Attacking Zcash Protocol For Fun And Profit
Whitepaper Version 0.1

Duke Leto + The Hush Developers[†]

May 5, 2020

Abstract.

This paper will outline, for the first time, exactly how the "ITM Attack" (a linkability attack against shielded transactions) works against Zcash Protocol and how Hush is the first cryptocoin with a defensive mitigation against it, called "Sietch". Sietch is already running live in production and undergoing rounds of improvement from expert feedback. This is not an academic paper about pipedreams. It describes production code and networks.

We begin with a literature review of all known metadata attack methods that can be used against Zcash Protocol blockchains. This includes their estimated attack costs and threat model. This paper then describes the "ITM Attack" which is a specific instance of a new class of metadata attacks against blockchains which the author describes as "Metaverse Metadata" attacks.

The paper then explains Sietch in detail, which was a response to these new attacks. We hope this knew knowledge and theory helps cryptocoins increase their defenses against very well-funded adversaries including nation states and chain analysis companies.

A few other new privacy issues and metadata attacks against Zcash Protocol coins will also be enumerated for the first time publicly. The ideas in this paper apply to all cryptocoins which utilize transaction graphs, which is to say just about all known coins. Specifically, the Metaverse Metadata class of attacks is applicable to all Bitcoin source code forks (including Dash, Verge, Zerocoin and their forks), CryptoNote Protocol coins (Monero and friends) and MimbleWimble Protocol (Grin, Beam, etc) coins but these will not be addressed here other than a high-level description of how to apply these methods to those chains.

> In privacy zdust we trust.
>
> If dust can attack us, dust can protect us.
>
> – Sietch Mottos

Keywords: anonymity, zcash protocol, cryptographic protocols, zk-SNARKs, metadata leakage, deanonymization, electronic commerce and payment, financial privacy, zero knowledge mathematics, linkability, transaction graphs, shielded transactions, blockchain analysis .

Contents

1 **1 Introduction** **3**

[†] myhush.org, https://keybase.io/dukeleto, F162 19F4 C23F 9111 2E9C 734A 8DFC BF8E 5A4D 8019

2 Metadata Analysis of Zcash Protocol Blockchains: Basics **3**

2.1 Concepts and Definitions . . . 3

2.2 Types Of Shielded Transactions . . . 3

3 Metadata Analysis of Zcash Protocol Blockchains: Advanced **4**

3.1 Active vs Passive Attacks/Analysis . . . 4

3.2 Timing Analysis . . . 4

3.3 Value Analysis . . . 4

3.4 Fee Analysis . . . 4

3.5 Dust Attacks . . . 5

3.6 Input/Output Arity Analysis . . . 5

3.7 Exchanges and Mining Pools . . . 5

3.8 What does the explorer not show? . . . 5

4 De-anonymization techniques literature review **5**

4.1 Applications to new Shielded-only Chains . . . 5

5 ITM Attack: z2z Transaction Linkability **5**

5.1 ITM Attack: Assumptions . . . 6

5.2 ITM Attack: Defeating ***ZK-SNARKs*** . . . 6

5.3 ITM Attack: Infrastructure . . . 7

5.4 ITM Attack: Consensual Oracles . . . 7

6 Metaverse Metadata Attacks **7**

7 Sietch: Theory **8**

8 Sietch: Code In Production **8**

9 Thoughts On Device Seizure **9**

10 Advice To Zcash Protocol Coins **9**

11 Special Thanks **9**

12 References **9**

1 Introduction

2 Metadata Analysis of Zcash Protocol Blockchains: Basics

2.1 Concepts and Definitions

This paper will be concerned with **transaction graphs**, which we define in the traditional mathematical sense, of a set of nodes with a set of vertices connecting nodes. In cryptocoins these always happen to be directed graphs, since there are always funds which are unspent becoming spent, i.e. a direction associated with each transaction. This direction can be mathematically defined using the timestamp of the transaction. Inputs are unspent at the time of the transaction and outputs are spent at the time of the transaction.

There is a great deal of mathematical history devoted to the study of **graph theory** that has not been applied to blockchain analysis, mostly because there was no blockchains to analyze just a few years ago and there was no financial profit in studying the data. That has obviously drastically changed.

This paper will be primarily concered with **shielded transaction graphs** which are **directed acyclic graphs (DAGs)**. A ***shielded*** transaction does not reveal the address of Alice, nor Bob, nor the amount transacted but it does leak a large amount of metadata at the protocol level, which is not rendered by block explorers nor well understood by the industry.

A ***shielded*** transaction has at least one ***shielded*** address, referred to as a ***zaddr***.

We here concern ourselves only with **Zcash Protocol** which allows us to specify a coherent language and symbols to describe the new ITM ***zaddr*** linkability attack and mitigations against it. All techniques here could technically also be used against transparent blockchains, but since they leak all the useful metadata already, it would serve no purpose. These new attacks can be thought of as "squeezing" new metadata leakage from zaddrs out of places that nobody thought to look.

For those coins which only have a transaction graph at the network p2p level but not stored on their blockchain (such as MimbleWimble coins), it does raise the bar and attack cost. Since nation-states and are not cost-sensitive and obviously have a vested interest to de-anonymize all blockchains, MW coins are not immune to these new attacks being applied. A transaction graph still exists and so the core concepts here can be applied.

2.2 Types Of Shielded Transactions

There are many types of shielded transactions, mirroring the complexity of transparent transactions in Bitcoin Protocol. Here we introduce a convention for describing transactions.

- A fully shielded transaction T with change $T : z \to z, z$
- A fully shielded transaction T with no change $T : z \to z$
- A shielded transaction T with transparent change $T : z \to z, t$
- A dehielding transaction T with change $T : z \to t, z$
- A dehielding transaction T with no change $T : z \to t$
- A shielding transaction T with no change $T : t \to z$
- A shielding transaction T with shielded change $T : t \to z, z$
- A shielding transaction T with transparent change $T : t \to z, t$

The above summarizes the most common transactions. Now say we want to describe a transaction which sends to 5 ***zaddrs*** and 3 transparent addresses with no change: $z \to z, z, z, z, z, t, t, t$. To describe very large transactions subscripts can be used : $z \to z_{52}, t_{39}$.

An individual transaction T is a sub-graph of the full transaction graph $T \subset \mathbb{T}$ with vertex count of one.

3 Metadata Analysis of Zcash Protocol Blockchains: Advanced

3.1 Active vs Passive Attacks/Analysis

In addition to purely analyzing public information available to every full node, there is an **active mode** possible in any analysis. That is, to inject data (funds) and see how the blockchain reacts, to "follow the money" as it were. Some organizations must provide ***zaddrs***to their customers or know the ***zaddrs***of their customers, such as exchanges, mining pools and wallet providers. Also, many individuals choose to publicly post zaddrs and txid's which tie their social media and real life identities to unique blockchain identifiers. Many users accidentally paste this information, not realizing that Github issues and forum posts are mined for this OSINT data, but other defiantly choose to post it, such as zecpages.com . Our opinion is that they mean well, and are helping adoption in some way, but they are making the job of de-anonymization much too easy. Many of these users will post screenshots including their zaddr and transaction id or explorer link. This allows linking a zaddr to a ShieldedInput or ShieldedOutput, which should never normally be possible, and makes the job of the analyst that much easier. It allows software to potentially say "This twitter user owns this zaddr and sent funds in this txid which eventually ended up in a zaddr owned by another twitter user" and other similar inferences.

As an example of active mode against an exchange that supports ***zaddrs***, the attacker can create an account and get a deposit ***zaddr***at the exchange. All forms of dust attacks are now available to the attacker.

Similarly for mining pools which support paying out to ***zaddr***, an attacker can join the pool and mine enough to get a single payout. They will now know one of the zaddrs and the exact amount being paid out in that transaction. Mining pools are a wealth of information to de-anonymize ***zaddrs***and must be very careful to not leak useful metadata.

3.2 Timing Analysis

This analysis uses the heuristic that transactions that are close together are likely to be related, or transactions that form a similar temporal pattern are related. For instance, if you make a transaction at exactly the same time every day, or two transactions, spaced 1 hour apart once per week. In transparent blockchains, the value is always available and timing/value analysis is very powerful. In Zcash Protocol, we only have the timing, and only sometimes the value. Fully shielded $z \to z$ have no value info, while $z \to t$ and $t \to z$ have only partial value information.

3.3 Value Analysis

Value Analysis and Timing Analysis are essentially the same in Bitcoin Protocol but bifurcate into complimentary methods when we add ***zaddrs***to the analysis. In a $t \to z$ transaction, we have "perfect metadata leakage" in the sense that we know the exact amount of funds going into that shielded output. These are somewhat rare but do happen, in the case of spending an output which exactly equals the amount being sent plus fee. There is also the case of $t, t, ..., t \to z$ transaction, which are created by z_shieldcoinbase RPC. This turns transparent coinbase outputs to a single shielded output and leaks the total amount of value transferred to that single shielded output. The more common $t \to z, z$ transaction introduces uncertainty.

Now we consider the de-shielding $z \to t$ which can also be considered to be "perfect metadata leakage" in the sense that we definitely know that an exact amount was in a ***zaddr***which owned that Shielded output and now is in a transparent address. The more common $z \to t, z$ with a change address adds uncertainty and we do not know the exact amount going to the shielded change address.

3.4 Fee Analysis

This analysis is not very clever nor effective but it's simple to analyze the fee of every transaction, no matter whether it is shielded or not, and look for patterns such as non-standard fee use, using lower fees than normal for transaction size and those that pay large fees. Sometimes it is automated software which creates this fee metadata, by

standing out from the crowd of most implementations. Other times it it individual users choosing a custom fee in their wallet, trying to save money. This analysis is essentially free and does not involve ***zaddrs*** at all.

3.5 Dust Attacks

Dust is a term used colloquially and also a very specific term that comes from Bitcoin source code internals. We do not need a strict definition and we use it to mean any very small (potentially zero) amount that does not meaningfully cost much to the attacker.

3.6 Input/Output Arity Analysis

For better or worse, Sapling ***zaddr*** transactions have a publicly visible number of inputs and outputs. This is perhaps the only feature loss from the previous Sprout ***zaddr*** implementation, which used JoinSplits that obscured the exact number of inputs and outputs. The number of inputs you use in your shielded transaction and the number of shielded outputs tells a story.

One simplified example of an "Input Arity Attack", which is active, is as follows: The attacker Alice discovers or finds out the zaddr of Bob and knows it currently has no funds. A brand new created address. She now sends 69 (or some other very unique number) dust outputs in a single transaction, paying the transaction fee. If an when Bob spends those funds, Alice can look for a transaction containing 69 inputs and then identify that txid contains the ***zaddr*** she sent to and link together her original inputs to the outputs of that transaction.

As for output arity analysis, if you have a very unique number of outputs in your transaction on the network, that is bad for your own privacy. If nobody on the network makes transactions with 42 shielded outputs every Tuesday at 1pm, except you, all your transactions can be analyzed as from a single owner, instead of potentially different owners.

3.7 Exchanges and Mining Pools

These entities leak massive amounts of metadata in their normal operations and must expend large amounts of effort to reduce the leakage for their own benefit as well as the blockchains they rely on.

3.8 What does the explorer not show?

A surprisingly large amount!

4 De-anonymization techniques literature review

4.1 Applications to new Shielded-only Chains

5 ITM Attack: z2z Transaction Linkability

The **ITM Attack** specifically "attacks" a transaction $T : z \to z, z$, i.e. a fully-shielded Zcash Protocol transaction which has the highest level of privacy. First we describe the definition of the attack success, if any of the following datums can be ascertained:

- The value in the ***zaddr*** sending funds.
- The value any of the ***zaddrs*** receiving funds.
- The value of any ShieldedInputs spent in the transaction.

- A range of possible values being sent to any ***zaddr***, such as between 0.42 and 1.7 (with error estimate)
- A range of possible values stored in the sending ***zaddr***.

If any of the above metadata can be "leaked", the attack is a success. We note that this attack is completely passive in it's core, but can be greatly improved by adding active components "to taste". This is why metadata leakage attacks such as this can be thought of a method of analysis or an outright attack.

The **ITM Attack** takes transaction id's and ***zaddrs***as input, or other OSINT which is readily available on Github, Twitter, Discord, Slack, public forms, mailing lists, IRC and many other locations. With these public resources, the **ITM Attack** can bridge the gap from theoretically interesting attack to actually de-anonymizing a ***zaddr***to it's corresponding social media accounts, email addresses, IP addresses, location data and more.

This attack is not for weekend warriors or individuals with small budgets and is not cost-effective for attacking a single ***zaddr***. It's best suited for the largest players in The Great Game, i.e NSA, GCHQ and friends. It's highly likely they already utilize analysis and attacks described in this paper.

Only the most well-funded private blockchain analysis companies will be able to afford the infrastructure for this attack, but once the data is "mined" it is a commodity that can be bought and sold to those with less resources.

The ITM Attack is an additional "layer" of analysis that can be overlaid on top of all other types of analysis, and in that way it has the potential to "finish" a lot of "partial de-anonymizations", i.e. places where blockchain analysis provides some data, but not enough to fully de-anon. When added to timing analysis, amount analysis and fee analysis, it can identify that certain ***zaddrs***being involved in many transactions and their approximate input and output values. This data is not available any other way and exact values are not very important.

If a blockchain analyst can ascertain a transaction involves at least 1M USD in value versus a few pennies of value, that directly the course of analysis and investigation. Perfect de-anonymization is not needed and in practice does not matter. Software enabled with data from ITM analysis will be able to identify transaction outputs as having certain ranges of values and potentially their associated zaddrs from OSINT data.

5.1 ITM Attack: Assumptions

Fully working example code is left as an exercise to the interested blockchain analysis company. We shall describe the attack in enough detail for experts to verify our claims and for developers to implement attacks and or defenses, in the spirit of radical transparency.

We assume an attacker has at least 100,000 USD in funds to dedicate to the operation of studying one particular Zcash blockchain. Most of this cost is in the purchase of a GPU/FPGA farm to crunch data. Blockchains with more history and larger shielded pools will be more costly to study.

We note that this attack is not financially feasible as a one-off, it's a methodology to study an entire blockchain which can then be indexed and search for potentially valuabledata. Blockchain anlaysis companies and the IC are strategically positioned to use this information with the least cost, since they already have massive infrastructure to support this new dataset.

5.2 ITM Attack: Defeating *ZK-SNARKs*

We can think of this attack as a "defeat" of zero-knowledge mathematics only in practice, not in theory. Many qualifications are needed. We in no way "broke" the mathematics of ***ZK-SNARKs***, we are taking advantage of how ***ZK-SNARKs***are being used in higher level protocols, i.e. the Zcash Transaction Format Protocol and it's associated consensus rules.

So ***ZK-SNARKs***are sound and we have not actually leaked **knowledge** directly from a **zero-knowledge proof**, that is mathematically impossible. We have leaked knowledge from how these proofs are used in the larger system called Zcash Protocol, itself an extension of Bitcoin Protocol which notoriously leaks metadata.

5.3 ITM Attack: Infrastructure

This attack requires storing a lot of intermediate data in addition to the raw blockchain data and data storage costs are likely the number two expense after computing power. It is possible renting compute power can lower computing expenses but will not lower data storage costs. If one is analyzing a blockchain of B bytes then a reasonable estimate is that $100 * B$ bytes of intermediate storage will be needed to analyze the data and then a highly compressed version of the final useful data can likely be stored in $B \div 100$ bytes or less. That is, the final datasize will be much smaller than the input data but our intermediate will likely be two orders of magnitude larger.

Assume we have a simulated blockchain at block N, held in stasis and the analyst has their own mining hashrate to "push" the chain forward by it's own defined consensus rules. This can be accomplished by blocking all outside nodes and only connecting to the local hashrate.

We also assume the analyst can easily "spin up" a blockchain at a certain block height and try a new change to extract new data. This is trivially possible with virtual machine images, docker containers and/or Git, and is left as an exercise to the motivated blockchain analyst.

5.4 ITM Attack: Consensual Oracles

We now analyze a specific $T : z \rightarrow z, z$ at a speficic block height H which defines a specific **shielded pool** containing unspent shielded outputs and their associated metadata, such as **Merkle Tree** data.

Very specifically, the simulation will use the **SaplingMerkleTree** internal Zcash Protocol datastructure defined in src/zcash/IncrementalMerkleTree.hpp . The ITM Attack focuses on this data structure but others can and should be explored as metadata oracles, such as the **SaplingWitness** data.

At any given block height H a shielded "note" or **zUTXO** is either spent or unspent. Just like transparent **UTXOs**, a **zUTXO** can be spent from the mempool, i.e. the output of a transaction in this block can be spent by another transaction.

Different implementations of Zcash Protocol may react differently to spending zfunds from the mempool and so that is definitely a potential area of research.

Known Sapling commitments/anchors are "swapped" into the SaplingMerkleTree one at a time, in an attempt to identify if they are being spent. If the new solution tree is invalid, then the data that was added caused it to become an invalid tree for a particular reason and that particular reason is conveniently given when consensus-level errors are emitted in Bitcoin and Zcash Protocols. These errors have their own error codes and provide a wealth of information leakage to the aspiring analyst. By trying various known bits of data and analyzing the exact consensus error codes emitted, information is leaked.

6 Metaverse Metadata Attacks

The ITM Attack is a special case of what we name **Metaverse Metadata Attacks**, applied to Zcash Protocol shielded transaction graphs.

The term **Metaverse** is appropriate because alternate possible blockchain histories can be simulated to see what consensus rules would have produced. By meticulously changing one piece of data at a time, the analyst can use the consensus rules at that moment in blockchain history as an **oracle**. In this sense, **Metaverse** attacks can be classified as **consensus oracle attacks**, similar to **compression oracle** attacks and **padding oracle** attacks such as BREACH and CRIME against TLS.

As far as the authors know this is a new technique that has not been publicly described. Blockchain consensus rules can be simulated in a vacuum and the scientific method of changing one variable at a time can be used to extract metadata from privacy coin public data.

7 Sietch: Theory

The ITM Attack relies on the fact that the most common shielded transaction on most currently existing Zcash Protocol blockchains have only 2 outputs $T : z \to z, z$ and the basic fact that if some metadata can be leaked about one output, if it's **spent** or **unspent** or it's range of possible values, it provides a lot of metadata on the other output as well.

If there were 3 outputs, then there would be uncertainty involved, instead of a more direct algebraic relation such as "if one output had amount=5 then the other output had an amount of $total - 5$". When 3 ***zaddr***outputs are involved, knowing the value of one ***zaddr***output does not provide as much information on the value of any other particular ***zaddr***.

This principle obviously increases, as the number of outputs increases, the leakage of the amount of any one ***zaddr***input becomes exceedingly less valuable and expensive metadata to utilize.

By design, Sietch is opt-out and by default all users use it without knowing it, which has worked well. Sietch makes every individual shielded transaction more complex which creates a harder-to-analyze transaction graph, helping even users which have custom software that does not use Sietch.

The effect of almost all Hush users using Sietch all the time without knowing it, is a "herd immunity" against de-anonymization. The price is waiting a few extra seconds for each transaction and the Hush community feels it is quite well worth it.

Even if some outputs of a transaction are completely de-anonymized, there are so many other outputs that exact values being transferred cannot be ascertained. This mimics the case where an infected person cannot easily infect another person with a virus because the people near them are already in recovery or immune.

8 Sietch: Code In Production

Sietch uses a default rule of a minimum of 7 ***zaddr***outputs in a transaction. Because the average shielded transaction does not spend the input values exactly and there is a change output, in practice the average Hush transaction has 8 ***zaddr***outputs.

This is currently not a consensus rule and only enforced at RPC layer. There are currently various implementations of Sietch in our full node and lite wallets.

Whenever a transaction is made with less than 7 ***zaddr***outputs, the RPC layer automatically adds them, which means all software which uses the RPC layer is protected with absolutely no code changes. Software which uses raw transactions must take care of this themselves.

This has the practical effect of hiding the number of recipients to the average transactions on the Hush network. When you see a $z \to z, z, z$ transaction on ZEC mainnet, you can be almost sure it is one ***zaddr***sending to 2 other ***zaddrs***and a change output. It could also be sending to three outputs with no change, with drastically less probability. This type of transaction is "upgraded" to $z \to z_7$ at a minimum and so you don't know how many recipients are being sent to, except if it is a large number. In practice, this obscures most transactions on the network and it is mostly mining pool payouts which routinely use many ***zaddr***outputs or other automated software.

Some transactions look like $t \to t, t, z, t$ which is a transparent address sending to two other transparent addresses, one shielded address and a change output. When Sietch is enabled, this transaction is "upgraded" to $t \to t, t, z, t, z_6$ to satisfy the minumum of 7 ***zaddr***output rule. Originally the exact amount of value being transferred to the ***zaddr***would be known, because all other values in the transaction are transparent and appear on the public blockchain. But in the "upgraded" transaction we can only ascertain that some amount A was sent and spread out across 7 outputs, some of which may be of zero value.

In general, Sietch transactions make the job of de-anonymizing a chain much harder at the individual transaction level, which then builds up into a very strong and complex shielded transaction graph. The average ZEC mainnet shielded transaction has two outputs and so it's shielded transaction graph looks like a binary tree, while the Hush

blockchain with Sietch looks like a tree that splits into 8 parts at each node. Trying to follow the flow of funds becomes combinatorially impractical and expensive for even the largest players.

9 Thoughts On Device Seizure

TLDR: You should really care about this.

For example, say Alice sent Bob and Charlie funds in a fully shielded transaction with shielded change: $z \rightarrow z, z, z$

Now let us say that Alice and Charlie have their devices seized, wallet.dat's "liberated" and uploaded into chain analysis software that understands Zcash Protocol and ITM-Style Attacks. Bob is now in a posistion where his ***zaddr***is known by the analyst/attacker, the exact amount sent to him in a certain txid and potentially other meta-data in a memo field. All of this data is valuable input which makes the ITM attack better at it's job, and can often help "complete" partial de-anonymization which was unable to fully "resolve" the data.

Even without any new attacks, device seizure and uploading wallet.dat contents into blockchain analysis software poses an enormous threat to privacy coins and so they should design systems that assume this will happen and to isolate and comparmentalize the damage possible. Sietch provides one such way to provide a safety and privacy buffer against real-life scenarios.

10 Advice To Zcash Protocol Coins

Low numbers of ***zaddr***outputs are bad for privacy, especially 1 or 2. Enforcing at least 4 likely makes the ITM attack likely impractical. Hush chose 7 as a security buffer and because the slowdown associated with 7 outputs amounts to about 5 seconds on modern hardware, when spending a small number of inputs. This seemed like a reasonable amount of time for users to make a transaction, given that the original Sprout ***zaddrs***took over a minute to make the simplest of transactions.

Shielded coinbase seems interesting but leaks a grave amount of metadata about the zaddress of the miner, which can feed into this analysis. We recommend Pirate, Arrow and other coins implementing enforced ***zaddr***usage avoid implementing the new ZIPXXX.

Allowing users to spend huge numbers of inputs at once makes their transactions stand out. GUI wallets and education need to improve to reduce loss of privacy.

Do not advocate that users post ***zaddrs***and the txid's and explorer links they are involved in! Educate them to keep this metadata to private messages, DMs and other non-public places. The fewer people that know your ***zaddr***, the better!

11 Special Thanks

Special thanks to jl777, ITM, denioD and Biz for their feedback.

12 References

Speak And Transact Freely