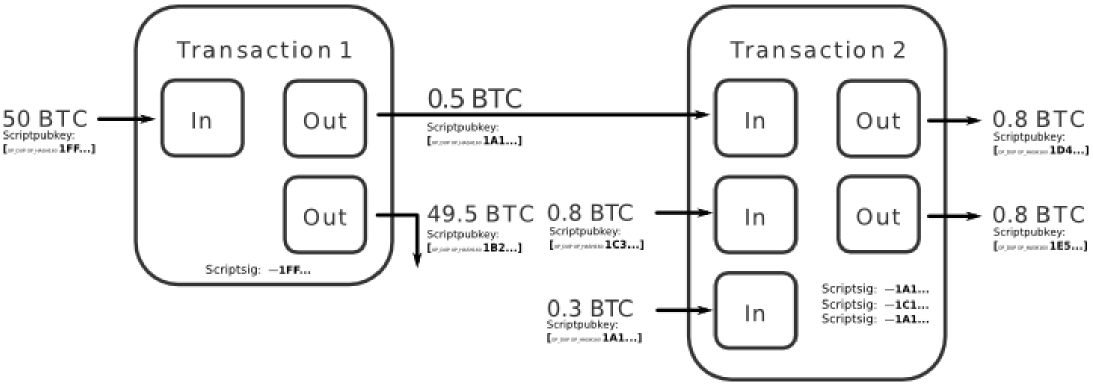
I think this must be improved, urgently.

This message describes a transaction style Bitcoin users can use to dramatically improve their privacy which I've been calling **CoinJoin**. It involves no changes to the Bitcoin protocol and has already seen some very limited use spanning back a couple of years now but it seems to not be widely understood.

I first publicly described this transaction style in a whimsically-named thread— "I taint rich!"— where I focused on a specific side effect of these transactions, with an expectation that people would see the rest of the implications on their own.

Explicit beats implicit, and even people who understand the idea have had some questions which could use answering. Thus this post.

The idea is very simple, first some quick background:



A Bitcoin transaction consumes one or more inputs and creates one or more outputs with specified values.

Each input is an output from a past transaction. For each input there is a distinct signature (scriptsig) which is created in accordance with the rules specified in the past-output that it is consuming (scriptpubkey).

The Bitcoin system is charged with making sure the signatures are correct, that the inputs exist and are spendable, and that the sum of the output values is less than or equal to the sum of the input values (any excess becomes fees paid to miners for including the transaction).

It is normal for a transaction to spend many inputs in order to get enough value to pay its intended payment, often also creating an additional 'change' output to receive the unspent (and non-fee) excess.

There is no requirement that the scriptpubkeys of the inputs used be the same; i.e., no requirement that they be payments to the same address. And, in fact, when Bitcoin is correctly used with one address per payment, none of them will be the same.

When considering the history of Bitcoin ownership one could look at transactions which spend from multiple distinct scriptpubkeys as co-joining their ownership and make an assumption: How else could the transaction spend from multiple addresses unless a common party controlled those addresses?

In the illustration 'transaction 2' spends coins which were assigned to 1A1 and 1C3. So 1A1 and 1C3 are necessarily the same party?

This assumption is incorrect. Usage in a single transaction does not prove common control (though it's currently pretty suggestive), and this is what makes **CoinJoin** possible:

The signatures, one per input, inside a transaction are **completely** independent of each other. This means that it's possible for Bitcoin users to agree on a set of inputs to spend, and a set of outputs to pay to, and then to individually and separately sign a transaction and later merge their signatures. The transaction is not valid and won't be accepted by the network until all signatures are provided, and no one will sign a transaction which is not to their liking.

To use this to increase privacy, the N users would agree on a uniform output size and provide inputs amounting to at least that size. The transaction would have N outputs of that size and potentially N more change outputs if some of the users provided input in excess of the target. All would sign the transaction, and then the transaction could be transmitted. No risk of theft at any point.

In the illustration 'transaction 2' has inputs from 1A1 and 1C3. Say we beliece

1A1 is an address used for Alice and 1C3 is an address used for Charlie. Which of Alice and Charlie owns which of the 1D and 1E outputs?

The idea can also be used more casually. When you want to make a payment, find someone else who also wants to make a payment and make a joint payment together. Doing so doesn't increase privacy much, but it actually makes your transaction *smaller* and thus easier on the network (and *lower* in fees); the extra privacy is a perk.

Such a transaction is externally *indistinguishable* from a transaction created through conventional use. Because of this, if these transactions become widespread they improve the privacy even of people who do not use them, because no longer will input co-joining be strong evidence of common control.

There are many variations of this idea possible, and all can coexist because the idea requires no changes to the Bitcoin system. Let a thousand flowers bloom: we can have diversity in ways of accomplishing this and learn the best.

FAQ:

Don't you need tor or something to prevent everyone from learning everyone's IP?

Any transaction privacy system that hopes to hide user's addresses should start with some kind of anonymity network. This is no different. Fortunately networks like Tor, I2P, Bitmessage, and Freenet all already exist and could all be used for this. (Freenet would result in rather slow transactions, however)

However, gumming up "taint analysis" and reducing transaction sizes doesn't even require that the users be private from each other. So even without things like tor this would be no worse than regular transactions.

Don't the users learn which inputs match up to which outputs?

In the simplest possible implementation where users meet up on IRC over tor or the like, yes they do. The next simplest implementation is where the users send their input and output information to some meeting point server, and the server creates the transaction and asks people to sign it. The server learns the mapping, but no one else does, and the server still can't steal the coins.

More complicated implementations are possible where even the server doesn't learn the mapping.

E.g. Using chaum blind signatures: The users connect and provide inputs (and change addresses) and a cryptographically-blinded version of the address they want their private coins to go to; the server signs the tokens and returns them. The users anonymously reconnect, unblind their output addresses, and return them to the server. The server can see that all the outputs were signed by it and so all the outputs had to come from valid participants. Later people reconnect and sign.

Similar things can be accomplished with various zero-knowledge proof systems.

Does the totally private version need to have a server at all? What if it gets shut down?

No. The same privacy can be achieved in a decentralized manner where all users act as blind-signing servers. This ends up needing n^2 signatures, and distributed systems are generally a lot harder to create. I don't know if there is, or ever would be, a reason to bother with a fully distributed version with full privacy, but it's certainly possible.

What about DOS attacks? Can't someone refuse to sign even if the transaction is valid?

Yes, this can be DOS attacked in two different ways: someone can refuse to sign a valid joint transaction, or someone can spend their input out from under the joint transaction before it completes.

However, if all the signatures don't come in within some time limit, or a conflicting transaction is created, you can simply leave the bad parties and try again. With an automated process any retries would be invisible to the user. So the only real risk is a persistent DOS attacker.

In the non-decentralized (or decentralized but non-private to participants) case, gaining some immunity to DOS attackers is easy: if someone fails to sign for an input, you blacklist that input from further rounds. They are then naturally rate-limited by their ability to create more confirmed Bitcoin transactions.

Gaining DOS immunity in a decentralized system is considerably harder, because it's hard to tell which user actually broke the rules. One solution is to have users perform their activity under a zero-knowledge proof system, so you could be confident which user is the cheater and then agree to ignore them.

In all cases you could supplement anti-DOS mechanisms with proof of work, a fidelity bond, or other scarce resource usage. But I suspect that it's better to adapt to actual attacks as they arise, as we don't have to commit to a single security mechanism in advance and for all users. I also believe that bad input exclusion provides enough protection to get started.

Isn't the anonymity set size limited by how many parties you can get in a single transaction?

Not quite. The anonymity set size of a single transaction is limited by the number of parties in it, obviously. And transaction size limits as well as failure (retry) risk mean that really huge joint transactions would not be wise. But because these transactions are cheap, there is no limit to the number of transactions you can cascade.

In particular, if you have can build transactions with m participants per transaction you can create a sequence of m*3 transactions which form a three-stage switching network that permits any of m^2 final outputs to have come from any of m^2 original inputs (e.g. using three stages of 32 transactions with 32 inputs each 1024 users can be joined with a total of 96 transactions). This allows the anonymity set to be any size, limited only by participation.

In practice I expect most users only want to prevent nosy friends (and thieves) from prying into their financial lives, and to recover some of the privacy they lost due to bad practices like address reuse. These users will likely be happy with only a single pass; other people will just operate opportunistically, while others may work to achieve many passes and big anonymity sets. All can coexist.

How does this compare to zerocoin?

As a crypto and computer science geek I'm super excited by Zerocoin: the technology behind it is fascinating and important. But as a Bitcoin user and developer the promotion of it as *the* solution to improved privacy disappoints me.

Zerocoin has a number of serious limitations:

- It uses cutting-edge cryptography which may turn out to be insecure, and which is understood by relatively few people (compared to ECDSA, for example).
- It produces large (20kbyte) signatures that would bloat the blockchain (or create risk if stuffed in external storage).
- It requires a trusted party to initiate its accumulator. If that party cheats, they can steal coin. (Perhaps fixable with more cutting-edge crypto.)
- Validation is very slow (can process about 2tx per second on a fast CPU), which is a major barrier to deployment in Bitcoin as each full node must validate every transaction.

- The large transactions and slow validation also means costly transactions, which will reduce the anonymity set size and potentially make ZC usage unavailable to random members of the public who are merely casually concerned about their privacy.
- Uses an accumulator which grows forever and has no pruning. In practice this means we'd need to switch accumulators periodically to reduce the working set size, reducing the anonymity set size. And potentially creating big UTXO bloat problems if the horizon on an accumulator isn't set in advance.

Some of these things may improve significantly with better math and software engineering over time.

But above all: **Zerocoin requires a soft-forking change to the Bitcoin protocol**, which all full nodes must adopt, which would commit Bitcoin to a particular version of the Zerocoin protocol. This cannot happen fast—probably not within years, especially considering that there is so much potential for further refinement to the algorithm to lower costs. It would be politically contentious, as some developers and Bitcoin businesses are very concerned about being overly associated with "anonymity". Network-wide rule changes are something of a suicide pact: we shouldn't, and don't, take them lightly.

CoinJoin transactions work today, and they've worked since the first day of Bitcoin. They are indistinguishable from normal transactions and thus cannot be blocked or inhibited except to the extent that any other Bitcoin transaction could be blocked.

(As an aside: ZC could potentially be used externally to Bitcoin in a decentralized CoinJoin as a method of mutually blinding the users in a DOS attack resistant way. This would allow ZC to mature under live fire without taking its costs or committing to a specific protocol network-wide.)

The primary argument I can make for ZC over CoinJoin, beyond it stoking my crypto-geek desires, is that it may potentially offer a larger anonymity set. But with the performance and scaling limits of ZC, and the possibility to construct sorting network transactions with CJ, or just the ability to use hundreds of CJ transactions with the storage and processing required for one ZC transactions, I don't know which would actually produce bigger anonymity sets in practice. E.g. To join 1024 users, just the ZC redemptions would involve 20k * 1024 bytes of data compared to less than 3% of that for a complete three-stage cascade of 32 32-way joint transactions. Though the ZC anonymity set could more easily cross larger spans of time.

The anonymity sets of CoinJoin transactions could easily be big enough for common users to regain some of their casual privacy and that's what I think is most interesting.

How does this compare to CoinWitness?

CoinWitness is even rocket-sciency than Zerocoin, it also shares many of the weaknesses as a privacy-improver: Novel crypto, computational cost, and the huge point of requiring a soft fork and not being available today. It may have some scaling advantages if it is used as more than just a privacy tool. But it really is overkill for this problem, and won't be available anytime real soon.

Sounds great! Where is it?

Theres the rub: There exist no ready made, easy-to-use software for doing this. You can make the transactions by hand using bitcoin-qt and the raw transactions API, as we did in that "taint rich" thread, but to make this into a practical reality we need easy-to-use automated tools.

Luke has written up some sketches a protocol which would enable establishing joint transactions over the regular Bitcoin network.

The Bitcoin-qt RPC system provides everything someone needs to write a side-car applet (including the ability to lock txouts to prevent them from being spent out from from under it) that participants in such a system. But the fact that so many users use centralized webwallets today which can spy on them will ultimately limit the userbase for these tools.

Personally, most of my coding brain capacity is spent on other things which are even more important to me. And what I could spare on Bitcoin is spent on more core and security things— if I work on anything wallet related anytime soon it will likely be improving the privacy behavior of coin selection... But moreover:

Anyone who builds this is going to be accused of enabling criminal activity, it doesn't matter if any actual criminals use this or not: Criminal activity sells headlines. Being a Bitcoin core developer already fills my quota for accusations of this kind, especially my quota for risk that I'm not even paid for. 🙂

In reality, real criminals don't need CoinJoin if they have even the slightest clue: They can afford to *buy* privacy in a way that regular users cannot, it's just a cost of their (often lucrative) business.

Joe-criminal can go out and buy 120% PPS mining to get brand new coins, or run his money through a series of semi-sham high cashflow gambling businesses for a 50% cut, they can afford the cost of seeking out and interfacing with these seedy services... Joe and Jane doe? Their names are up in neon on blockchain.info. It might not seem great to them, but if there a high cost of fixing it they simply won't, because the cost of fixing it is very concrete and the cost or privacy loss is speculative and distant. They might just need to give up bitcoin and switch to something almost totally private: cash... Regular users need efficient and inexpensive privacy if it is to help them at all.

I know that making such a tool doesn't fit into the get-rich-quick mold of many Bitcoin businesses, but the importance is self-apparent and the simplest versions of this don't require very deep technical wizardry. I think the "political" risk of improving people's privacy is a real one that you should carefully consider, but around these parts I see people sticking their names on some rather outrageously risky stuff. I'd hoped the "taint rich" thread would be enough to inspire some community action, but perhaps this will be.

So, instead, I ask *you*: Where is it?

gmaxwell
Moderator
Legendary

Activity: 3024
Merit: 3468

Re: CoinJoin: Bitcoin privacy for the real world (someday!)
August 22, 2013, 02:35:24 AM

#2

In order to further incentivize work in this space there is now a multisignature escrow bounty fund:

3M8XGFBKwkf7miBzpkU3x2DoWwAVrD1mhk
(yes, Bitcoin addresses can also start with a 3)

This is a two-of-three multisignature escrow with myself, Theymos, and Pieter Wuille as signers. To release any coin sent to this address at least two of these people must sign the transaction.

The bounty fund will pay out as funds are available according to the signers best judgment for completed work proposed in this thread that furthers the goal of making improved transaction privacy a practical reality for Bitcoin users.

Please feel free to contribute to the above address to support work on this infrastructure.

Multisig address construction details:

Code:

```
Key from Theymos:
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA256

Here is a public key of mine, usable for the CoinJoin bounty fund:
02d5f2b9c68b22006161dfe58a78b37dc2b577e8bb4e4522940830264eb3b3a38b
-----BEGIN PGP SIGNATURE-----

iF4EAREIAAYFAlISs5MACgkQxlVWk9q1kednkgD/WvE3F1hSoKHIr+y7q3O6xbGp
FM+P/lVbi/nZugrlNKABALMhYih2Ov80OS1PLMX9UpONn2eE2Xu+ZkxZ2SkQFfCU
=lFI0
-----END PGP SIGNATURE-----

Key from Gmaxwell:
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Here is a public key of mine, usable for the CoinJoin bounty fund:
027b48575a15712867a8a1e6a9f52f510946130bbdf3b1e2feb344b8b68232ffb1
```

TheButterZon
Legendary

Activity: 2646
Merit: 1011

RIP Mommy

Re: CoinJoin: Bitcoin privacy for the real world (someday!)

August 22, 2013, 02:51:02 AM #3

Yum

Quote from: SaltySpitoon on February 21, 2018, 12:07:45 AM

> Saying that you don't trust someone because of their behavior is completely valid.

Johnathan
Newbie

Activity: 39
Merit: 0

Re: CoinJoin: Bitcoin privacy for the real world (someday!)

August 22, 2013, 02:54:22 AM #4

Quote from: gmaxwell on August 22, 2013, 02:32:31 AM

> (As an aside: ZC could potentially be used externally to Bitcoin in a decentralized CoinJoin as a method of mutually blinding the users in a DOS attack resistant way. This would allow ZC to mature under live fire without taking its costs or committing to a specific protocol network-wide.)

This is an extremely interesting idea. Could you elaborate on how the Zerocoin transaction stages map to the stages of CoinJoin transaction creation?

gmaxwell
Moderator
Legendary

Activity: 3024
Merit: 3468

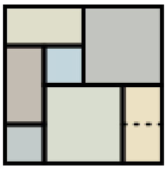
Re: CoinJoin: Bitcoin privacy for the real world (someday!)

August 22, 2013, 03:14:49 AM #5

Quote from: Johnathan on August 22, 2013, 02:54:22 AM

> This is an extremely interesting idea. Could you elaborate on how the Zerocoin transaction stages map to the stages of CoinJoin transaction creation?

For non-decenteralized coincoin, you simply pass around a transaction and sign it. It's a single sequence and an atomic transaction, you'd make two loops through the users, one to discover the inputs and outputs, and another to sign them. There really aren't stages to it.



Making a decenteralized CoinJoin secure, private, and resistant to DOS attack (people refusing to sign in order to make it fail) is trickier... for the privacy and dos attack resistance you can use ZC:

Presume the participants for a transaction are sharing some multicast medium and can all communicate. They need to accomplish the task of offering up inputs (txid:vout) for inclusion in the transaction and then, in an unlinkable way, providing outputs to receive their coins.

Each participant connects and names bitcoin input(s), an address for change (if needed), and the result of performing a ZC mint transaction to add to the ZC accumulator. They sign all this with the keys for the corresponding inputs proving its theirs to spend.

Then all the parties connect again anonymously and provide ZC redeem transactions which specify where the resulting bitcoins should go.

On their original connections all parties can now see the redeem transactions and are convinced that they were provided by the correct parties, and that they themselves will be paid. So they all sign the transaction.

If a party fails to sign, everyone else is convinced that its because they are jamming the process (intentionally or maliciously) and then can all ban (ignore in the future) whatever costly identity they used to enter the mix, or — if there is no other mechanism— that particular txin which they used.

This isn't the only way to do this in a decentralized manner, the way to do it with blind signatures is fairly similar:

Each participant connects, names Bitcoin input(s), an address for change (if needed), a key for blind signing, and a blinded hash of the address they want paid. They sign all this with the keys for the corresponding inputs proving its theirs to spend.

Each participant then blind signs the blinded hashes of all participants (including themselves).

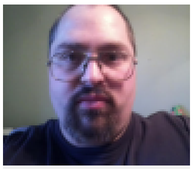
Each participant then reconnects anonymously and discloses their unblinded values and all the signatures. Because all the participants can see all the signatures, they know all are authentic. They sign, and if they refuse to sign everyone is convinced that the refusing signer is attempting to jam and bans them.

The most obvious difference between the two techniques is that the blind signing requires N^2 signatures, and potentially three communication phases (submit, blindsign, redeem) instead of two (mint, redeem) if you wanted this process to span more than a single event.

As I said above, I generally think the non-decenteralized versions of these transactions will be implemented and commonly used first, simply because they're so much less work to do.

jnagyjr
Member

Activity: 96
Merit: 10



Re: CoinJoin: Bitcoin privacy for the real world (someday!)
August 22, 2013, 05:25:34 AM

#6

Quote from: gmaxwell on August 22, 2013, 02:35:24 AM

> In order to further incentivize work in this space I'd like to create a multisignature escrow bounty fund.
>
> Theymos has agreed to be a cosigner with me and I'm currently looking for a third party. (Unfortunately, many of the people I'd normally ask would like to be potential bounty recipients, and I'd rather reduce the possible conflict of interest)
>
> The bounty fund would pay out of this fund as funds are available according to the signers

Psalm 15

> best judgment for completed work proposed in this thread that furthers the goal of making improved transaction privacy a practical reality for Bitcoin users.
>
> I'll update this post when I have it ready.

I'd like to help out with this if possible. I have little BTC to put into such a bounty fund, nor am I a programmer. Let me know what I can do to help.

Proverbs 12:1

domob
Legendary

Activity: 1111
Merit: 1113

Re: CoinJoin: Bitcoin privacy for the real world (someday!)
August 22, 2013, 07:47:43 AM

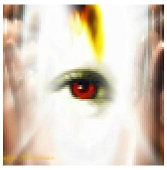
#7

Take a look also here: https://bitcointalk.org/index.php?topic=200952.0 I think this is a similar concept, which is currently being worked on already (and

Use your Namecoin identity as OpenID: https://nameid.org/
Donations: 1**domob**KsPZ5cWk2kXssD8p8ES1qffGUCm | NMC: NC**domob**cmcmVdxC5yxMitojQ4tvAtv99pY

ShadowOfHar
Legendary

Activity: 1470
Merit: 1003



Bringing Legendary Har® to you since 1952

Re: CoinJoin: Bitcoin privacy for the real world (someday!)
August 22, 2013, 07:52:01 AM

#8

@gmaxwell

I'm saluting to the unquestionalble genius of you and (almost ?) everybody else in this community.

Do you have any plans to start working on this feature soon ?
EDIT:
OK i read your post completely and appears that you are not.

{{ Bitcoin.pl Elite Member, Bitcointalk ★Legendary★ [1344] }}
1NLWBAD7ZD82fJDDawKfp5RAKSR8YWWYd3 | GPG/PGP Keys [3072D/F92EDBA4]
Hate forced fees ? Use "No Forced TX Fee" Bitcoin Client fork. **Now updated to v0.10.1 !** (use

Mike Hearn
Legendary

Activity: 1526
Merit: 1008

Re: CoinJoin: Bitcoin privacy for the real world (someday!)
August 22, 2013, 11:42:30 AM

#9

This is a really nice writeup, thanks Gregory. Such ideas have been kicked around informally for a while:

https://bitcointalk.org/index.php?topic=175156.msg1829259#msg1829259

see the part about p2p mixing protocols ... but it's good to have a name and a formal writeup.

The examples of how ordinary, everyday privacy leaks can cause people problems are great. I think I've named the "people learning each others salaries" one before, but birth control is an interesting one.

I think adding a rendezvous mechanism to the P2P network makes sense. It's already a broadcast network after all. So perhaps the right design is not to try and do absolutely everything over the existing P2P network but rather allow people to announce rendezvous points (Tor hidden services?) over the broadcast channel and then allow nodes to set announcement filters like they set Bloom filters today. If you are an SPV/leaf node on the network you wouldn't hear announcements until you request them. Other nodes would relay them all.

The difficult part is that you need a lot of traffic to make this work. Current tx volumes don't even reach one per second. So to accumulate enough users for a mix, you'd need to wait a while. It's fine for some kinds of payments that aren't time sensitive, but it's not going to work today for restaurant bills.

If I were doing it, I'd want to do the bulk of the implementation in bitcoinj of course, just because that's what most users are going to end up using (given current trajectories). It also has the advantage that using a managed language like Java eliminates entire classes of security holes, always a concern when writing financial software.

The advanced crypto part isn't necessarily that advanced and doesn't require ZK proof systems. Such protocols were already designed:

http://blog.ezyang.com/2012/07/secure-multiparty-bitcoin-anonymization/

It just requires secure multi-party sorts, which is a more well studied subset of general MPC.

bg002h
Donator
Legendary

Activity: 1438
Merit: 1005

Bitcoin
Foundati
Lifetime Memb

I outlived my lifetime membership:)

Re: CoinJoin: Bitcoin privacy for the real world (someday!)
August 22, 2013, 01:17:54 PM
#10

I'm too dumb to contribute to the code...but not so dumb as to not contribute to the bounty. Thank you for the detailed explanation of this important issue.

Hardforks aren't that hard. It’s getting others to use them that's hard.
1GCDzqmX2Cf513E8NeThNHxiYEivU1Chhe

blueadept
Full Member

Activity: 225
Merit: 100

Re: CoinJoin: Bitcoin privacy for the real world (someday!)
August 22, 2013, 01:28:53 PM
#11

Meni Rosenfeld has also proposed using commutative encryption to mask participants from each other in such a scheme:

Like my posts? Connect with me on LinkedIn and endorse my "Bitcoin" skill.
Decentralized, instant off-chain payments.

Peter Todd
Legendary

Activity: 1106
Merit: 1027

Re: CoinJoin: Bitcoin privacy for the real world (someday!)
August 22, 2013, 03:05:08 PM
#12

FWIW I'll try implementing a "coin dust" collector this weekend that just does makes SIGHASH_NONE|ANYONECANPAY signatures for the dust in your wallet and sends them to a central collection point.

Regarding mixing in general if you can split up the mix protocol into separate stages of "I want these txouts", "I'll sign for those txouts", make broadcasting those requests use up a limited resource, and finally have a P2P flood fill network where finding the originator of a message is hard you can easily make a system that is both anonymous and DoS resistant. If we add a messaging layer to Bitcoin where messages are paid for somehow, perhaps by fees paid, this can be easily developed into a automatic 'coinjoin' system for every transaction made. In addition if you can add expiry times to 'txout requests' when peers connect you can give them the outstanding requests, which means allows nodes that just connected to their peers to quickly make a transaction even if they weren't online while the request was made.

What's really nice about this implementation is that while on the one hand it's a mixer, on the other hand it's also just a DoS-resistant way of making transactions smaller by getting multiple parties together to make them. It doesn't require any dedicated "rendezvous" points that may find their actions legally frowned upon, nor does it require a separate system (like Tor) be installed to talk to those points. The general purpose messaging layer could be useful for other things too - it'd be easy to use it for alerts for instance.

Rough technical sketch:

1) Create the messaging layer

1.1) Define NODE_MSG to signal that a node will relay messages. Define "message" inventory type and add to ppszTypeName. Optional: define "realms" of messages split up by a UUID or something.

1.2) Write code to maintain an inventory of such messages. Basically this will look kinda like the mempool, and there will be some time period for which old messages are expired, as well as a limit on total messages stored.

1.3) Relay those messages to peers/respond to getdata requests. If "realms" are supported, a bloom filter to select which realms a peer is interested in is useful.

2) Make DoS-attacks expensive

2.1) For every mempool tx, record the scriptPubKeys spent, and take the fees of the tx and proportion them to those scriptPubKeys.

2.2) Add a way to for messages to be signed by scriptPubKeys, and reduce the "balance" of fees recorded for each message. (amount sacrificed per msg should be configurable) Note that it may be necessary to only allow for fees to be used from transactions that have been actually mined - not sure how much that opens up attacks.

2.3) Other methods, PoW, fidelity bonds etc. possible too.

3) Add mix protocol

2.1) Define "I want these txouts" and "I'll sign for these txouts" messages.

2.2) Add logic for the announce/sign sequence, along with options for the user to decide how fast they want the tx to be generated. Typically after a few seconds I'd expect the program to give up and just sign the txouts it already has, thus creating a non-mixed tx. Note how if you need more than one txout in a given transaction this can still be a privacy improvement, because you can arrange so that the txin/txout set is still indistinguishable from a two-party tx.

2.3) Add better logic to make sure the fee-paying tx's used don't break anonymity - you wouldn't want to use the same scriptPubKey for txout announce and signature announce. Also clever crypto can help here too - but that can be added later. Remember that timing is a potential anonymity breaker too, so randomize it, and also randomly sometimes wait for the other party to broadcast signatures, or sometimes broadcast signatures yourself.

2.4) Long-term: add more SIGHASH options to make it easier to specify what txin's and txouts a signature applies too on a set basis, rather than the current inflexible options available.

4) Add prioritization so that blocks always get priority over message data and are transferred around the network fastest. Note how having non-blockchain data actually helps the overall resistance to traffic analysis for the whole system by providing a constant stream of data for which latency is less important to hide the data for which latency is more important.

Another fun one: with realms you can port completely different applications, like IRC chat, to run over this basis concept. Only nodes that are interested in a particular application, or are willing to lend bandwidth to allow users of that application more anonymity, need to relay messages for a given realm, which keeps the whole system scalable. Needs some work to actually find peers for a given realm, but that's just a matter of extending the address gossip functionality.

BTC: 1FCYd7j4CThTMzts78rh6iQJLBRGPW9fWv PGP: 7FAB114267E4FA04

TierNolan
Legendary

Activity: 1232
Merit: 1005

Re: CoinJoin: Bitcoin privacy for the real world (someday!)

August 22, 2013, 03:44:37 PM #13

Quote from: retep on August 22, 2013, 03:05:08 PM

> FWIW I'll try implementing a "coin dust" collector this weekend that just does makes SIGHASH_NONE|ANYONECANPAY signatures for the dust in your wallet and sends them to a central collection point.

That basically sends the dust to anyone who wants it? How does that help?

Quote

> 3) Add mix protocol
>
> 2.1) Define "I want these txouts" and "I'll sign for these txouts" messages.

Since everyone has to pay for the service, if it fails, everyone loses. However, the DOS attacker loses faster?

The "currency" for paying for message relay is effectively coin-age?

The transaction would be flooded in stages, everyone gets version 1 then version 2 then version 3 and so on.

Eventually someone signs and that ends the mixing.

Signing the transaction early also acts as a DOS attack.

There would also be a requirement that "cleaned" coins have standard sizes. You pay for something and you get 2 change payments, a cleaned standard value coin and the remainder.

This creates an incentive for merchants to set prices as one of the standard coin sizes. One coin moving through the system doesn't link multiple transactions together at least.

1LxbG5cKXzTwZg9mjL3gaRE835uNQEteWF

Peter Todd
Legendary

Activity: 1106
Merit: 1027

Re: CoinJoin: Bitcoin privacy for the real world (someday!)

August 22, 2013, 05:59:15 PM #14

Quote from: TierNolan on August 22, 2013, 03:44:37 PM

> **Quote from: retep on August 22, 2013, 03:05:08 PM**
>
> > FWIW I'll try implementing a "coin dust" collector this weekend that just does makes SIGHASH_NONE|ANYONECANPAY signatures for the dust in your wallet and sends them to a central collection point.
>
> That basically sends the dust to anyone who wants it? How does that help?

People appear to have been sending very large numbers of addresses dust as a way to break anonymity. Granted, they also may have been doing it as a way to get signatures from scriptPubKeys due to the 'R' re-use issue, but the script would use bitcoind to spend the dust which is known to not be vulnerable.

Also there's lots of pretty much unspendable dust out there from Satoshidice and others, and again such a script can help.

Quote from: TierNolan on August 22, 2013, 03:44:37 PM

> **Quote**
>
> > 3) Add mix protocol
> >
> > 2.1) Define "I want these txouts" and "I'll sign for these txouts" messages.
>
> Since everyone has to pay for the service, if it fails, everyone loses. However, the DOS attacker loses faster?
>
> The "currency" for paying for message relay is effectively coin-age?

No, it's fees paid by previous transactions.

Now that does raise the question of where do those fees come from if you haven't made a transaction in awhile? One decent option is in fact to spend coin-age by signing to a txout that you don't actually intend to spend, however that's open to DoS attack by entities that simply have a lot of Bitcoins. Fidelity bonds are another option.

As an aside, the actual process of spending credit should include a field for the current balance of credit prior to that spend, and the hash of the previous time this credit was used to make double-spending credit not possible. This is particularly important with the coin-age or fidelity bond version, because there it's quite possible for nodes to securely give their peers the current status of the credit balance based on smallest balance left.

Quote from: TierNolan on August 22, 2013, 03:44:37 PM

> The transaction would be flooded in stages, everyone gets version 1 then version 2 then version 3 and so on.
>
> Eventually someone signs and that ends the mixing.
>
> Signing the transaction early also acts as a DOS attack.

I'm actually thinking that mixes should have a small number of participants, usually two or three. Remember that we want the process of creating a tx to be as fast as possible, and multiple rounds of mixes wind up with just as much anonymity as fewer rounds with more participants in each round.

The economics of the DoS attack are such that the attacker wastes only a small integer multiple less fees than the target(s), and the targets are already spending the fees anyway. IE the defenders already have the resource, fee paying transactions, that the attacker has to buy specifically to launch the attack.

Quote from: TierNolan on August 22, 2013, 03:44:37 PM

> There would also be a requirement that "cleaned" coins have standard sizes. You pay for something and you get 2 change payments, a cleaned standard value coin and the remainder.

Standard sizes do not have to be a requirement actually. Suppose Alice has 2BTC and wants to send 1.5BTC to Bob: she can announce that she wants a 1.5BTC txout with Bob's scriptPubKey, and a 0.5BTC txout to a change address and broadcast that. Now suppose Charlie has 0.75BTC and simply wants to mix some of it, but doesn't really care how much. He can note that Alice has a 0.5BTC txout, and broadcast a request to make a transaction with her txouts, as well as a 0.5BTC txout to one of his addresses, and a 0.25BTC txout to a change address of his.

When the final transaction gets mined there are two 0.5BTC txouts - but who's are they?

Even in the general case where you have an Alice and a Bob who both want to send money it's often hard to figure out whose inputs and outputs are whose. Do the txouts belong to the same person, and they were just paying multiple people? Which of the multiple txins was actually owned by who?

It's easy to make the task of following the transaction graph very difficult without trying provided a lot of people are combining their transactions.

BTC: 1FCYd7j4CThTMzts78rh6iQJLBRGPW9fWv PGP: 7FAB114267E4FA04

TierNolan
Legendary

Activity: 1232
Merit: 1005

Re: CoinJoin: Bitcoin privacy for the real world (someday!) #15
August 22, 2013, 06:56:45 PM

Quote from: retep on August 22, 2013, 05:59:15 PM

> Also there's lots of pretty much unspendable dust out there from Satoshidice and others, and again such a script can help.

True. People might be willing to throw it away.

Quote

> I'm actually thinking that mixes should have a small number of participants, usually two or three. Remember that we want the process of creating a tx to be as fast as possible, and multiple rounds of mixes wind up with just as much anonymity as fewer rounds with more participants in each round.

True, and as you say, it isn't about perfection, it just increases the effort.

1LxbG5cKXzTwZg9mjL3gaRE835uNQEteWF

phelix
Legendary

Activity: 1708
Merit: 1005

nmc:id/phelix

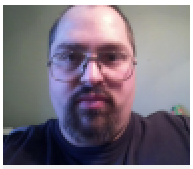
Re: CoinJoin: Bitcoin privacy for the real world (someday!) #16
August 22, 2013, 08:40:52 PM

Finally... atomic coin laundry. Nice. 😁

blockchained.com ▪ bitcointalk top posts

jnagyjr
Member

Activity: 96
Merit: 10



Psalm 15

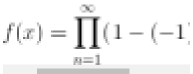
Re: CoinJoin: Bitcoin privacy for the real world (someday!) #17
August 22, 2013, 08:57:11 PM

If the goal is a simple way to regain anonymity, why are there counter-proposals to make regaining it more complex? Just curious.

Proverbs 12:1

marcus_of_au
Legendary

Activity: 3150
Merit: 1382



Re: CoinJoin: Bitcoin privacy for the real world (someday!)
August 23, 2013, 02:49:13 AM

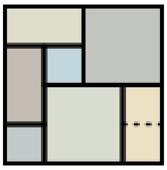
#18

I can probably contribute commits to this, debugging, testing, cross-platform building, etc.

Monetary Freedom - an inalienable right
Per aspera ad astra

gmaxwell
Moderator
Legendary

Activity: 3024
Merit: 3468



Re: CoinJoin: Bitcoin privacy for the real world (someday!)
August 23, 2013, 03:19:15 AM

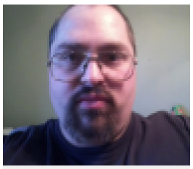
#19

Quote from: jnagyjr on August 22, 2013, 08:57:11 PM
> If the goal is a simple way to regain anonymity, why are there counter-proposals to make regaining it more complex? Just curious.

I'm not sure what you're referring to, but in general I think a lot of people have not thought the fungibility implications through, and are also confusing privacy and anonymity because they are intimately related, especially where pseudoynmity is used to achieve privacy. Anonymity is fundamentally hard, and consider anonymity improvements a side effect of good privacy. If you want to go around telling someone which transactions are yours, you still can. And, an interesting point of that is that it's not at all incompatible with what is proposed here. Even if you're happy telling a particular set of people all about your transactions, that doesn't imply you also want the whole world to know.

jnagyjr
Member

Activity: 96
Merit: 10



Psalm 15

Re: CoinJoin: Bitcoin privacy for the real world (someday!)
August 23, 2013, 03:45:54 AM

#20

Quote from: gmaxwell on August 23, 2013, 03:19:15 AM
> **Quote from: jnagyjr on August 22, 2013, 08:57:11 PM**
> > If the goal is a simple way to regain anonymity, why are there counter-proposals to make regaining it more complex? Just curious.
>
> I'm not sure what you're referring to, but in general I think a lot of people have not thought the fungibility implications through, and are also confusing privacy and anonymity because they are intimately related, especially where pseudoynmity is used to achieve privacy. Anonymity is fundamentally hard, and consider anonymity improvements a side effect of good privacy. If you want to go around telling someone which transactions are yours, you still can. And, an interesting point of that is that it's not at all incompatible with what is proposed here. Even if you're happy telling a particular set of people all about your transactions, that doesn't imply you also want the whole world to know.

I'll accept that, but the counter-proposals/additions to your initial proposal do nothing to keep the complexity for the user down. As an end user who is looking for ease of use and decent privacy, some of the other schemes here, unless done transparently, are just too much to keep up with.

Proverbs 12:1

Pages: [**1**] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 »

print

« previous topic
next topic »

Bitcoin Forum > Bitcoin > Development & Technical Discussion > **CoinJoin: Bitcoin privacy for the real world**

Jump to: => Development & Technical Discussion ▼ go