An Analysis of Darkcoin's Blockchain Privacy via Darksend+

Authored By: Kristov Atlas
Document version: 2, published 2014/09/19

To Hal, who reminds us that time is scarce and privacy sacred.

Table of Contents

Introduction

A complete copy of this document can be downloaded from the following URL:
http://cdn.anonymousbitcoinbook.com/darkcoin/darksend-paper/

This document analyzes the blockchain privacy afforded by Darkcoin's Darksend+ technology. It is not intended to be an exhaustive review of the programmed behavior of Darksend+. It also does not consider other channels for leaking privacy-impacting information, such as the transaction broadcasting system found in all Bitcoin-like crypto-currencies [Biryukov] [Koshy].

This document is based on the analysis of typical Darkcoin transactions on the currency's testnet, using versions 0.10.12.32 and 0.10.13.1 of the Linux client.

Financial Disclosure

My efforts to analyze the security and privacy of Darkcoin's Darksend+ technology have been crowd-funded by the Darkcoin community in the form of Darkcoin tip payments [crowdfund]. In addition, I have at various times owned darkcoins, and may own them in the future.

Background

Privacy issues are both inherent in the design of Bitcoin, and inflicted by users and service providers upon themselves through ignoring best practices. Early in the Bitcoin ecosystem, entrepreneurs launched trust-requiring, third-party mixing services to help Bitcoin users improve their privacy. These services required users to send their bitcoins to an intermediary, and trust that the intermediary would send an honest and successfully mixed amount of coins after a delay. In late 2013, various decentralized mixing protocols were proposed, including CoinJoin [CoinJoin]. These decentralized protocols make the mixing process "trust-less" in that participants cannot steal each other's funds, unlike a trust-requiring mixing service. If properly designed, these protocols can also be extended to become "blind" [Matonis]. A small number of Bitcoin clients have implemented variations of CoinJoin to date, some with fatal privacy flaws [CoinJoin Sudoku] that illustrate the significant number of privacy-impacting design decisions involved in implementing CoinJoin.

In the years since Bitcoin was invented, a number of alternative crypto-currencies have been developed with a focus on improving on Bitcoin's privacy weaknesses. Approaches include:

- Trust-requiring mixing nodes (e.g. FedoraCoin)
- Ring signatures (e.g. CryptoNote)
- Zero-knowledge proofs, zkSNARKs, etc. (e.g. Zerocoin/Zerocash)
- Off-chain transactions (e.g. BitcoinDark)
- Secure multi-path routing (e.g. XCurrency)
- Anonymizing stake mining (e.g. CloakCoin)

In 2014, Evan Duffield launched XCoin using the X11 proof-of-work algorithm and with plans to improve on Bitcoin's privacy issues by baking superior privacy directly into the currency [birth]. Later, XCoin was renamed to Darkcoin. Evan's privacy technology was branded Darksend, and remains closed source to date, with promises to open source Darksend later in 2014.

Darksend is based on the CoinJoin protocol. The original version of Darksend greatly resembled the initial CoinJoin proposal [Duffield], requiring users' clients to coordinate at the same place and time on the Internet to mix amounts of exactly 10 darkcoins (DRK) together. This coordination, characteristic of the CoinJoin protocol, was performed by a special kind of node on the Darkcoin network called a "Masternode"; eventually, these Masternodes were incentivized to facilitate Darksend transactions by receiving dividends in the form of transaction fees, and artificial scarcity of Masternodes was created through a 1000 DRK reserve requirement.

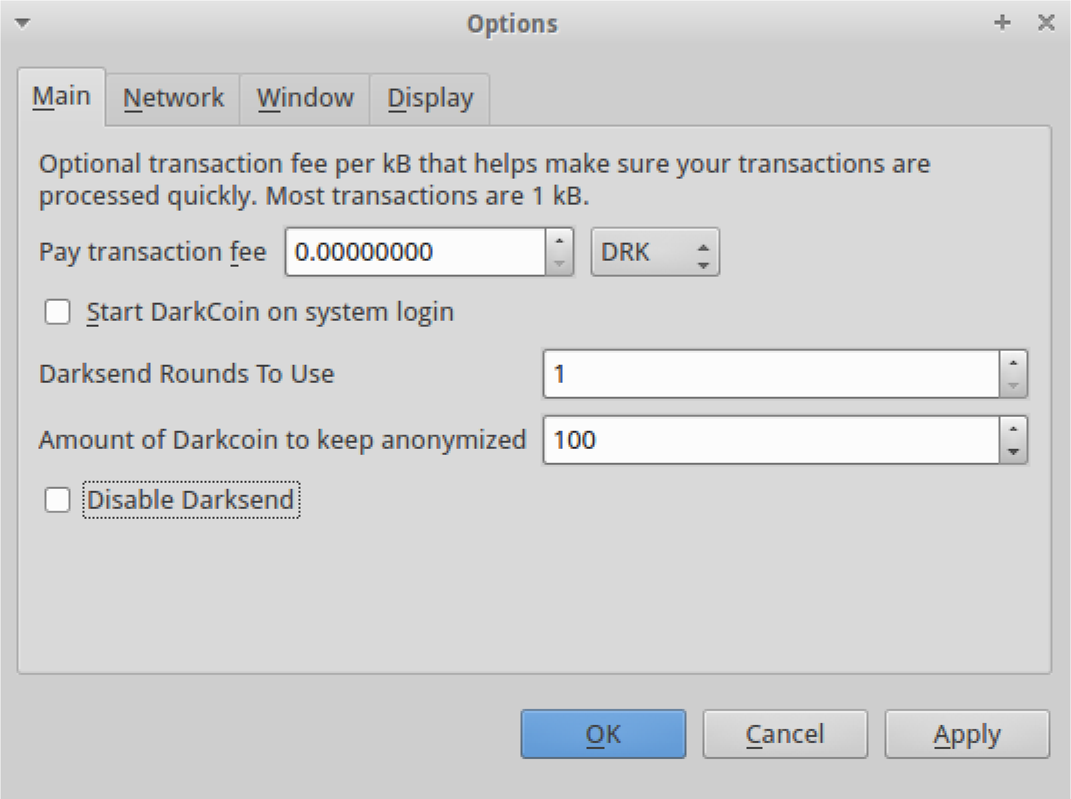
The most recent version of Darksend, Release Candidate 4 ("RC4"), introduces improvements to the original Darksend design, and has been rebranded as "Darksend+." Darksend+ is still based on the CoinJoin protocol, but improves user convenience by turning the mixing process into a background, passive (but still online) process. This approach can be described as ahead-of-time (AOT) mixing, as opposed to the customary just-in-time (JIT) mixing typical of most CoinJoin implementations today. When sending funds, users of the reference Darkcoin client can specify whether the funds should have first gone through the Darksend+ process or not. Users can also specify the number of rounds of Darksend+ that their funds must go through in order to be considered "anonymized." This change to Darksend modifies the language that we use to describe the process of improving privacy; whereas CoinJoin has in the past been an operation that takes place at the last moment when funds need to be transmitted between two users, funds sent via Darksend+ have already gone through an anonymization process and only a standard, single-payer transaction is required to transmit funds on to the next user. If CoinJoin was a car wash that you traveled to and drove through in order to clean your car, Darksend+ would be a service that washes it in your driveway every night so that you always find the car clean when you're ready to drive.

Darksend+ improves user privacy by breaking darkcoins up into smaller pools of funds — called "denominations" — and by making the ownership of those denominations ambiguous. Two or more Darksend+ peers will create a common transaction, each feeding their input funds into the transaction and receiving output denominations out of it that cannot be clearly attributed to either user. Funds are broken down into common, denominated amounts so that the greatest number of users can mix funds with each other in this fashion. The sameness of the denominations is what provides the ambiguity of ownership. You can think about it this way: Imagine you're flying in a helicopter trying to track a red car on the highway, and it passes under a bridge. If two red cars emerge on the other side of the bridge, it's ambiguous which one you want to follow. If a blue car and a red car emerge, then it's not ambiguous at all, and the chase ensues.

A Typical Round of Darksend+

Client Configuration

The client can be configured to determine how much darkcoin will be put through the Darksend+ process, and how many rounds of Darksend+ are required by the user in order to be considered “anonymized.” The current default of the client is to “anonymize” 100 DRK with 1 round of Darksend+ (See Figure 1). The amount of darkcoin to be “anonymized” may be rounded up to a slightly higher number, but will never fall below the specified amount so long as the wallet possesses sufficient funds.



Optional transaction fee per kB that helps make sure your transactions are processed quickly. Most transactions are 1 kB.

Pay transaction fee 0.00000000 DRK

□ Start DarkCoin on system login

Darksend Rounds To Use 1

Amount of Darkcoin to keep anonymized 100

□ Disable Darksend

Figure 1: A screenshot of the Darkcoin client, applying 1 round of Darksend to 100 DRK.

The Splitting Phase

Before any given round of Darksend+, the client may create one or more intermediary transactions to split funds into more manageable amounts; this is referring to the "splitting phase" in the Darksend+ design, and is applied whenever a transaction input is larger than a pre-determined constant (currently 1000 DRK). In the Darksend source code, this algorithm is governed by the `SplitUpMoney()` and `DoAutomaticDenomination()` functions.

The Anonymizing Phase

The client will then initiate one or more multi-party transactions. These transactions will include the funds of two or more Darksend client peers. A randomly-selected Masternode orchestrates the signing of the transaction between the peers. On the blockchain, this type of transaction resembles several individual Bitcoin transactions "pasted" together into a single Darkcoin transaction. The outputs are either broken into into a constant Darksend+ denomination (currently 500.00000001, 100.00000001 DRK, 10.00000001 DRK, or 1.00000001), an amount suitable for paying transaction fees (currently 0.0125 DRK), or leftover change that does not fit into either category. The 0.00000001 DRK (colloquially known as a "duff" after the Darkcoin inventor) included in denominations aids the software in distinguishing Darksend+ denominations from other funds.

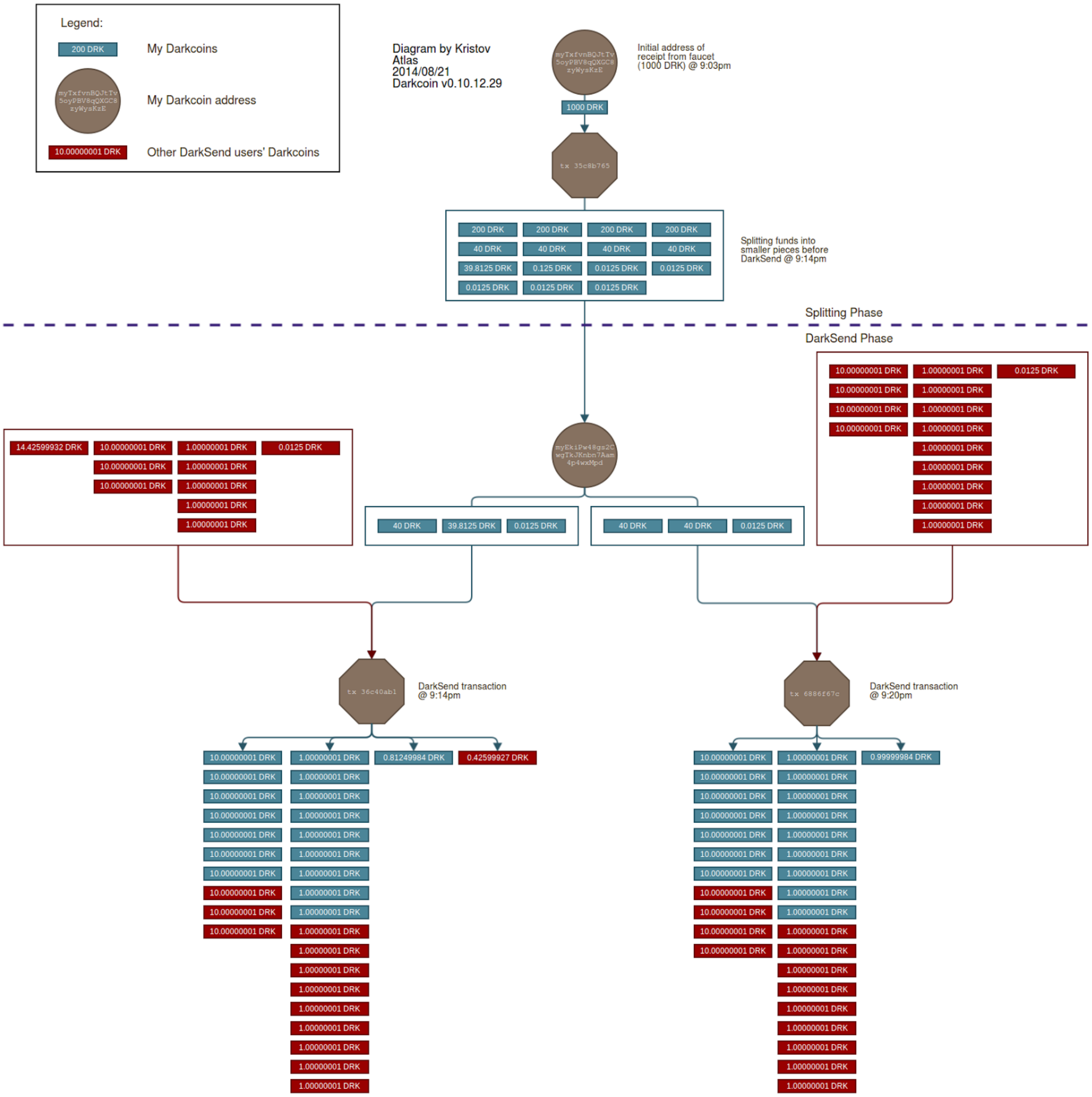


Figure 2: A typical series of Darksend+ transactions for 158.00000032 DRK and 1 round.

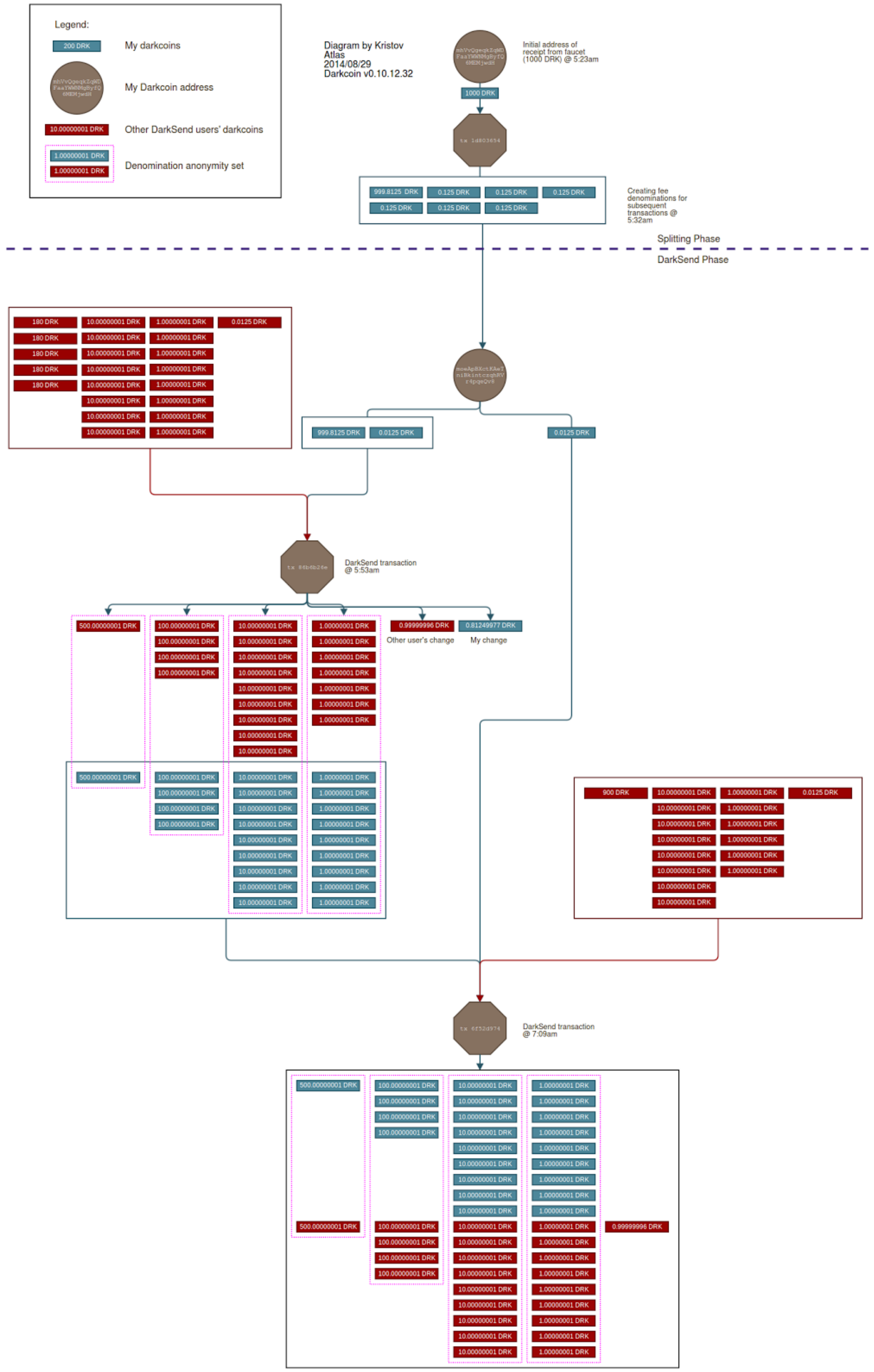


Figure 3: A typical series of Darksend+ transactions for 999.00000023 DRK and 2 rounds.

Types of Darksend Observers

In order to understand the level that Darksend+ provides to users, we must first consider which types of transactions observers there are. Each type will have different privacy requirements and attack capabilities based on what information they have access to. These types can be broken down into the following:

- **Darkcoin Senders and Receivers**: Darkcoin users who send and receive darkcoins to and from one another. These are differentiated from Darksend peers, who are sending to and from themselves.
- **Darksend Peers**: Other Darksend users who one connects with in order to mix with each other's funds.
- **Masternodes**: The nodes in the Darkcoin network that orchestrate CoinJoin transactions between Darksend Peers.
- **Passive Blockchain observers**: Interested parties who attempt to de-anonymize transactions without engaging in the sending or receiving of funds, nor the Darksend protocol.

Types of Attacks and Weaknesses

There are several types of attacks on privacy and privacy weaknesses that are present in Bitcoin and CoinJoin implementations. We will discuss each type of relevant attack or weakness, how Darksend+ does or does not mitigate it, and possible improvements to Darksend+ in the future to better mitigate it.

Masternode Snooping Attack

When Darksend peers are ready to mix with each other's funds, they randomly select a Masternode to orchestrate their multi-party transaction. In order to permit the Masternode to do this, they communicate both their input and output addresses to the Masternode. In other words, to the Masternode, the transaction is not an anonymous one, because the Masternode can fully trace funds from beginning to end. You can think of a Masternode as the conductor of an orchestra: even if the individual musicians are too busy playing and studying their sheet music to watch their fellow musicians, the conductor watches over all of them at all times. We'll call a Masternode that intentionally records these traces, or one that has been compromised to do so a "malicious" Masternode.

A Masternode must have 1000 DRK in reserve in order to participate. This raises the cost in order to operate an intentionally malicious Masternode. This also provides higher incentive for Masternode operators to prevent their Masternodes from being compromised, since a Masternode that is compromised for malicious purposes could also be stolen from. At the same time, this puts a hard limit on the total possible number of

Masternodes that could ever exist simultaneously (approximately 22,000), and raises the costs for altruistic users who wish to run Masternodes purely to support the privacy of Darksend. Dividends paid to Masternodes partially offset the cost to altruistic operators, but also offset the cost to malicious operators. Ultimately, whether this reserve requirement helps or harms the privacy of the Darksend ecosystem is an economic calculation that goes beyond the scope of this document.

Masternode snooping affects the different types of transaction observers as follows:

Darkcoin Senders and Receivers: Not relevant, since they aren't Masternodes.

Darksend Peers: Not relevant, since they aren't Masternodes.

Masternodes: Malicious Masternodes can record the input and output relationships for any transaction they are chosen to orchestrate. This is a significant weakness of the current Darksend+ design.

Passive Blockchain Observers: Not relevant, since they aren't Masternodes.

The minimum number of peers that use a malicious Masternode affects privacy in a few different ways. On one hand, the fewer peers that use a malicious Masternode, the fewer that are immediately affected by it. However, a malicious Masternode can potentially impact not only the transactions that it is snooping on, but also the privacy of users who mix with the impacted funds in subsequent transactions. Consider the following diagram (Figure 4):

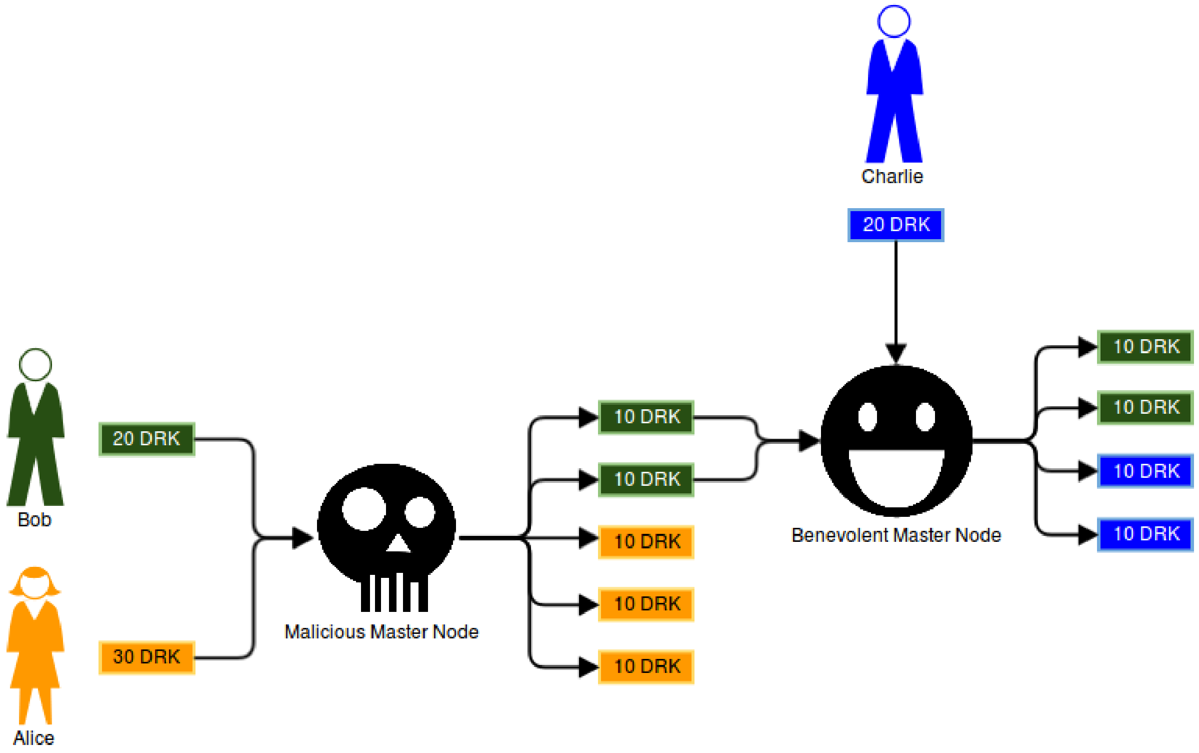


Figure 4: An example of how a malicious Masternode indirectly impacts privacy of future Darksend+ peers.

In this diagram, Alice and Bob mix their funds using a malicious Masternode that records the relationship between their inputs (20 DRK and 30 DRK) and the outputs (5 denominations of 10 DRK). Bob mixes funds with Charlie during a second round of Darksend+. To any passive observer, any of the 10 DRK denominations coming out of Bob and Charlie's round could belong to Alice, Bob, or Charlie. However, the malicious Masternode knows that the funds either belong to Bob or Charlie, reducing the privacy of Charlie's funds by 50%. Still, the malicious Masternode will not have perfect knowledge of all of the Darksend+ denominations indicated in the diagram.

Users can decrease the impact of malicious Masternodes by increasing the number of Darksend+ that they require their funds to go through in order to be "anonymized."

In the future, there are at least two ways that this could be mitigated. Chaumian blinding of inputs and outputs could "blind" the Masternode to this relationship [maaku]. The Darkcoin developers have expressed interest [questions] in implementing such a mechanism [Jeng]. Also, the CoinShuffle protocol [CoinShuffle] uses secure, multi-party sorting in a way that might be adaptable to blind the Masternode.

Sybil Attack

Whereas a Masternode snooping attack involves a malicious Masternode, a Sybil attack — also known as a pseudospoofing attack — involves malicious Darksend+ peers. If you are trying to defend your privacy from passive blockchain observers, then your Darksend+ peers aid this effort by mixing their funds with yours. But what if the people you want to defend your privacy against are, in fact, your Darksend+ peers? This is exactly how one carries out a Sybil attack on Darksend+. This is probably the most powerful vector of attack against Darksend+. It's worth noting that Sybil attacks are notoriously difficult to mitigate — after all, how do you eliminate bad actors from a system that desires to erase identity? — and that it's a class of attack that applies to many other approaches to anonymizing crypto-currency transactions, such as the ring signatures employed by CryptoNote.

Sybil attacks affect the different types of transaction observers as follows:

Darkcoin Senders and Receivers: Not relevant, since they aren't Darksend peers.

Darksend Peers: Sybil attacking peers can eliminate their own funds from a pool of mixed funds, and thus access privileged information about the source and destination of their Darksend+ peers. The future peers that are involved in a given Darksend+ transaction, the more impact a given Sybil peer has on the privacy of its partners.

Masternodes: Not relevant, since they aren't Darksend peers.

Passive Blockchain Observers: Not relevant, unless a Sybil attacker releases the details of his participation to an observer, and thus removes his own contributing funds from contention when the observer attempts to de-anonymize funds.

Users can reduce the impact of Sybil attacking peers by increasing the number of rounds of Darksend+ they require their funds to go through in order to be "anonymized." Increasing the minimum number of Darksend+ peers per mixing transaction also increases the amount of work required for would-be Sybil attackers.

There's not much research about how to combat this kind of Sybil attack. "SybilGuard" [Yu] is a technique proposed by researchers in the field of social networks to exploit trust relationships, but there are no such trust relationships in the Darksend+ network.

A zero-knowledge proof system of reputation might be one way to build trust among Darksend+ peers without sacrificing anonymity.

A simple way to reduce the effectiveness of Sybil attackers is to instruct wallets to occasionally mix with themselves during Darksend+, excluding any possible attacker from participating. However, this would need to be very carefully implemented in order to make these transactions indistinguishable from genuinely multi-party transactions.

One way to combat Sybil attacks is to raise their cost. However, costs to Sybil attackers are currently symmetric to benevolent peers in the system, because raising fees will necessarily make the system more expensive for everyone. The Darksend+ protocol could be modified so that users can set the amount of fees they require peers to pay in order to peer with them; this would allow users to determine the balance they prefer between cost and privacy. Alternatively, a proof-of-burn system might make costs for Sybil attackers asymmetric with users. In such a system, when a Sybil attacker is exposed (e.g. by presenting evidence in court, getting doxxed, etc.) their burned funds would be blacklisted.

If a typical Darksend+ transaction costs 0.0125 DRK in fees, then a Sybil attacker can perform 80,000 Sybil attacks for the approximate cost of operating a malicious Masternode (1000 DRK reserve).

Contextual Fingerprinting Attack

Example: Alice sends Bob 4 BTC on August 28th at 9am. At 9:15am, Bob tweets to Alice "received all 4 bitcoins, thanks." An attacker trying to link Alice and Bob's identities with Bitcoin addresses can then scan the blockchain for transactions of approximately 4 BTC, around the time of Bob's tweet. If the attacker finds multiple matching transactions, he can use other techniques to try to figure out which transaction was for Alice and Bob. With this information, the attacker knows Alice's sending address, Bob's receiving address, the balances of their addresses before and after the transaction, and may be able to link additional addresses to the known addresses through further blockchain analysis.

Contextual fingerprinting is an attack that takes contextual information about transactions from outside of the blockchain (such as Bob's public tweet to Alice) and applies it to the information on the blockchain to associate addresses or transactions with particular identities. An attacker may obtain this information from public sources (forum posts, social media sites, exchange order books) or from private ones (Know Your Customer data, hacked exchange databases, surveillance cameras pointed at ATMs).

In some cases, the users impacted may have little control over the attack. For example, consider a Darkcoin user who receives frequent payments throughout the day, and keeps her client open all day to put the funds through Darksend+. Someone working for the power company or ISP that services this user could observe service outages (intentionally inflicted or otherwise), and use this information to observe which addresses were idle during the outage. These forms of contextual fingerprinting attacks underline one of the weaknesses of privacy technology that requires wallets to be online. Such timing-based contextual fingerprinting attacks may be partially mitigated by starting Darksend+ rounds at random intervals; users can currently simulate this by incrementing the number of rounds of Darksend+ they require from their client by one after a random number of hours or days have passed.

An attacker using contextual fingerprinting can still examine the Darkcoin blockchain to correlate their contextual knowledge with particular transactions. The only known crypto-currencies that would combat this attack would be Zerocash-based currencies (not yet released as of writing this document) or those using off-chain transactions. Darksend+ helps to limit the effects of a successful contextual fingerprinting attack by obfuscating the flow of funds into and out of individually de-anonymized transactions. So long as senders use Darksend+ before sending, and receivers use Darksend+ after receiving, they enjoy this level of mitigation.

Contextual fingerprinting attacks affect the different types of transaction observers as follows:

Senders and Receivers: This attack is not relevant to senders or receivers, since they already know the details of their mutual transaction.

Darksend Peers: Normally, Darksend+ helps to limit the impact of a contextual fingerprinting attack by allowing users to mix funds on either side of a de-anonymized transaction. Darksend peers involved in that mixing have special insight that allows them to find out more about the de-anonymized parties, especially if there are only two peers involved in a Darksend transaction.

Masternodes: Normally, Darksend+ helps to limit the impact of a contextual fingerprinting attack by allowing users to mix funds on either side of a de-anonymized transaction. Masternodes elected to orchestrate those mixing operations have special insight that allows them to find out more about the de-anonymized parties, learning more about where the funds came from before a contextually fingerprinted transaction or where they were sent after.

Passive Blockchain Observers: The knowledge a passive observer can obtain through a contextual fingerprinting attack is limited by the fact that users can use Darksend+ to mix funds before or after the impacted transaction.

Significand Attack

Because of the high price volatility of crypto-currencies like bitcoin and darkcoin, goods, services, and crypto-currency/fiat exchanges are all frequently denominated in fiat currencies like the US dollar. When converting between a fiat currency price and a crypto-currency price that is divisible down to 8 decimal places (and in the case of bitcoin and darkcoin, can rise up to 7 digits to the left of the decimal point), this tends to create many-digit, distinctive quantities of the crypto-currency that can be recorded in a transaction on the currency's blockchain. For example, if the world knows that I sent someone exactly $5 of darkcoin at a price of 0.00607739 BTC per DRK and $484.47 USD per BTC, this will equate to sending 1.69818894 DRK — a rather distinctive amount. We can refer to the number "169818894" as the "significand" of this quantity.

When analyzing crypto-currency transactions that are divisible down to 8 decimal places, it often makes the most sense to standardize all quantities by removing the decimal place and representing them as their "standardized significands." For example, the standardized significand of 1.69818894 would be 169818894; the standardized significand of 10.5 would be 1050000000.

Even if the person paying 1.69818894 DRK were to break it up into pieces, e.g. 1 DRK, 0.09010804 DRK, and 0.60808090 DRK, statistical analysis would easily tie those pieces to the original sum. The ability for an observer of the blockchain to tie known amounts of other currencies ($5) to highly specific significant digits of a crypto-currency transaction (1.69818894 DRK) is known as the significand attack, or sometimes the "mantissa" attack [mantissa]. Since this is also a way of taking contextual information about a transaction and using it to find the transaction on the blockchain, the significand attack can be categorized as a specific type of contextual fingerprinting attack.

Just as we discussed earlier with contextual fingerprinting attacks, Darksend+ helps to mitigate the effects of a significand attack by allowing senders to obfuscate where their funds are coming from and receivers to obfuscate where their funds are going to. This limits the impact of the privacy breach to only the targeted transaction. More importantly, the design of Darksend+ prevents the significand attack from de-anonymizing the mixing transactions that take place during the Darksend+ process. All "anonymized" funds are broken into uniformed sized denominations without distinctive significant digits. Left-over funds that cannot be denominated in this way are simply returned to the user as change, and are kept separate from the "anonymized" denominations. If the user wishes, these left-over funds can be later reconstituted into large pools of funds and also put through the Darksend+ process. While even payments put through Darksend+ can be later subject to a significand attack — if I provide two *Darksent* inputs of 1.00000001 DRK and receive 0.30181108 DRK as change, it's still obvious that I've sent 1.69818894 DRK or $5 — at least the significand issue does not come up *during* Darksend+.

Significand attacks affect the different types of transaction observers as follows:

Darkcoin Senders and Receivers: Darksend+'s design prevents senders and receivers from using significands to de-anonymize each other's funds before or after their transaction.

Darksend Peers: Not relevant.

Masternodes: Not relevant.

Passive Blockchain Observers: Passive observers can still single out individual Darkcoin transactions using contextual information about the transaction, but cannot use significands to de-anonymize mixing that takes place during the Darksend+ process.

Excluded Denomination Weaknesses

The privacy that Darksend+ provides is derived from mixing funds with multiple Darksend+ peers during transactions that provide equal-sized output denominations (500, 100, 10, and 1 DRK). These equivalent denominations, belonging to different wallets, contribute to each other's "anonymity set," or the list of addresses that might be attributable to any one of several possible owners as a result of the mixing process. An Excluded Denomination Weaknesses occurs whenever transactions take place after a mixing transaction that eliminate a given denomination from another's anonymity set.

Lonely Denomination Weakness

The most obvious of Excluded Denomination weaknesses is the lonely denomination weakness. If a denomination is created that does not match any other outputs in an anonymizing transaction, then it is obvious which Darksend+ client the denomination came from.

Example: Alice and Bob are peered together for a denominated CoinJoin transaction using a crypto-currency called COIN. This crypto-currency is prone to the lonely denomination weakness. Alice mixes 111 COIN, while Bob mixes 110 COIN. Alice receives denominations of 100 COIN, 10 COIN, and 1 COIN, while Bob receives denominations of 100 COIN and 10 COIN. Alice then pays someone 11 COIN. However, in doing so, she exposes to any passive blockchain observer that she was the sender, and not Bob, since Bob's funds could not add up to 11 COIN from their denominated CoinJoin transaction.

Darksend+ ensures that lonely denominations cannot occur by checking compatibility before peering can begin, so this weakness does not apply to Darksend+.

As a side note, the fact that this compatibility check is in place may make it slightly easier for a malicious users to perform a targeted Sybil attack. If I want to disrupt the privacy of a particular Darkcoin address, and I know that I will need to be able to generate at least one 500 DRK denomination, one 10 DRK denomination, and one 1 DRK denomination, then ensuring that all of my Sybil nodes have the correct amount of funds in order to produce those denominations will increase the chances that one of them will get peered with my target.

Fat Sum Weakness

This weakness occurs whenever someone spends denominations in a single transaction that make it impossible for other Darksend+ peers to have spent. This is most easily illustrated with an example. The following diagram (Figure 5) shows Alice and Bob peering together for one round of Darksend+, followed by Alice sending Charlie some funds that have were denominated by that previous Darksend+ transaction:

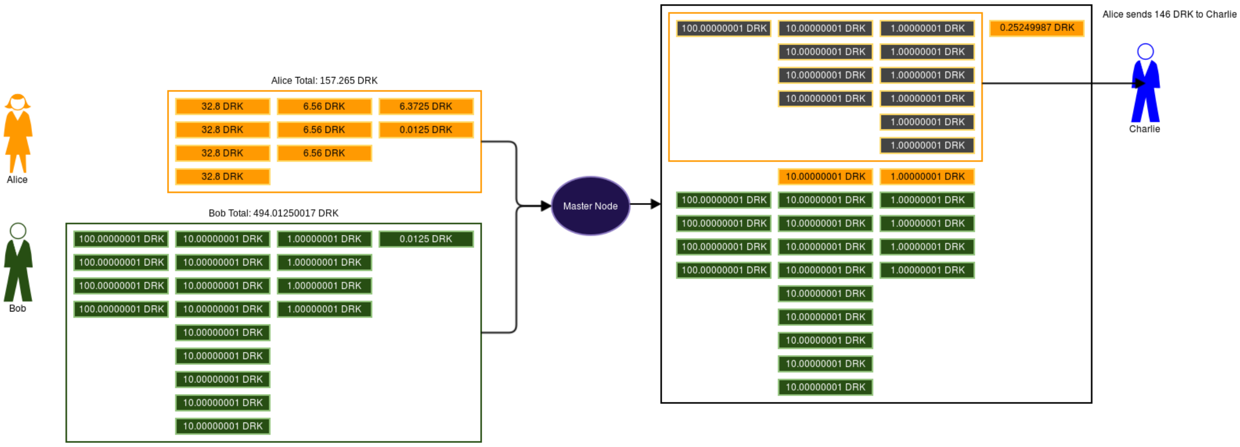


Figure 5: A series of two transactions illustrating the Fat Sum denomination weakness.

In the Darksend+ transaction, Alice inputs 157.265 DRK and Bob inputs 494.01250017 DRK. They receive roughly the same amounts denominated in the traditional Darksend+ quantities. This denomination operates deterministically; the amounts that they receive as outputs for the Darksend+ transaction are the only amounts that they could ever receive. Thus, any passive observer of the blockchain can predict which and how many denominations each user will receive based on the observed inputs.

Unfortunately, this means that when Alice sends 146 DRK to Charlie, this includes six 1.00000001 DRK denominations that could only have belonged to her (i.e. the amount she sent is too "fat" to match Bob's denominations). In doing so, she has reduced Bob's privacy. Now Bob's funds could only be mixed up with two of Alice's denominations: a single 10.00000001 DRK denomination, and a single 1.00000001 DRK denomination not sent to Charlie.

The fat sum denomination weakness affects the different types of transaction observers as follows:

Darkcoin Senders and Receivers: A fat sum condition can potentially impact a sender's privacy by making sent funds attributable only to him during a preceding round of

Darksend+. If a receiver puts his funds through Darksend+, she may also be impacted and better permit her sender to continue to track her funds.

Darksend Peers: In the case where there are only two peers in a Darksend+ transaction, this condition makes no difference, since each peer can identify the other's funds by process of elimination. If there are more than two peers, a fat sum weakness will help a malicious peer to trace other peers' funds.

Masternodes: Not applicable -- Masternode already knows everyone's input/output mappings.

Passive Blockchain Observers: Passive blockchain observers can much more easily trace funds whenever this weakness is expressed on the blockchain, especially when there are only two peers per Darksend+ transaction.

There are at least two possible ways to remedy this weakness in Darksend+:

1. Client-side restrictions that prevent users from spending denominations that are not mirrored by all of their Darksend+ peers. These restrictions would get significantly more imposing the more rounds a user has instructed their client to go through, and might be confusing for users. Trolls could still modify the client software to bypass the restriction.
2. Modify Darksend+ so that users can only create identical sets of denominations as outputs. The remaining difference between users' inputs will simply returned as change, as currently takes place for any leftover funds below 1.00000001 DRK.

Disparate Spending Weakness

Example: Right after waking up at 8AM, Nicholas configures his Darkcoin client to send his 160 DRK through Darksend+. He plans to purchase some equipment from DarkEgg later that day, and wants to do so with privacy. At the same time in Tokyo (10PM), Satoshi opens his laptop to wake it from a sleeping state. His Darkcoin client comes to life, and once it synchronizes with the network, he observers that he has received 150 DRK in tips for his outstanding whitepaper on CryptoNote. His Darkcoin client immediately searches for compatible peers, and within a couple minutes, his funds have been mixed with those of Nicholas. Satoshi goes to bed. A few hours later after returning from ninjitsu practice, Nicholas returns to his laptop to find that his funds have been mixed. He logs into DarkEgg and purchases a stick of RAM with his mixed darkcoins. After paying with darkcoins during checkout, he quickly realizes that he forgot to purchase a replacement for his cracked BlackPhone case. He logs backs into DarkEgg, and within minutes, has ordered the case, as well. It will be several more hours before Satoshi will awake to check on his funds again.

Darkcoin, like Bitcoin and almost all crypto-currencies, operates on the basis of a public ledger called the blockchain. Each transaction is not only timestamps, but indicates the specific amounts of funds that have been transacted. Since humans have schedules and habits, transactions often have patterns that hint at the identity of their owners, despite the pseudonymity afforded by addresses. An early paper on Bitcoin privacy studied, among other things, the way that users often tend to use multiple addresses in batches of time [Reid]. In the example above, Nicholas made two separate transactions within a short period of time. Satoshi, on the other hand, has a totally different schedule living across the world. An observer of the blockchain looking at the mixture of their funds from Darksend+ might not be able to tell for sure which denomination belonged to whom, but they would be able to make a strong educated guess that the two spending transactions that Nicholas initiated involved funds owned by the same person.

Based on this principle, we can deduce that Darksend+ denominations that have been mixed together are more likely to be co-owned by a single person if they are spent around the same time, and less likely to be co-owned if spent at totally different times. This can be characterized as a disparate spending weakness.

One way to reduce the privacy information leaked by the disparate spending weakness would be to try to draw spent funds from a variety of Darksend+ output groups. However, doing so would exacerbate another weakness, called the Conjoined Spend Weakness.

Conjoined Spend Weakness

Example: Script Kiddie Scott decides to "anonymize" his bitcoins by putting them through the centralized Bitcoin Fog mixing service. He sends the funds to the Fog, and provides two withdrawal addresses to receive his funds from once they have been split up and mixed via off-chain accounting. His bitcoins originate from address A, and end up mixed in addresses B and C. Both addresses B and C are listed in Scott's Bitcoin wallet. Unfortunately, when he spends his mixed funds, he forgets to use Coin Control, and he simultaneously spends funds as inputs to a transaction from addresses B, C, and D. Address D is Scott's public tipping address.

A conjoined spend weakness occurs when funds, purposely separated for privacy purposes, are conjoined as inputs to a single spending transaction, and therefore re-associated with each other on the blockchain. This can occur with Darksend+ as well, and is likely to occur, given its design.

The following diagram (Figure 6) demonstrates this weakness as it applies to Darksend+:

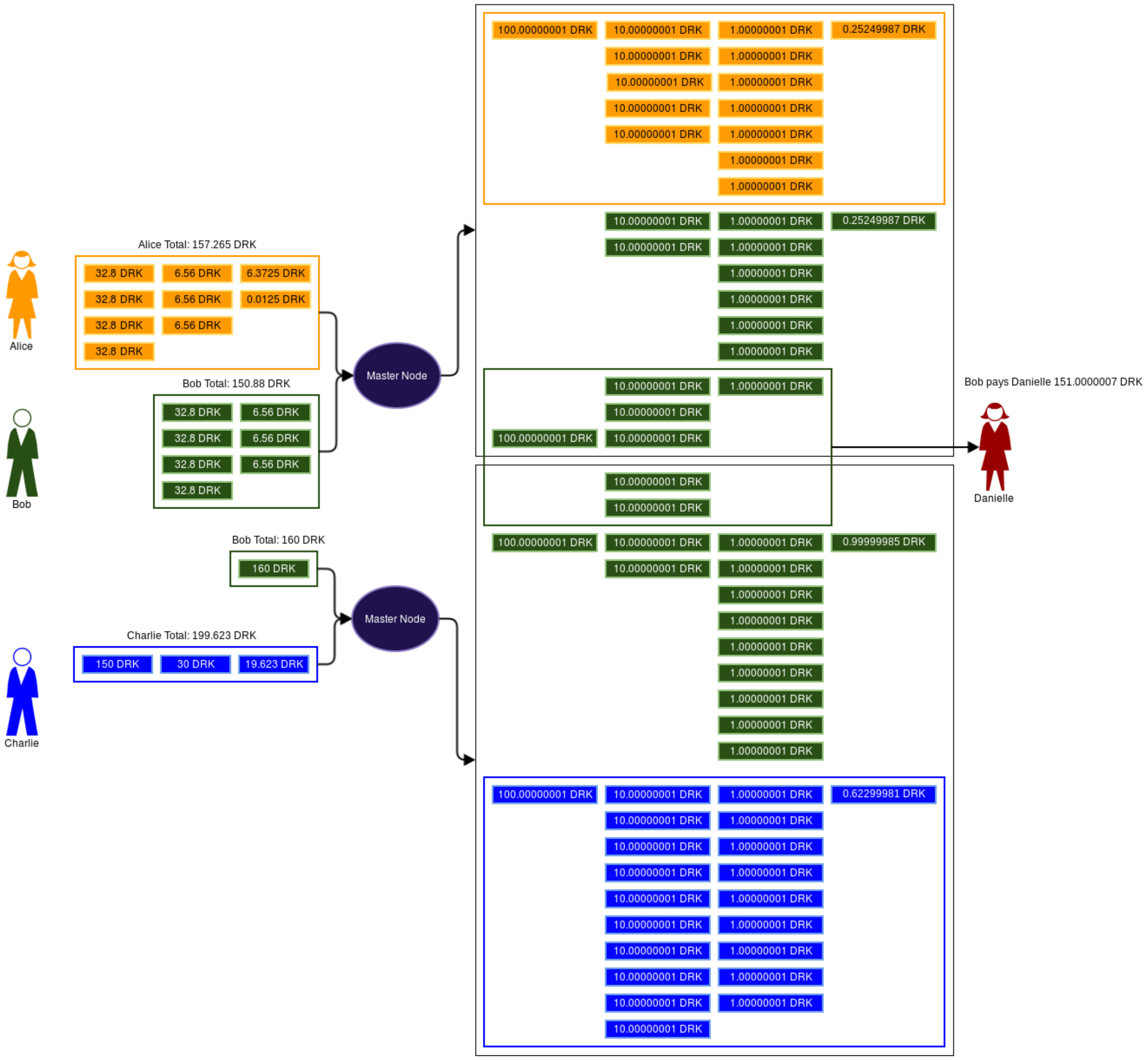


Figure 6: A conjoined spend weakness illustrated by funds spent from multiple groups of mixing peers.

In the example above, Bob is peered with Alice to mix 150.88 DRK, peered with Charlie to mix 160 DRK, and then sends 151.00000007 DRK to Danielle. The denominations he sends to Danielle come from peering with both Alice and Charlie separately. This reveals to any passive observer of the blockchain that at most 3 wallets were involved in the two Darksend+ mixing transactions, rather than the maximum of 4 that would normally be implied.

The conjoined spend weakness affects privacy in various ways depending on the type of observer:

Darkcoin Senders and Receivers: If a darkcoin recipient receives denominations from only a single group of mixed funds, this will only leak privacy information about the other users whose funds are mixed in to that group. If a recipient receives denominations from multiple groups, this leaks privacy information about other users from all of those groups.

Darksend Peers: When a malicious Darksend peer notices that his peer as conjoined funds funds they mixed together with other funds, this leaks information about the peers of those other transactions to the malicious Darksend peer. How much information is leaked depends on various factors including how many peers were included in the Darksend+ mixing transaction that the malicious peer participated in.

Masternodes: If users A and B are peered with a malicious Masternode, this typically only impacts users A and B’s funds directly. However, if B also spends impacted funds at the same time as he spends funds mixed with user C, the malicious Masternode knows significantly more about which funds belong to user C.

Passive Blockchain Observers: As described earlier, this weakness allows a passive observer to ascertain that a single wallet is associated with multiple Darksend+ mixing transactions, marginally reducing the privacy of all users involved.

Darksend+ can be improved nominally by trying to denominations funds from the same mixing group whenever possible, and by avoiding the conjoining of the 0.0125 DRK denominations split off for mining fees. It might also be helpful to users to do either or both of the following:

1. Warn users when spending will result in conjoining denominations from multiple mixing groups.
2. Provide a checkbox option to users to either prevent conjoined spending, or to put their funds through an additional round of Darksend+ while acting as their own peers, nullifying the privacy impact to other users. This would have to be carefully implemented, since this extra round of Darksend+ might stand out on the blockchain as an obviously just-in-time (JIT) operation, rather than the typical ahead-of-time operation typified by Darksend+. Applying an extra round of no-peers Darksend+ mixing unconditionally once other rounds have completed may remedy this.

Output Index Bias Weakness

Imagine if you implemented CoinJoin on a crypto-currency called COIN. Input funds are accepted from wallet clients, and the output funds sent to their new addresses. Unfortunately, you also make a fatal flaw by instructing the output funds to be listed in the same order as the input funds were accepted. This would be considered an output index bias. Any observer of the blockchain should be unable to use the order in which output funds are listed to determine which input funds they correspond to.

Darksend+ combats this by listing output funds in a pseudo-random order in a mixing transaction. A future code review should confirm that a reasonable amount of entropy ensures the randomness of this ordering and that there are not subtle statistical biases.

Darksend Queue Gaming

As of Darkcoin version 0.10.12.17, Darksend peers looking to mix funds will use the following protocol [changelog]:

1. *Users now will join a random Masternode (1 of the entire list, just completely randomly).*
2. *Upon joining if it's the first user, the Masternode will propagate a message stating it's taking participants for a merge.*
3. *Another user will check that queue, if it's got a recent node, it will try that node first, otherwise it will go to step 1.*

Given this protocol, Sybil attackers and malicious Masternodes are incentivized to collude. Sybil attackers can make malicious Masternodes attractive to connect to by immediately connecting to them with highly compatible funds available for mixing while ensuring that its connection meets the standard of being sufficiently “recent.” This can be accomplished by bypassing client-side checks to connect to a random Masternode, or by repeatedly disconnecting (prior to pledging any collateral) from Masternodes until the desired, malicious Masternode has been connected to. This will have the effect of artificially drawing victim peers to malicious Masternodes.

Likewise, malicious Masternodes can potentially disconnect Darksend+ peers until a known Sybil cohort has peered with a victim.

It may possible to identify such behavior as unusual and uncharacteristic from normally behaving Masternodes and Sybil peers. Subscription based services could potentially publish a blacklist of nodes identified as suspicious.

Denial of Service Attacks

Attacks that fall into this category are intended to degrade the availability of the Darksend+ service. Attackers who utilize this category generally have one of two goals:

1. To disrupt the Darksend+ system entirely, discouraging users from taking advantage of its privacy protections.
2. To disrupt particular regions of the Darksend+ network, driving target users toward malicious Darksend+ peers or Masternodes for subsequent privacy attacks.

Autistic Peer Attack

An autistic peer attack [cyberbrain] occurs when a Darksend+ peer initiates the process of participating in a Darksend+ round with other peers, but does not complete the round. Usually, this means that the offending Darksend+ peer refuses to or is unable to sign the transaction.

Darksend+ currently mitigates this threat by imposing a collateral system. When peers first offer to join in a Darksend+ round, they submit a signed transaction that the Masternode can broadcast; doing so takes away a small amount of darkcoin (currently 0.025 DRK) from any uncooperative peer.

Uncooperative Masternode Attack

When peers connect to an eligible Masternode, that Masternode commits to help them put together a Darksend+ mixing transaction. Uncooperative Masternodes can string these peers along until the last moment, refusing to actually publish the transaction for each peer to sign off on, causing the process to time out. Because Masternodes are chosen at random by peers, it is unlikely that an uncooperative Masternode will be chosen twice in a row.

Masternodes that fail to present a valid transaction, with information modified or omitted from what the peers are expecting will simply cause the peers' Darkcoin clients to refuse to sign the transaction. This is consistent with the “trust-less” design of the CoinJoin protocol.

Masternodes can maliciously publish the collateral transactions regardless of peer behavior. Identifying such misbehaving Masternodes is a difficult task. The Darkcoin network could ban the IP address of the Masternode, but IP addresses are fairly disposable. The network could also blacklist the 1000 DRK held in reserve to enable that Masternode, but the funds could be obfuscated using Darksend+ itself, and used to instantiate another Masternode controlled by the attacker once the 1000 DRK are reconstituted.

Conclusion

The core functionality of Darksend+ is a denominated, ahead-of-time CoinJoin implementation that introduces significant privacy improvements over Bitcoin. These improvements are available not only to Darkcoin users, but also users of other crypto-currencies willing to hold darkcoins only temporarily for the purposes of mixing funds.

A number of variables must be carefully selected in order to practically limit the size of Darksend+ transactions and “blockchain bloat,” such as the number of Darksend+ peers per round, the number and quantities of denominations, etc. Note that the majority of Darkcoin transactions do not necessarily use Darksend+, such as exchange transactions, mining, mining pool payouts, web services, etc. Considerations for whether Darksend+ is scalable in the future are beyond the intended scope of this document.

This document identifies a few weakness that can be improved in subsequent version of Darksend+, including “blinding” Masternodes, eliminating the fat sum denomination weakness, and proposals to limit how frequently the conjoined spend weakness is expressed. The following table summarizes the status of weaknesses and attacks explored in the document:

Attack/Weakness	Applicable To	Darksend+ Status	Remaining Impact	Fix Difficulty
Masternode Snooping	CoinJoin	Vulnerable	High	Easy
Sybil Attack	CoinJoin	Vulnerable	High	Difficult
Contextual Fingerprinting Attack	Bitcoin-like Currencies	Partially Mitigated	Low	Difficult
Significand Attack	Bitcoin-like Currencies	Partially Mitigated	Low	Difficult
Lonely Denomination Weakness	Darkcoin	Protected	N/A	N/A
Fat Sum Weakness	Darkcoin	Vulnerable	Medium	Easy
Disparate Spending Weakness	Bitcoin-like Currencies	Vulnerable	Low	Difficult
Conjoined Spend Weakness	Bitcoin-like Currencies	Partially Mitigated	Medium	Moderate
Output Index Bias Weakness	CoinJoin	Protected	N/A	N/A
Darksend Queue Gaming	Darkcoin	Vulnerable	Moderate	Difficult
Autistic Peer Attack	CoinJoin	Partially Mitigated	Low	Difficult
Uncooperative Masternode Attack	CoinJoin	Vulnerable	Low	Difficult

Sybil attacks in the form of malicious Darksend peers represent a cheap and effective threat to users of Darksend+. Requiring as few as two peers per mixing transaction means that Sybil attackers will have 100% success in reversing the positive effects of the mixing for transactions they participate in. Raising this to a three peer minimum would significantly increase the amount of work required for Sybil attackers, but must be weighed against potential blockchain bloat. *Update: Darkcoin's developers are experimenting with raising the minimum to three peers per transaction [RC5].*

A dedicated attacker on the Darksend+ network would likely use a combination of malicious Masternodes and malicious peers to game the queuing system and draw victim Darksend+ peers. Behavioral analysis and blacklist of suspicious nodes may make this more difficult for attackers.

References

[crowdfund]: http://www.reddit.com/r/DRKCoin/comments/2aw4wy/donations_for_darksend_code_review/
[Biryukov]: https://www.cryptolux.org/index.php?title=Bitcoin&oldid=1257
[Koshy]: http://fc14.ifca.ai/papers/fc14_submission_71.pdf
[CoinJoin]: https://bitcointalk.org/index.php?topic=279249.0
[Matonis]: http://www.coindesk.com/taxonomy-bitcoin-mixing-services-policymakers/
[CoinJoin Sudoku]: http://www.coinjoinsudoku.com/advisory/
[birth]: https://darkcointalk.org/threads/the-birth-of-darkcoin.162/
[Duffield]: https://www.darkcoin.io/downloads/DarkcoinWhitepaper.pdf
[maaku]: https://bitcointalk.org/index.php?topic=291283.0
[questions]: https://darkcointalk.org/threads/kristiv-atlas-want-to-ask-some-questions.1144/
[Jeng]: http://ojs.academypublisher.com/index.php/jnw/article/viewFile/0508921928/2053
[CoinShuffle]: http://crypsys.mmci.uni-saarland.de/projects/CoinShuffle/coinshuffle.pdf
[Yu]: http://www.math.cmu.edu/~adf/research/SybilGuard.pdf
[mantissa]: http://en.wikipedia.org/wiki/Significand#Use_of_.22mantissa.22
[Reid]: http://arxiv.org/pdf/1107.4524.pdf?origin=publication_detail
[changelog] https://raw.githubusercontent.com/darkcoinproject/darkcoin-binaries/master/CHANGELOG.md
[cyberbrain] http://ghostintheshell.wikia.com/wiki/Cyberbrain
[RC5] https://darkcointalk.org/threads/rc5-testing.2245/page-22#post-19850

Figures

Figure 1: http://cdn.anonymousbitcoinbook.com/darkcoin/darksend-paper/fig1.png
Figure 2: http://cdn.anonymousbitcoinbook.com/darkcoin/darksend-paper/fig2.png
Figure 3: http://cdn.anonymousbitcoinbook.com/darkcoin/darksend-paper/fig3.png
Figure 4: http://cdn.anonymousbitcoinbook.com/darkcoin/darksend-paper/fig4.png
Figure 5: http://cdn.anonymousbitcoinbook.com/darkcoin/darksend-paper/fig5.png
Figure 6: http://cdn.anonymousbitcoinbook.com/darkcoin/darksend-paper/fig6.png

Contacting the Author

Email: author@anonymousbitcoinbook.com
Web: http://anonymousbitcoinbook.com
Blog: http://blog.anonymousbitcoinbook.com
Twitter: @anonymouscoin
BitMessage: BM-2cTJVeZ19piDB1sxavMhQs4t3kDS33ruBX