BitMingle: Distributed Bitcoin Mixing with Interest

Reid Bixler and Carter Hall
University of Virginia
{rmb3yz, carter}@virginia.edu

Abstract

For those who desire truly anonymous online transactions, Bitcoin itself is not sufficient; analysis of the public ledger can often deanonymize users. Therefore, many privacy-seeking Bitcoin users either pay mixing services like BitMixer[0] to launder their coins for them, or use a decentralized protocol such as CoinJoin[1] or CoinShuffle[2]. Here we present BitMingle, a distributed peer-to-peer Bitcoin mixing protocol with small fees paid out to those who add their coins to the mixing pool. This fee incentivizes normal Bitcoin users to contribute to the anonymity of the system, while making passive income.

Introduction

Bitcoin is a popular decentralized digital cryptocurrency introduced by Satoshi Nakamoto in 2009. It has gained notoriety for its use in various criminal enterprises, notably the Silk Road online drug marketplace. Bitcoin is often an attractive payment solution for criminals and others who desire privacy due to its disconnect from any governments or existing financial institutions.

However, contrary to what many believe, Bitcoin is not an anonymous currency; the full transaction history stored in the blockchain inhibits true anonymity. However, it is often referred to as “pseudonymous”; while a single identity can often be traced within the Bitcoin network, it is supposed to be difficult to link Bitcoin addresses to real identities. In reality, with modern data analysis tools, it has been shown that deanonymizing Bitcoin users can be very feasible.

Bitcoin mixing can add a layer of obscurity to Bitcoin addresses. The idea behind mixing is to pass coins through a network of addresses before returning to the original owner on a different address, making it much less obvious that the original owner is still in control of the coins.

A few distributed Bitcoin mixing protocols exist; see CoinShuffle[2] and CoinJoin[1]. There are several reasons why a decentralized mixing protocol is more desirable than a for-profit mixing service such as BitMixer[0]. The primary reason is that a centralized service has to control all of the mixed coins at the same time, which means that the mixers can know whose coins end up where. This opens the door for government subpoenas and other violations of privacy.

BitMingle adds an incentive for non-launderers to participate in the mixing transactions, in the form of a small mixing fee that is distributed from launderers to “lenders.” This is intended to add a large volume of clean addresses to the mixing pool, which benefits the launderers, as well as providing an opportunity for holders of Bitcoin to make passive income.

Terminology

We will use the following definitions throughout the paper:

Launderer: Someone who wishes to push their coins through a mixing service

Lender: Someone who does not require that their coins are mixed, but is willing to add their coins to a mixing pool

Mingle: Transaction using the BitMingle protocol

Background

Bitcoin

Bitcoin is a decentralized digital currency that solves the double-spending problem with a public ledger that can only be forged if an attacker controls over half of the computing power of the entire network. One potential downside of this public ledger is that transactions are not truly anonymous; funds can be traced through the blockchain, and the more transactions there are to trace, the more likely it is that a vulnerability will allow an attacker to deanonymize an individual.

CoinShuffle

CoinShuffle[2] is an existing distributed Bitcoin mixing protocol designed to protect anonymity while remaining flexible. The basic idea behind CoinShuffle is that several intended launderers gather into one transaction where all inputs and outputs are equal; for example, each participant may put in 10 BTC and get out 10 BTC. The idea is that since everyone launders the same amount of money at a time, it is impossible to link the inputs and outputs. CoinShuffle uses cryptographic concepts to guarantee that each participant will get the correct amount of coins at the end of the transaction. As an additional advantage, there are no mixing fees to be paid to some service. BitMingle is built on top of a modified version of the CoinShuffle protocol.

Desired Properties

BitMingle shares most of CoinShuffle's desired properties; namely, unlinkability, verifiability, robustness, compatibility, efficiency, and a small impact on Bitcoin. These are fully explained in Section 3.1 of the CoinShuffle paper [2]. The additional desired property is the fee, which is collected from launderers and distributed to lenders.

The Protocol

The Network

In order to make this protocol the best it can be, the introduction of a network available to all Bitcoin users who wish to participate is necessary. The intent of the network is to allow any user to broadcast to all users or read the broadcasts of any other user. There is no requirement to be a part of the network, and much in the same as CoinShuffle, the network would be some form of "a public bulletin board acting as a facilitator or through a peer-to-peer protocol specifically crafted for this purpose". Also in the same vein as CoinShuffle, "the participants must agree on a channel for further communication" once they end up finding each other in the network.

The Minglers

The users of the network, which we have decided to call *minglers* (M), would be broken up into 2 unique categories: *Launderers* (M_A) and *Lenders* (M_E). Any mingler can broadcast their intent as a Launderer or a Lender at any time, but each has separate data that they are going to be broadcasting on the network.

What Is A Mingle?

A *Mingle* will be the requested mixing transaction that is created by a Launderer (M_A) that has criteria set by said Launderer. The criteria are as follows:

Mingle Size (**S**) - The required number of participants for the mingle transaction (includes M_A)

Expiration (**E**) - The amount of time the Launderer is willing to wait for **S** participants, if the Launderer does not finish the Mingle by **E**, either by not getting enough participants or by the Launderer not wishing to proceed, then the broadcast will completely cancel and all data related will be deleted

Required Input (**R**) - The specific amount of Bitcoin that the Launderer wants to launder, which is required by all participants to have

Fee (**F**) - The percentage of **R** that the Launderer is willing to pay to create the Mingle, the minimum of which will be described later in more detail

Output Addresses ($\mathbf{O_A}$) - The number of output addresses required by each participant to be a part of the Mingle transaction

With the criteria above, the Launderer will broadcast the Mingle request across the network looking for Lenders wanting to join in.

In order for a Lender to join a broadcasted Mingle they must meet the criteria and broadcast a *Join Request* (J) to the Launderer which contains:

> Required Input (**R**) - The amount of Bitcoin that the Lender must have in a single Bitcoin address to join the Mingle, the address of which (Input Address) will be sent in the Join Request
>
> Output Addresses ($\mathbf{O_A}$) - The number of output addresses the Lender must have available, the addresses of which (Output Addresses) they will send in the Join Request. For maintaining anonymity, the output addresses must not be the same as the input addresses.

As Lenders send their Join Requests to the Launderer, they will be added to the Request List (RL) which will be visible to the Launderer until the Expiration of the Mingle. As soon as the Request List contains enough participants (including the Launderer) to match the set Mingle Size, the Launderer can finalize the Mingle.

For the safety and anonymity of the Launderer as well as any honest Lenders, the Launderer will perform a couple of simple checks on the Lenders; first, the Launderer will check to make sure that no Lenders share an IP address, and second, will do a breadth-first search on all of the Lenders' Input Addresses up to whatever the Launderer considers reasonable (about 3-4 deep is good in the average case). The intention of this search is to find if any of the Input Addresses are linkable to one another, because there is a risk that a malicious user could attempt to undermine the unlinkability of the Mingle by being the only entity besides the Launderer in a Mingle. If the malicious party can significantly decrease the number of other participants besides the Launderer in the Mingle, then there is a high risk to the Launderer that their mixed output will be linkable by the malicious party. By performing this breadth-first search and checking the IP addresses, the Launderer will be maximizing their own anonymity in the transaction by minimizing the likelihood that many if not all the Lenders are the same entity.

While the Launderer can finish the Mingle at any time as long as the Mingle Size is reached, there are incentives to wait for the Request List to grow. The intention of waiting for the Request List to grow is the same as the search on Input Addresses, in that there is a risk that many or all of the lenders in the Request List could be the same malicious entity, with Input Addresses that are not linkable to one another. While it would be difficult to do this as a malicious entity, it is still a possibility, so in order to counteract this the Launderer will be able to randomly select Lenders from a large Request List to meet the required Mingle Size. By doing this, the Launderer significantly decreases the likelihood that many or all of the Lenders in the Mingle transaction will be the same malicious entity.

After a Launderer picks enough Lenders from the Request List to meet the Mingle Size, the Launderer creates a transaction with his/her own Input Address with the Required Input and all of the Lenders' Input Addresses as the inputs, such that all the inputs for the transaction will be the same exact amount. The outputs of the transaction will be broken down into 2 different types: Fee Outputs (**F**) and Launder Outputs (**L**). The outputs will be set up following the layout as follows:

Fee Outputs ($\mathbf{O_F}$) = Mingle Size (**S**) - 1

The number of fee outputs will equal to the number of Lenders in the transaction, as the Launderer will not be getting any fees.

Launder Outputs ($\mathbf{O_L}$) = Mingle Size (**S**) * Output Addresses ($\mathbf{O_A}$)

The number of launder outputs will equal to the number of participants in the transaction multiplied by the required number of output addresses per participant.

Total Outputs (**T**) = # Fee Outputs (**F**) + # Launder Outputs (**L**)

The number of total outputs will be equal to the number of fee outputs and launder outputs combined.

Val(Launder Outputs) = Required Input (**R**) - (Required Input (**R**) * Fee (**F**))

The value of the launder outputs per participant will be equal to the required input minus the fee that the Launderer is paying, such that all the launder outputs equal the same value as the Launderer is getting in the end. If there are multiple output addresses, the Launderer will decide the amounts that go into each, but those output addresses that get the launder outputs must add up to be same same for all participants.

Val(Fee Outputs) = Fee (**F**) / # Fee Outputs ($\mathbf{O_F}$) + Fee (**F**)

The value of the fee outputs per participant will be equal to the fee that the Launderer is paying plus the fee divided by the number of fee outputs. Put simply, the Lenders will get back the Bitcoin they put in plus the fee they get for lending their money in the mixing transaction. The launder outputs are what all parties get back, including the Launderer. The fee output is the rest of what the lender put in plus the fee that they get for participating.

Lender Gain (**G**) = Fee (**F**) / # Fee Output ($\mathbf{O_F}$)

This is the actual amount of bitcoin that the Lender will be getting out of the transaction. The gain will be equal to the fee that the Launderer is paying divided by the lenders in the transaction. The biggest thing to consider with this value is that if it is too low then there is not likely to be many lenders willing to be a part of the protocol, and if it is too high then there is not likely to be many launderers. We have considered both sides of the balance, and come up with the thought that a minimum lender gain of **0.1%** is a hard requirement in this protocol. Without this as a requirement, there is the risk that the Launderer requests hundreds of Lenders and pays a very small Fee, therefore decreasing the incentive of being a Lender.

Another thing to be considered is that the Mingle transaction will almost definitely require a transaction fee to be included on the blockchain. In order to make this simple, we have decided that the transaction fee will be taken equally from all participants. Essentially, the Fee must meet the criteria such that the cost of the transaction fee for the lenders is not making the lender gain less than the minimum lender gain to all lenders, otherwise there is risk that the transaction won't be included on the blockchain or there is not enough incentive to the Lender.

By following the format as set above, the entire transaction will be a completed Mingle and will be broadcast in the same format as a CoinShuffle transaction. This means that the Launderer will sign off on his/her inputs and outputs, then pass it on to the next participant who signs off on his/her inputs and outputs, and so on and so forth until all participants have verified and signed for the transaction to be valid. With the transaction fully signed for, the Mingle transaction can be broadcast to the Bitcoin network and thus the transaction will be complete.

Visualization

For a clearer representation, we will go about describing a hypothetical situation of a mingle transaction. Say there exists a Launderer (A) that wishes to launder 10 BTC and he decides to use the BitMingle protocol to do so. He will join in the network of BitMingle and broadcast the creation of a Mingle with himself as the Launderer with the following requirements:

Required Input (**R**) - 10 BTC
Fee (**F**) - 10% or 1 BTC
Mingle Size (**S**) - 5
Output Addresses ($\mathbf{O_A}$) - 1
Expiration (**E**) - 72 hours

First we must check if the Fee is high enough. Assuming that the transaction fee for being added to the blockchain is ignored (in reality it would be split between the participants and the lender gain must still be at minimum or greater when considering the cost of transaction fee), we then check the lender gain to be 0.25 BTC because there is a 1 BTC fee for a Mingle Size of 5,

or 4 Lenders. A Fee of 10% leads to a lender gain of 2.5% which is more than enough to satisfy the 0.1% minimum. Obviously in a real life case, the Launderer will more likely than not set it at or near the minimum, but this is much higher for the ease of explanation.

With the broadcast of A's Mingle Request, any and all users of the BitMingle network would be able to see his request. For the sake of the example, say 10 Lender's send a Join Request to A within 50 hours of the Mingle Request. These requests will include the following:

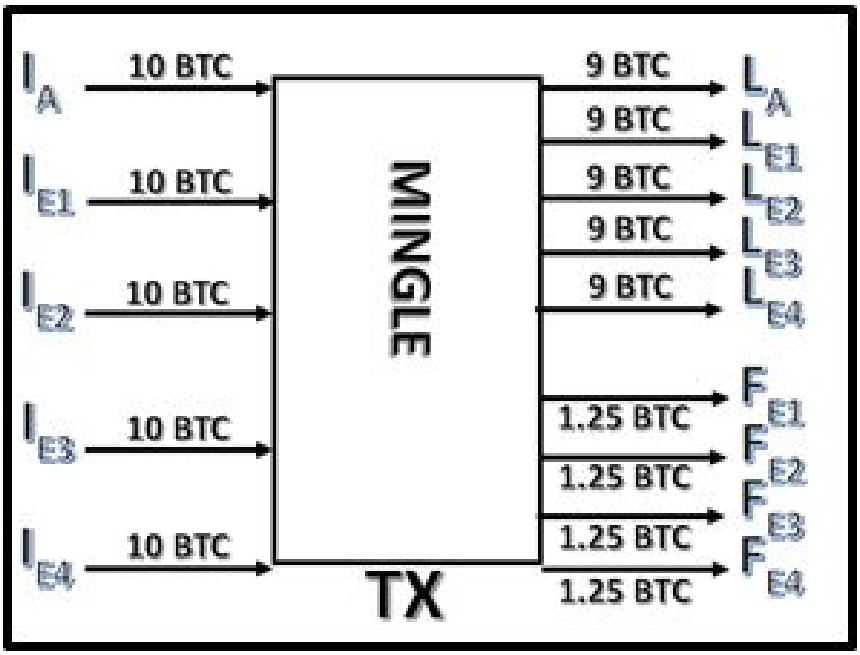
> Required Input (**R**) - Input addresses ($\mathbf{I_X}$) containing exactly 10 BTC from Lender X
>
> Output Addresses ($\mathbf{O_A}$) - 1 Output address ($\mathbf{L_X}$) for the launder output and 1 Output address ($\mathbf{F_X}$) for the fee output both owned by Lender X

Launderer A will look through the Request List and check the following criteria:

1. For all I_X, I_Y there doesn't exist an X, Y such that there is no relation within 3 transactions (the value 3 is chosen by Launderer A). If any X, Y is found, remove the pair from the Request List.
2. Similarly, for all I_X, I_Y, there does not exist a pair X, Y that broadcast from the same IP address. If any X, Y is found, remove the pair from the Request List.
3. For all I_X, L_X, and F_X, for Lender X, there does not exist an X such that I_X, L_X, or F_X are the same. If any X is found, remove from the Request List.
4. For any I_X, there doesn't exist an X such that I_X does not only contain the specified R (10 BTC in this instance). If any X is found, remove from the Request List.

If the Request List is still equal to the Mingle Size - 1 or greater, the Launderer will randomly select the Lender's from the Request List to satisfy the Mingle Size. The randomly selected Lenders will be E_1, E_2, E_3, and E_4 in this instance.

The Launderer now creates the Mingle Transaction with 5 input addresses (I_A, I_{E1}, I_{E2}, I_{E3}, and I_{E4}), each containing exactly 10 BTC. The output addresses will be split into launder outputs (L_A, L_{E1}, L_{E2}, L_{E3}, and L_{E4}) and fee outputs (F_{E1}, F_{E2}, F_{E3}, and F_{E4}). The launder outputs will get the initial Required Input of 10 BTC minus the fee of 10% or 1 BTC, therefore making all launder outputs exactly 9 BTC. The fee outputs will get the leftover amount divided by the number of lenders, as in the lenders will get back 1.25 BTC, with 1 BTC being from their initial 10 BTC and 0.25 BTC being their Lender Gain from the Fee. This entire transaction can be seen in the diagram below:



With all of the inputs and outputs set (ignoring transaction fee for ease of example), the Launderer follows the same protocol as CoinShuffle by signing off on his own inputs and outputs, and then passing it on to the next participant. If at any point any participant does not wish to continue with the transaction, they simply just do not have to sign off on the transaction, thus forwarding to the blame phase like in the CoinShuffle protocol. Assuming that all participants are satisfied with the given transaction, the fully signed and verified transaction will be broadcast to the Bitcoin network to be added to the blockchain.

Once added to the blockchain, as long as the Launderers and Lenders try to maintain the anonymity and unlinkability of their mixed coins after the transaction by ensuring their “network metadata, e.g., their IP address, does not reveal their identities or make the spending transaction linkable to a run of the mixing protocol”, then the Launderer will have successfully mixed his 10 BTC and can guarantee his own anonymity, at a slight cost of 1 BTC.

Analysis

Benefits

With the introduction of this protocol, we are providing a much more likely to be used protocol that still maintains are large amount of anonymity depending on how much the launderer wants. However, what are the incentives to be a launderer? Why can't I as a potential launderer just wait around looking at the network for a transaction that I can join in so that I can both launder my coin and make money from it at the same time? Our solution to that is the fact that the launderer is in charge of most everything. It is the launderer who sets the size, fee, expiration, number of output addresses, and gets to choose the participants from the passing Request List. It would make sense as a user who wishes to mix coins, but also has both time and doesn't care as much about the anonymity or being in charge of the conditions to become only lenders.

However, there are many instances where, especially with a minimum lending fee being not too high, many users would be happy to take the launderer option. Being a launderer means that you can focus on maximizing anonymity, speed of transaction, or a mix of the two. In order to maximize anonymity, the launderer can: require many more participants (therefore increasing the fee they pay), vary the fee they pay above the minimum, require more output addresses to make it more difficult to track, and they can choose who is being added in the transaction. In order to maximize speed of transaction, the launderer can: ask for a very minimal required input to get many lenders to choose from, ask for a small size to minimize wait time, and can pay a higher fee in order to get more lenders quicker.

At the same time, if you have Bitcoin just sitting in your wallet, why not become a lender? There are a few incentives of being a lender, but the largest of which is obviously the fact that you can make money from just having Bitcoin. Simply through the act of offering to be a part of a transaction, even if the fee is the minimum, you still have an opportunity to make Bitcoin at a minimum of 0.1% of what you put in. At the same time, you also have the opportunity to mix your own Bitcoin at the same time as making it, which means that you are still also laundering your money but are actually being paid for it. You also have an incentive as a lender to try to remain honest because you will more often than not get removed from the Request List if you are showing signs of malicious intent because of the protocol checks. The quicker the transactions that happen, the more Mingles that you as a lender can join in, the more money that you can get from just having Bitcoin.

Restrictions and Requirements

All inputs must be the same with the idea that it will be more difficult to track the transaction from its inputs if all of them are the same. Also without the inputs being the same, it would fundamentally break the protocol because then the outputs could not be equal.

All related outputs must be the same, including if there are multiple outputs, because that is the core functionality of the protocol. If the outputs are not the same, then it is easy to trace what output goes to what input and it is therefore no longer anonymous and unlinkable. In consideration of multiple outputs, say the Required Input is 10 BTC, the Launderer wants 3 Output Addresses, and he's willing to pay a fee of 1 BTC. The Launderer decides that the 3 Output Addresses will get 2, 3, and 4 BTC respectively across all participants' launder addresses. If the launderer does not enforce this, then he is risking the ability for an outside party to link output addresses together based off of their values.

There must be a minimum Lender Gain, and therefore a minimum Fee dependent upon Mingle Size. We have set this at 0.1% at the moment, but this could change in order to maximize the usage of BitMingle for both Launderers and Lenders. The transaction fee will be included across all participants, as explained previously, and the Lender Gain must still be equal to or greater than 0.1% when including the cost of transaction fee. The introduction of this prevents attacks from malicious parties wishing to create too many transactions by giving too small of fees or by trying to make the lenders not want to participate because of minute gain.

There must also be a minimum Required Input in order to prevent attacks perpetrated by a malicious Launderer wishing to create too many easy transactions that either the BitMingle or Bitcoin network could not handle. If the Required Input is too small then either it won't cover the transaction fee, or there will be too many Lender's wanting to join in on the transaction which could potentially upset the BitMingle network with too many Join Requests.

Risks

First and foremost, it goes without saying that Launderers and Lenders should use their mixed outputs logically and not try to link it back to their old inputs. As mentioned before they must ensure their metadata doesn't reveal who they are when using the outputs or that they can't be traced back to the Mingle transaction. However, this is still a risk taken on by the Launderer as he is the one who cares the most about mixing his coins.

As talked about in the explanation of the protocol, there is always going to exist some malicious party with the intent to ruin the system. There are a few cases that the malicious party may try to ruin this protocol, but we have introduced safeguards to prevent most of these from happening. By searching on the inputs to see if they are not from the same party, checking that the input and output addresses are not the same, checking the sources of the lenders, and randomly picking from the Request List will prevent users wishing to put up the facade that they are multiple lenders.

Even with these safeguards though, there are still risks, if very minimal, that these do not catch a malicious party. If a malicious party happened to have a multitude of addresses, not linked to one another, all containing the Required Input, and enough addresses not linked to one another

for fee and launder outputs and was able to send all the requests from different IPs, then there is a chance that the Launderer ends up randomly picking multiple of the malicious user's lender requests. In the worst case, the Launderer creates a transaction with only himself and the malicious party. This means that the malicious party knows exactly which output address the Launderer is, and therefore is able to link the input addresses to the output addresses, breaking the anonymity of the protocol. However, having considered this, as long as the launderer selects a logical Mingle Size and follows all the safeguards for checking for malicious users, it is unlikely that somebody would end up being able to do this, but the Launderer must always keep this in mind when using the protocol.

Another risk in this protocol is in the introduction of the output addresses. If the output addresses of the Lender's are already 'known', as in a malicious party knows or is able to find out which output addresses already belong to specific Minglers in the transaction, then there is a possibility that the Launderer's anonymity is at risk. Our solution to this is not only the safeguards suggested earlier to maximize the randomness of those chosen in a transaction, but also if a Launderer wishes for maximum anonymity they can request that the output addresses of the Lenders to be totally new addresses. Without the output addresses being new addresses, there is also a high risk that of the launder outputs, the Launderer's output address is the only one that is totally new, because he is the one who cares about his anonymity. This means it would be extremely easy to see which of the output addresses was newly created and then assume with fairly high certainty that that address is the Launderer's. However, with the requirement that all output addresses being new will surely guarantee the Launderer's anonymity.

Finally, and most importantly, the biggest risk that the Launderer is taking with this protocol is the possibility of Lender's putting their 2+ outputs together, therefore revealing who the Launderer was. If all of the Lenders in a Mingle transaction decide to combine their launder and fee outputs, it would be obvious that the one leftover launder output was the launderer. We have thought through many possible solutions to this, but many of them require things that no longer makes this a decentralized mixing protocol. Our best solution is first and foremost, require that the lender must act in good faith and not combine their outputs. This is of course a very big risk for the Launderer to be taking on, but then we propose that a Launderer can technically check the 'history' of a Lender by checking all the addresses used by a single IP when sending a Join Request. This obviously could end up requiring a lot of work to do, but it is still an option for a paranoid Launderer wanting to maximize their own anonymity. As long as most of the Lenders do not combine their outputs, then the Launderer has nothing to fear, but this is by far the biggest risk taken with this protocol.

Conclusion

The goal of BitMingle is to add lender incentives to the existing CoinShuffle protocol without significantly reducing the anonymity provided by CoinShuffle. While adding lenders who do not care about privacy as much as the launderers do has inherent risk, we leave the launderer in control of his/her Mingle, and provide tools to help ensure that privacy is not being violated. As long as the launderer attempts to maximize his/her anonymity by following the protocol and safeguards correctly, then there is a small but nonzero chance that their transaction can be de-anonymized. The rest of the anonymity of the transaction requires that the lenders act in good faith and remain honest with the idea that if they try to break the protocol that they will likely never be included in Mingle transactions again. We believe this protocol has lots of opportunities and benefits over similar mixing protocols with the introduction of the incentivized fee for lenders. This incentive along with the launderer's control over the transaction introduces a good balance between the two different options in the BitMingle protocol.

References

[0] BitMixer https://bitmixer.io/

[1] CoinJoin https://bitcointalk.org/index.php?topic=279249

[2] CoinShuffle http://crypsys.mmci.uni-saarland.de/projects/CoinShuffle/coinshuffle.pdf