Zcash Threat Model and Network Privacy Assessment

The Zcash Foundation

August 3, 2020

1 Introduction

The amount of data leaked about individuals online is ever growing; and financial data in particular provides a highly granular lens about personal daily habits. Zcash is a cryptocurrency that ensures strong financial privacy for users by leveraging zero-knowledge proofs and randomization techniques [8] to protect against data leakage that could otherwise lead to deanonymization of a payer or recipient of Zcash. Use of zero-knowledge proofs in this manner ensures that the raw bytes of a Zcash *shielded* transaction cannot leak identifiable information about the payer, payee, or amount.

However, an adversary observing the Zcash network could perform passive or active attacks with the goal of linking users and their end recipients, even if that adversary cannot decrypt the Zcash shielded transaction directly. Consequently, ensuring additional mechanisms to protect against network-level adversaries seeking to deanonymize users is important.

In this technical report, we assess the capabilities of such nework adversaries, and present a more formal threat model for Zcash, with a focus on *shielded transactions*. We identify what is considered sensitive information in the Zcash ecosystem, potential adversaries, and enumerate possible attack vectors. We then consider several network privacy mechanisms and assess the extent to which these mechanisms improve the privacy and security of Zcash users. Finally, we identify near-term and long-term recommendations for improvements.

Organization. We begin in Section 2 by discussing background information useful to understanding Zcash and its threat model. In Section 3, we describe the difference between adversarial network models in blockchains as opposed to other applications such as web browsing or email. In Section 4, we introduce a threat model for Zcash. In Section 5 we describe the network privacy mechanisms that we include in our assessment, and in Section 6 we perform this assessment. In Section 7 we outline furture improvements we are planning to implement, and we conclude in Section 8.

2 Background

2.1 Zcash Shielded Transactions

While Zcash does allow for use of unshieleded transactions, in this technical report we focus entirely on the threat model assuming the use of shielded transactions. However, we will now briefly describe the difference between the two transaction types. As mentioned before, unshielded transactions are similar to Bitcoin transactions, and expose the pseudonyms of the payer, payee, and the amount paid. When this information is exposed to the network and persisted indefinitely to the Blockchain, it is effectively "Twitter for your bank account".

Shielded transactions, on the other hand, expose what is essentially a "one time pad" to a network observer. Because not only the transaction information is encrypted, the encryption process itself is randomized such that two transactions that have identical payers, payees, and transaction amounts result in bytes that are completely indistinguishable from randomly-generated bytes. In other words, given only a series of shielded transactions, an adversary would not be able to gain enough information to distinguish any information about the transaction plaintext.

2.2 Comparison of Zcash Privacy Expectations to Bitcoin

While Zcash provides the ability to make shielded transactions to completely hide the information contained within a transaction, users of Zcash can also make non-shielded transactions. Because Zcash is a fork of Bitcoin, non-shielded transactions consequently have effectively the same expectation of privacy as plain Bitcoin transactions, which had been proven to be insecure against de-anonymization attacks [9]. In this post, we'll evaluate the threat model for Zcash considering only shielded transactions.

While both Zcash and Bitcoin have future goals of stable Tor integration and other network-privacy mechanisms, this routing via an anonymity layer does not prevent attacks that examine the bytes of the transaction itself. So unshielded transactions in Zcash and all Bitcoin transactions leak information to a network observer, which can be exploited to perform deanonymization attacks.

2.3 Zcash Protocol for Producing and Receiving Transactions

Producing a Transaction (Spending Funds). A user wishing to spend funds can publish a new transaction to the network in several ways. The user could operate a *full node*, meaning this node also participates in all network behaviour, such as gossiping transactions, responding to queries for the current state of the network, etc.

The user can also spend funds via a *light client* by publishing their transaction to a *light wallet node*, which itself is a full node which additionally can handle light wallet functionality. Note that while publishing a transaction to a light wallet, the user exposes the fact that they are making a Zcash transaction (as the light wallet learns a particular transaction, along with the IP address of the user). However, the light wallet will not learn the identity of the receiver of the funds [7].

Receiving a Transaction (Accepting Funds). A user receiving a transaction similarly has two options for how to receive these funds, either via a full node or via a light wallet node.

An important point to emphasize that because Zcash is a broadcast protocol (all full nodes sync the full state of the network), a user receiving transactions via a full node will not expose which transaction they are interested.

Similarly, a user fetching transactions from a light wallet will also fetch all block headers and therefore not expose which transaction they are interested in. However, one slight exception exists for clients that wish to learn the full details of a transaction (such as the memo field). For this case, they must query the light wallet for those details separately. As such, the light wallet can link a recipient to a particular transaction in the case the user queries for extended transaction fields for a specific block [7].

3 Observations About Attacks in a Blockchain Setting

We now describe several observations that we made while assessing the network privacy of users of *cryptocurrencies*, or more notably users whose state is synced to a globally consistent data store such as a blockchain. We contrast these observations with threats that are typical for a applications where users are browsing the web, exchanging email, or engaging in a messaging platform, for example.

We assume the use of shielded transactions for the below analysis, as end-to-end correlation via bitwise linkability is simple in an unshielded setting.

Observation One: Consensus in blockchains adds more latency than mixnets that are tuned for web traffic. While different cryptocurrency networks have a range of consensus mechanisms, in Zcash and other proof-of-work (POW) networks today, the amount of time between when a transaction is published and when that transaction is included into the Zcash blockchain is upwards of one minute [**?**].

Hence, unless the mixnet imposes more latency than the blockchain consensus protocol itself, sending transactions over a mixnet will not add additional meaningful protection against an adversary seeking to perform an end-to-end timing correlation attack.

Observation Two: Blockchains present a different landscape for attackers seeking to perform end-to-end correlation attacks than other application traffic. In a setting such as accessing content online or sending messages or emails, application traffic is sent from a sender to receiver, proxied through a series of servers. In this setting, an adversary could perform end-to-end correlation attacks in a number of ways, such as observing distinguishing identifiable patterns as information enters and leaves the network. The latency of these applications is important, as less timing can increase the possibility for timing correlation attacks, but so are other factors such as the pattern of packets or exposed metadata.

However, blockchains create a different landscape for an attacker that wishes to perform end-to-end correlation attacks. Namely, blockchains require that transactions are *first* included in the blockchain (the global state of the network), only *after* which the receiver of the funds can learn about the transaction.

Observation Three: Generating noise is more effective to prevent end-to-end correlation attacks in for web traffic than blockchains. Building on the prior observation that transactions in blockchains must first be synced to the blockchain (the network global state) *before* the receiver can receive it, we can now examine how noise generated by the sender or receiver might impact users' overall privacy. Again, we consider an adversary that seeks to perform an end-to-end correlation attack.

As opposed to in systems like Loopix where noise (let's consider dummy packets) is generated by the sender, along with real packets. In Loopix, the destination of these dummy packets can depend on whether they are intended to be a "loop" message, meaning that the sender themselves is also the receiver, or if the dummy packet is sent to some arbitrary destination, which will simply discard the packet. In this setting, dummy packets are sent end-to-end, and consequently an attacker observing the edges of the network cannot perform a simple correlation between packets entering and leaving the network.

Conversely, dummy transactions in a blockchain setting will never make it through to a receiver (without encoding the possibility that a transaction is for zero value, which opens the door to DoS attacks). Hence, such noise in a blockchain setting will never be end-to-end. While we cannot give concrete estimates as to how less effective noise that is dropped before transactions are added to the blockchain is as opposed to end-to-end noise that is possible for other applications, we simply wish to point out that security claims about added noise to prevent end-to-end correlation attacks do not map neatly into blockchain settings from other settings such as for web traffic.

Takeaways While adversaries can certainly perform end-to-end attacks in a blockchain setting, the above observations indicate that doing so requires different techniques than in other application settings such as for web traffic or email applications. Hence, we consider end-to-end passive fingerprinting as out of scope for the threat model of Zcash, assuming the use of shielded transactions.

4 Zcash Threat Model

4.1 Analysis of Sensitive Information Exposure

As described in Section 2.1 an adversary could gain more information about a shielded transaction by observing information exposed to the network. Keeping this in mind, and assuming that a transaction is shielded, we now review sensitive information that could be exposed and used for malicious purposes by a adversary. We divide these information leaks into *in scope* to the Zcash threat model, and *out of scope*.

4.1.1 In Scope

Note that an adversary has the ability to observe both *on-chain* visible data as well as information or behaviour that is exposed to the network during the process of submitting a new transaction or receiving a transaction.

Linking (sender identity, transaction, receiver identity): Specifically, this three-tuple of sender identity, transaction, and receiver identity allows for an adversary to link the fact that a specific sender of Zcash made a payment to a specific receiver, even though the amount of the payment is not disclosed. While the sender and receiver identity is not exposed by the Zcash transaction itself, an adversary could gain this information by observing the network or colluding with one of the involved parties. For example, if the recipient is malicious and colludes with a party that can link the sender to their transaction, or if the recipient leaks to a provider which transaction they are interested in, the sender and receiver could be linked to a particular transaction.

Unique fingerprints of user habits, coupled to sender or receiver identity: By observing behavior of Zcash transactions sent over the network, an adversary could start to build a "fingerprint" of a user's behavior, partially when using machine learning algorithms to classify traffic. Information that is exposed in this category includes the time of day that a transaction occurs, the frequency of transactions, and even the route that a transaction passes through from sender to receiver.

Further, "on-chain" information could also allow for fingerprintability. For example, fees and timing of transactions could possibly provide additional information to an observer.

4.1.2 Deferred Threats for Future Improvements

Malicious Senders or Receivers of Zcash: Senders or receivers of Zcash can act honestly within the protocol but maliciously outside of the protocol (such as creating a "honeypot" for which Zcash users are tricked into sending valid transactions to).

While preventing malicious senders or receivers of Zcash from compromising the other part is certainly within the threat model of Zcash, we defer addressing this category of threats for future improvements.

Partitioning attacks. The Zcash network itself could be partitioned, such that some nodes think they are aware of the entire network, but only instead be aware of a small subset of nodes. Such attacks could be performed by DNS seeders (again, by lying about the state of the network), or even by highly-connected nodes. Such attacks are well-known in the literature and in practice and demonstrated to be practical.

4.1.3 Out of Scope

Learning the tuple (sender or receiver identity, transaction): Note that a network observer can also gain a subset of sender/receiver linkability information by observing just a sender *or* receiver's identity linked to a specific transaction. We consider this use case to be out of scope to the threat model of Zcash, as such information—to the best of our assessment—simply leaks that the sender or receiver is participating in sending and receiving Zcash, without any further details. Consequently, we consider such an information leak to be out of scope.

Note that learning the tuple such as (sender, transaction) along with additional information such as the time that a transaction was made can possibly leak undesirable information. However, we consider this to be out of scope for our current threat model until we can identify mechanisms that both ensure usability while meaningfully improving user privacy.

Unforeseen Software Flaws: While our team developing Zcash software works diligently to prevent software bugs and vulnerabilities to the best of our ability, we do not consider such flaws in scope to our threat model, as such cases are unforeseen and we cannot rule them out completely. As such, when we become aware of vulnerabilities or flaws, we will fix them, but we cannot claim such occurrences will not occur in the future.

User Behavior Outside of the Zcash Protocol: Zcash is not used in a vacuum; spending and receiving Zcash is linked to real-world value. As such, we do not consider user behavior outside of the Zcash protocol to be in scope to our threat model, even if that behavior results in spending or receiving Zcash. For example, someone who goes to a website and wishes to purchase an item in Zcash may be unlinkable purely when observing the Zcash network, but their behavior when browsing the website can still be observable to a network attacker. In order to spend Zcash in a *truly* private way, the user must use another privacy-preserving technology to hide their spending behavior outside of Zcash, such as accessing that website over Tor.

Block Access Patterns: Even if a user accessed information about the Zcash blockchain using an anonymity tool such as Tor, the access patterns for specific blocks could leak information to network adversaries, such as the time of day that users requested information about that block, or its popularity. However, preventing such information leaks can likely only be solved using a full broadcast system, which is already an option for full Zcash notes. For optimizations, we do not consider the access patterns on blocks to be in scope to the Zcash threat model.

4.2 Attacks that are not Applicable for Shielded Transactions

Epistemic attacks. A network observer could perform an epistemic attack by observing unique routing information that allows for eventual deanonymization of that user. Again, such attacks are possible in Zcash because users do not control routing information for their transaction.

One example of epistemic attacks against cryptocurrency networks involve the probability of "super-connected" nodes that can link the node from which a transaction originated [11, 5].

However, let's consider the case when only shielded transactions are used. In such a setting, the recipient of the transaction will not be disclosed if they are using a full node, and only disclosed if they fetch additional parameters if using a light wallet (such as by fetching the Zcash memo field). As such, since our analysis only considers the case where shielded transactions are used, we deem epistemic attacks such as those presented by the Dandelion authors as not applicable for Zcash.

Fingerprinting attacks by Correlating End-to-End Timing of Transactions. Fingerprinting timing of when transactions are made can lead to deanonymizing the sender and receiver *only* in the light wallet setting, even if shielded transactions are used. We note that in the full node setting, such attacks are not possible due to the full broadcast nature once transactions are published to the Zcash blockchain.

However, timing-based fingerprinting can be performed by a malicious light wallet node or even a malicious or compromised recipient. In this setting, these parties could build a profile of the person making payments to them by observing the timing of their payments, even if the sender wishes to remain anonymous.

While light wallets could perform this kind of end-to-end timing attack, we consider such attacks out of scope considering the latency and batching that is already added by time required for a transaction to be included into the blockchain, as further described in Section 3.

Fingerprinting attacks by Correlating End-to-End Behavior of Transactions. Fingerprinting user behavior such as the frequency and number of transactions made can lead to deanonymizing senders and receivers in the setting where light wallets are used, even the transaction is shielded and timing information is removed (such as by sending the transaction through a mixnet). For example, a malicious light wallet node could observe the frequency and pattern of a user's transactions, and build a pattern to match against over time.

However, as described in Section 3, it is unclear how end-to-end behavior correlation could occur, even if an adversary has a complete view of the network, considering how all transactions must first be synced to the network before a receiver can receive the funds. At minimum, a transaction will have the anonymity set of the size of one block, but this assumes a transaction is immediately synced to the blockchain; in reality, additional latency and randomness will be incurred due to proof of work requirements or network topologies that add latency to gossip protocols.

4.3 Adversarial Model

We now review possible adversaries that may be motivated to compromise Zcash users' privacy.

Regular Zcash user. Allowed to send and receive transactions and act outside the protocol, just as a real user.

Regular Zcash Node Operator. Can operate one (or many) full Zcash nodes, and can store and forward all traffic that is sent through the node. Can query for network information, and store and examine all information it receives.

DNS seeder operators. Operates the node that is used by new nodes when bootstrapping to the Zcash network, in order to learn about other nodes in the network to begin communicating with them directly.

Internet Service Providers (ISPs). Can view traffic sent and received from either users or Zcash nodes. Can store observed traffic, forward traffic to other parties, and compare traffic with other traffic.

Government actors. Can issue secret subpoenas and force ISPs and regular Zcash users to take actions they may not wish to take, such as turning over secret keys or server logs.

4.4 Attack Vectors

We now review known attacks in the literature that have been described for decentralized systems that are applicable to Zcash even when shielded addresses are used. Notably, some these attacks leverage decentralized networks where information about the network may or may not be consistent, or require the use of light wallets in a setting where user behavior can be differentiated.

Denial of service attacks. Such attacks could be performed against Zcash nodes, such as DNS seeders refusing to respond to certain queries, or against Zcash users, such as light wallets refusing to service certain classes of IP addresses.

On-Path end to end correlation attacks. This attack could occur by an adversary that can obtain the (sender identity, transaction, receiver identity) tuple discussed in Section 4. By *on-path end to end correlation*, we include attacks performed by nodes that are directly on the path between a sender and receiver when publishing or receiving a transaction. In the current model of Zcash, such an attacks are practical only when the recipient uses a light wallet in such a manner that leaks the transaction that the sender or receiver is publishing, as further described in Section [7].

5 Review of Network Privacy Approaches

Ideally, some of the attacks described in Section 4.4 could be mitigated by using a network privacy layer. We now review three classes of network privacy approaches, and then in Section 6, determine how these approaches address the existing described attacks against Zcash.

For brevity, we only review Dandelion [6, 2], Tor [4], and Loopix-based mixnets [10]. Further, we review private information retrieval (PIR) as a method which can be used in conjunction with the above systems.

Dandelion. Dandelion [2] and Dandelion++ [6] is a lightweight gossip protocol aimed at adding additional network privacy for distributed networks such as cryptocurrencies. Dandelion protects against *passive* deanonymization attacks, but does not consider active or targeted attacks. Such passive deanonymization attacks could be conducted by a "super node" that has a high degree of connections to other nodes (and could either be a single node or a botnet where adversarial machines share information). As such, it is assumed that this adversary is honest-but-curious, following the gossip protocol but wishes to learn as much information about users as it can directly observe. However, Dandelion++ does consider a stronger adversarial model where nodes are allowed an arbitrarily-number of connections to other nodes (acting outside of the Bitcoin gossip spec which only allows 8).

Both Dandelion variants follow a randomized design for how transactions are gossiped to the network, differing from the Bitcoin design where nodes publish transactions as widely as possible as quickly as possible. In the Dandelion design, whenever a node receives a transaction from a neighbor, it first flips a coin to determine if the traffic is sent to a single neighbor (constituting the "stem" phase) or to all the node's connections (constituting the "fluff" phase). Such an approach frustrates the ability for a super node to link a specific transaction to the node which originally published that transaction.

Tor. Tor [4] is an anonymity network that today has over 2.5 million users and a network size of over 6,500 nodes. Tor supports applications that require low latency, such as browsing the Internet anonymously or streaming videos. In order to support such a use case, Tor assumes that it is hard for network adversaries to gain an end-to-end view and provide correlation attacks, such as by injecting timing or dropping packets to test if the traffic it can view entering the network is the same as the traffic it can view leaving the network.

Tor distributes its routing information (i.e, information about each relay) via an authenticated document called the *consensus*, which is signed by a threshold number of trusted servers called *directory authorities*. In doing so, Tor ensures that all clients and relays have a consistent view of the network. By distributing a global authenticated document to all network participants, Tor avoids epistemic or routing attacks unlike completely decentralized networks which cannot guarantee a user or relay's view of the network is authentic or that all users fall within a global anonymity set.

Loopix-based Mixnets. While a range of mix network designs have been introduced in the literature, in this assessment we consider only those which instantiate the Loopix [10] design, which provides improved latency guarantees over prior designs. Similar to other mix network designs, Loopix uses dummy packets and message delays in order to protect against adversaries performing fingerprinting and end-to-end attacks of user traffic.

While Loopix specifies how messages are routed through a network, Loopix-based networks still require safely distributing network information to users and nodes. As one example, Katzenpost [1], a mix network which implemented Loopix, used a similar model to Tor by leveraging trusted network authorities to sign and distribute the state of the network. However, alternative network distribution mechanisms can be used, but similarly may be subject to epistemic and path-routing attacks similar to other distributed networks.

Private Information Retrieval (PIR). Even though the above network anonymity systems disassociate a sender or receiver's identity from the transaction, nodes can still observe which blocks are being fetched and perform fingerprinting attacks using this information. One mechanism that can be used in conjunction with network anonymity tools is private information retrieval (PIR) [3], which allows users to query information while preventing the service that hosts this information from learning the query.

Note here we do not specify *which* PIR design to use, we simply assume the properties of a PIR implementation, for which the content of the user's query is hidden from the service tasked with responding to that query.

6 Assessment of Network Privacy Approaches to Zcash Privacy (Assuming the Use of Shielded Transactions)

We now review the extent to which Dandelion++, Tor, some generic PIR mechanism, and a Loopix-based mixnet protect against the existing network-based attacks against Zcash described in Section 4.4. Note that we assume an attacker does not control either the sender or receiver of funds.

We summarize our results in Table 1, but describe our findings here.

Assumptions. We assume the mixnet is a seperate anonymity network entirely, such as in the style of Nym, and is *not* part of the cryptocurrency network itself (meaning that the initiator of a transaction forwards this transaction through the mixnet before it is published to the cryptocurrency network).

We assume that the PIR mechanism is used by the user to look up transactions in such a way that the node serving that transaction cannot infer the contents of the user's query.

Table 1: Effectiveness of network privacy mechanisms for Zcash security and privacy, assuming the use of shielded transactions

●=protects against; ○=does not protect against; ★=Not a threat in this setting;
Mixnet=Loopix-based routing capable of delaying packets but not creating dummy transactions;
DoS=Denial of Service;
On-Path Correlation=End to end correlation attacks where one node in the path is participating in the attack.

						PIR	
	Attack	Dandelion++	Tor	Mixnet	Only PIR	Tor	Mixnet
Full Node	DoS	○	○	○	○	○	○
	On-Path Correlation	★	★	★	★	★	★
Light Client	DoS	○	○	○	○	○	○
	On-Path Correlation	○	○	○	○	●	●

6.1 Analysis for Use of Full Nodes

Dandelion++. In the full node setting when shielded transactions are used, Dandelion++ does not meaningfully change privacy guarantees for users, as Dandelion++ provides privacy in the setting of a super node with knowledge of the network graph. Because receivers of Zcash transactions operate in a full broadcast setting when using a full node, Dandelion++ does not add meaningful additional privacy guarantees, where simply examining a transaction cannot reveal any additional information. Further, the protocol does not add any protection against denial of service attacks.

Tor. Again, in the setting where full nodes are used, the receiver cannot be linked to a specific transaction, unless the receiver themselves is malicious and colludes with other parties. As such, while Tor can hide which transaction a sender made, in the setting where shielded transactions are used and receivers receive all transactions in a full broadcast model, the use of Tor does does not meaningfully change user privacy.

Mixnets. While the use of a Loopix-based mixnet raises the cost to an adversary to conduct fingerprinting attacks that could result in end-to-end linking of senders and receivers, this protection does not meaningfully increase user security in the full node setting or in light of the observations of the inherent protections that blockchains offer against these attacks described in Section 3. Most specifically, because receivers operate in a full-broadcast mode, end-to-end linking of senders and receivers is not possible when full nodes are used, unless the receiver themselves is compromised or malicious.

As such, assuming receivers are honest, then mixnets do not meaningfully provide additional privacy or security to Zcash users over Tor when operating in the full node setting.

Only PIR. Because the receiver in a full node setting receives all transactions, use of PIR does not add any additional meaningful protections when shielded transactions are used. This is because the receiver has the full set of transactions themselves and does not need to perform a lookup to another server for this information.

Tor + PIR. Because receivers of Zcash when using a full node operate in full broadcast receiver mode—meaning that the user fetches the complete state of the network and consequently does not leak which transaction they are interested in—use of PIR in this setting does not meaningfully add privacy or security protections.

As such, use of Tor + PIR in the setting does not add meaningful privacy protections when shielded transactions are used.

Mixnet + PIR. Again, because users operate with full nodes, even though the sender's identity is hidden and so are any timing or behavior leaks, use of mixnets with PIR does not add meaningful security when shielded transaction are used.

6.2 Analysis for Use of Light Wallets

Dandelion++. While useful in a non-shielded context, Dandelion++ does not protect against end-to-end correlation attacks in the setting where light wallets are used. Specifically, while Dandelion++ protects against super nodes that can observe in-network gossip messages, Dandelion++ does not provide "last-mile" privacy, meaning that users can still be deanonymized by the light wallets they use.

Tor. Tor protects against a light wallet directly distinguishing between receivers in a light wallet setting, as the light wallet will not be able to learn the network identity (IP address) of a receiver.

Because Tor is a low-latency network, even if the receiver IP address is hidden from the light wallet, the light wallet or a network observer can still observe the user's timing and behavior when fetching transaction details. However, because a user's transaction is published to the Zcash network, and then published as the next block in the Zcash blockchain with a series of other transactions, it is unclear how much of an advantage an adversary has to perform end-to-end timing-based attacks in this setting (as further discussed in Section 3). Typically end-to-end timing attacks require performing packet correlation on the order of seconds, whereas the latency in a network like Zcash is on the order of minutes. As such, we deem timing-based attacks to be not applicable even when light wallets are used for Zcash.

However, the light wallet can still observe which queries a receiver wishes to fetch, even if the light wallet cannot observe that receiver's network identity. If the light wallet provided service to both the sender and receiver, it could possibly perform a linking attack based purely on this information. Hence, we do not grant the property of protecting against end-to-end correlation attacks by on-path nodes.

Mixnets. Because blockchains require publishing transactions to a global immutable data store (the blockchain), the property of mixnets to inject dummy packets to help discourage end-to-end packet correlation in a web traffic setting is less relevant in a blockchain setting.

Further, as described in Section 3, the ability of an adversary to use timing of transactions to infer information about the sender, receiver, and contents is less relevant in a blockchain setting due to the inherent latency incurred when publishing a transaction. While mixnets do hide timing information for packets by delaying messages sent through the mix network, in systems tuned for web or messaging application traffic, such delays are on the order of seconds or even microseconds, where as we observed above, the Zcash network itself imposes delays on the order of minutes as transactions are added to the Zcash blockchain. Performing an end-to-end correlation attack simply by observing the endpoints (nodes servicing the sender or receiver of the transaction) will have difficulty performing such a timing correlation even in Zcash today.

Similarly to the description of Tor above, a light wallet can observe which queries a receiver wishes to fetch, and so a malicious light wallet could use additional information to perform linking attacks. Hence, we do not grant the property of protecting against on-path end-to-end correlation attacks simply by use of mixnets alone.

Only PIR. PIR in the setting of Zcash ensures that a receiver can hide which transaction they are interested in learning. However, a malicious on-path light wallet could still perform end-to-end correlation attacks by observing the sender and receiver's identities, as well as link the sender to a specific transaction. Hence, we do not grant full protection against on-path correlation attacks when only PIR is used.

Tor + PIR. In the setting where senders and receivers use Tor along with PIR, we grant a full circle for protection against on-path correlation attacks. While nodes can learn that *someone* is sending and receiving a transaction, nodes cannot learn either *which* transaction, *nor* who the sender or receiver are.

Mixnet + PIR. Similarly to Tor, we grant a full circle for protection against on-path correlation attacks when a mixnet used in conjunction with PIR. Similarly to Tor, mixnets hide the identity of senders and receivers from the destination of their request by forwarding traffic along a series of hops, and so both systems achieve the same security properties for these reasons.

Note however that this protection is achieved simply by routing user's transactions among the mix network of independently-operated nodes, and the addition of message latency (to discourage timing fingerprinting) and dummy packets (to discourage behavior fingerprinting) does not add meaningful additional protections in this setting, as further described in Section 3.

7 Future Improvements for Zcash Network Privacy

7.1 Privacy Improvements for Unshielded Transactions

While the end goal for Zcash is to move all users to use shielded transactions, the reality today is that only a very small percentage of Zcash is transacted using shielded transactions. In the interim to achieving the goal of ubiquitous shielded transactions, we can take intermediate steps to improve the privacy for Zcash users in the unshielded setting.

With that said, incorporating a variant of the Dandelion++ gossip protocol is a lightweight and immediate first step to improving the privacy of unshielded transaction. However, we recognize that stronger protections are required in following steps to provide stronger security and privacy guarantees.

7.2 Privacy Improvements for Shielded Transactions

In looking at Table 1, the strongest protections are offered by the combination of PIR along with a network anonymity tool such as Tor or a mixnet. Because use of an anonymity network protects against directly leaking the network identity of a user, and PIR protects against disclosing the contents of a user's query, the combination of these methods leaves little room for an attacker to gain advantage.

While typically the ability for mixnets to offer protection against timing or behavior fingerprinting attacks provides some competitive advantage over Tor, in the setting where shielded transactions are used along with PIR, this advantage becomes minimal. It is unclear what advantage an attacker could gain simply by observing network patterns, without also learning the contents of a user's query or a user's identity, particularly when all traffic is for the same higher-level application—that of fetching transactions from the Zcash blockchain—as opposed to web traffic, which can differ depending on the website visited or application used.

As such, our immediate-term plan to proceed with Tor, which is immediately deployable and has mature security analysis and availability. We will then assess PIR as a second step. We will continue to observe the maturity of mixnets as a possible option in the future.

8 Conclusion

In this technical report, we sketched out a more formalized threat model for users of Zcash, and identified how this threat model differs between a setting where users exchange shielded versus unshielded transactions. We then assessed the extent to which well-known network anonymity tools improve protections for Zcash users in the setting where *shielded* transactions are used. Finally, we described next steps to improve user privacy in both a shielded and unshielded setting, starting with lightweight improvements to further longer-term structural changes.

References

[1] Yawning Angel, George Danezis, Claudia Diaz, Ania Piotrowska, and David Stainton. Katzenpost Mix Network Specification. https://katzenpost.mixnetworks.org/docs/specs/mixnet.html, 2017. last accessed 2019-12-16.

[2] Shaileshh Bojja Venkatakrishnan, Giulia Fanti, and Pramod Viswanath. Dandelion: Redesigning the Bitcoin Network for Anonymity. *Proc. ACM Meas. Anal. Comput. Syst.*, 1(1):22:1–22:34, June 2017.

[3] Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private information retrieval. *J. ACM*, 45(6):965–981, November 1998.

[4] Roger Dingledine and Nick Mathewson. Tor Protocol Specification. https://gitweb.torproject.org/torspec.git/tree/tor-spec.txt, 2020. last accessed 2020-06-22.

[5] Giulia Fanti, Shaileshh Bojja Venkatakrishnan, Surya Bakshi, Bradley Denby, Shruti Bhargava, Andrew Miller, and Pramod Viswanath. Dandelion++: Lightweight cryptocurrency networking with formal anonymity guarantees, 2018.

[6] Giulia Fanti, Shaileshh Bojja Venkatakrishnan, Surya Bakshi, Bradley Denby, Shruti Bhargava, Andrew Miller, and Pramod Viswanath. Dandelion++: Lightweight Cryptocurrency Networking with Formal Anonymity Guarantees. *Proc. ACM Meas. Anal. Comput. Syst.*, 2(2):29:1–29:35, June 2018.

[7] Jack Grigg George Tankersley. Light Client Protocol for Payment Detection. https://github.com/gtank/zips/blob/light_payment_detection/zip-XXX-light-payment-detection.rst, 2018. last accessed 2020-05-29.

[8] Daira Hopwood, Sean Bowe, Taylor Hornby, and Nathan Wilcox. Zcash Protocol Specification. https://github.com/zcash/zips/blob/master/protocol/protocol.pdf, 2020. last accessed 2020-05-29.

[9] Philip Koshy, Diana Koshy, and Patrick McDaniel. An analysis of anonymity in bitcoin using p2p network traffic. In Nicolas Christin and Reihaneh Safavi-Naini, editors, *Financial Cryptography and Data Security*, pages 469–485, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[10] Ania M. Piotrowska, Jamie Hayes, Tariq Elahi, Sebastian Meiser, and George Danezis. The Loopix Anonymity System. In *Proceedings of the 26th USENIX Conference on Security Symposium*, SEC'17, pages 1199–1216. USENIX Association, 2017.

[11] Shaileshh Bojja Venkatakrishnan, Giulia Fanti, and Pramod Viswanath. Dandelion: Redesigning the bitcoin network for anonymity, 2017.