Zcash Threat Model and Network Privacy Assessment

The Zcash Foundation

June 28, 2020

1 Introduction

The amount of data leaked about individuals online is ever growing. Financial data in particular provides a highly granular lens about personal daily habits, but securing one's own financial privacy is difficult due to the value of tracking financial habits for advertising purposes. Fortunately, Zcash ensures strong financial privacy using zero-knowledge proofs along with randomization techniques [7] to protect against data leakage that could lead to deanonymization of a payer or recipient of Zcash. This approach 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 network over which Zcash clients and nodes send transactions 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, it is important to assess the how such adversaries could be successful, what information is required in order to perform such deanonymization attacks, and what such adversaries require to do so.

In this technical report, we consider the above questions, and present the threat model for Zcash more formally, assuming the use of shielded transactions. We identify what constitutes sensitive information in the Zcash ecosystem, what adversaries may exist, and enumerate the attack vectors that such adversaries may employ. From this model, we then consider several different network privacy mechanisms, and analyze the extent to which these mechanisms improve the privacy and security of Zcash users. Finally, we lay out several 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 introduce a threat model for Zcash, by discussing sensitive information that attackers could observe in the Zcash network, reviewing possible attacker profiles and their associated capabilities and powers, and finally reviewing a range of potential attack vectors.

[1]

[1] CK says: TODO add in the second half of this work after it is added

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 deanonymization attacks [8]. 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 [6].

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 [6].

3 Zcash Threat Model

3.1 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*.

3.1.1 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: 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.

3.1.2 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.

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.

3.2 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.

3.3 Attack Vectors

We now review known attacks in the literature that have been described for decentralized systems similar to Zcash, although not all attacks have been demonstrated against Zcash specifically. Notably, these attacks leverage decentralized networks where information about the network may or may not be consistent.

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 [10, 4].

Fingerprinting attacks. Fingerprinting user behavior when making a transaction can lead to deanonymizing that user, even if shielded transactions are used. Fingerprinting can be performed by a range of adversaries across many different settings. For example, a malicious light wallet node could observe the frequency and timing of a user's transactions, or even the recipient of a user's transaction could create a profile of the person making payments to them via timing and frequency of their payments, even if that user wishes to remain anonymous.

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.

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.

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 3. Such an attack is possible only when the recipient uses a light wallet in such a manner that leaks the transaction that the recipient is interested in, as further described in Section [6]. Otherwise, as all recipients download information about every transaction via a broadcast-like protocol, and hence an adversary cannot gain a true end-to-end view, unless the recipient themselves is compromised or malicious.

4 Review of Network Privacy Approaches

Ideally, some of the attacks described in Section 3.3 could be mitigated by using a network privacy layer. We now review three classes of network privacy approaches, and then in Section 5, determine how these approaches address the existing described attacks against Zcash.

For brevity, we only review Dandelion [5, 2], Tor [3], and Loopix-based mixnets [9].

Dandelion. Dandelion [2] and Dandelion++ [5] 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 [3] 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 [9] 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.

5 Assessment of Network Privacy Approaches to Zcash Privacy

We now review the extent to which Dandelion++, Tor, and a Loopix-based mixnet protect against the existing network-based attacks against Zcash described in Section 3.3.

Note that we assume an attacker does not control either the sender or receiver of funds. We assume the mixnet only delays transactions and does not generate dummy packets (as generating dummy transactions that are verifiable require higher-level application support, and cannot be a property entirely provided by the mixnet itself).

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.

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

References

[1] Yawning Angel, George Danezis, Claudia Diaz, Ania Piotrowska, and David Stainton. Katzenpost Mix Network Specification. https://katzenpost.

Table 1: Effectiveness of network privacy mechanisms for Zcash security and privacy

●=protects against; ○=does not protect against; ?=unknown; ◇=Not a threat;
Mixnet=Loopix-based routing;

Attack	Full Node			Light Wallet		
	Dandelion++	Tor	Mixnet	Dandelion++	Tor	Mixnet
Epistemic	●	●	?	X	X	X
Timing Fingerprinting	◇	◇	◇	X	X	X
Content Fingerprinting	○	○	○	X	X	X
Denial of Service	○	○	○	○	○	○
Passive end to end correlation	○	●	●	X	X	X

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

[4] 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.

[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. *Proc. ACM Meas. Anal. Comput. Syst.*, 2(2):29:1–29:35, June 2018.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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