EDUCATION

Blockchain Privacy: Equal Parts Theory and Practice

BY IAN MIERS FEBRUARY 12, 2019

Ian Miers is a postdoc at Cornell Tech and a member of the Zcash Foundation Board of Directors. He helped create Zcash and co-authored the Zerocoin and Zerocash papers that were its precursors. In 2018, at Scaling Bitcoin and Devcon4, Miers gave talks about the privacy-violating attacks that are possible against decoy-based systems. The essay below is based on those presentations, developed by Miers and the Zcash Foundation. It was originally published by Token Daily.

Table of Contents

Satoshi Has No Clothes

The cryptocurrency community has done a poor job of evaluating privacy. We are even worse at explaining the tradeoffs of different implementations to regular users. Improvement is necessary and it needs to happen now. Many of these protocols aspire to be the future of payments — one of them may win. By the time that happens, it'll be too late to get the design right.

In 2011, when I started working on privacy in cryptocurrencies, it was commonly thought that Bitcoin was private. WikiLeaks solicited "anonymous Bitcoin donations" on Twitter, which is somewhat tragic; we can confidently guess that a few WikiLeaks donors were in sensitive positions, at least.

Now we're aware that Bitcoin is nowhere close to anonymous. A number of academic papers have shown that you can link pseudonymous transactions together and thereby track what someone is doing across a blockchain. In addition, companies like Chainalysis are in the business of discovering and surfacing such analytics.

Bitcoin is Twitter for your bank account. Anyone can see what you're doing. That includes your family members, friends, current and former romantic partners, business associates, competitors, all the way up to government agencies. Even people who are government decision-makers themselves should remember that other governments — the ones they don't like — will delve into the details of their finances.

It's common to say that "privacy is dead," suggesting that it's hopeless to protect your privacy. The idea is that someone — the government, Google, a mysterious bogeyman — will always know things about you. But there's a difference between one person knowing your deepest, darkest secrets, and everyone knowing them. Just because Google knows your browsing history doesn't mean that you want it to be public.

During the past seven or eight years, we've seen many proposals to add privacy to cryptocurrencies. The techniques range from simple things, like avoiding address reuse, to complex cryptographic protocols. Measuring the privacy afforded by a certain implementation is tricky.

Right now, we can't resort to empirical methods. It would be akin to evaluating internet privacy in 1992, when the only websites were ones at CERN. That was before targeted ads, and tracking cookies; Google AdWords didn't launch until 2000. Richard Stallman was considered an alarmist crank. It was before we really used the web for anything where it would be worth tracking people.

In the current cryptocurrency ecosystem, you cannot look at people's usage and then produce an authoritative estimate of whether (or which!) privacy techniques are effective. The necessary data isn't there. Today nearly all transactions are speculative, which illustrates the privacy needs of risk-loving investors, but leaves aside everyone else.

We don't have the rich tapestry of structure that results when you pay for your train trip, walk to the local market to buy a sandwich, then mail a package at the post office, then buy something at the vending machine. That kind of behavior, and the data generated by it, is not evident among the vast majority of cryptocurrency users.

As a researcher, even if this data existed, I couldn't use it. I have limited access to data due to cost concerns, and I and other academic researchers have ethical limitations imposed by the Institutional Review Board. Our adversaries do not.

The upshot is that an empirical evaluation of future privacy is impossible. Instead of relying on data, we must resort to thought experiments. We need to think through the usage of our systems in the coming decades and consider how that will play out. One viable approach is to look at the problems in related domains.

Real-World Privacy Threats

The most common threat that people bring up is governments and law enforcement leveraging blockchain data. As with the privacy needs of speculators, that is one threat, but it's not the only one. Nor is it the threat most likely to affect the public at large. (That said, we should not dismiss the concerns of activists and dissidents.)

Looking beyond cryptocurrencies, we recently learned that Google has been collecting offline payment data from Visa and MasterCard and using it to build up profiles for targeted advertising. You may think that Google does a good job and institutes reasonable security controls, or you may not. Regardless, it's a worrying trend (and not a new one). If Google is doing it, so are people and entities that are less scrupulous. You've never heard of them and you have no idea how they're using information about your transactions.

Similarly, we know that companies want to build up rich profiles of their customers' behavior. There are numerous sources of data for them to compile — for example, usage of loyalty cards and coupons. Retailers can track and analyze this information, to the extent that they're able to guess when customers are pregnant, since pregnant customers exhibit certain purchasing patterns. Other medical conditions likely fall in the same boat.

News reports have indicated that retailers aim to discover these things before you even know yourself… or at least before the other people in your family know. In 2012, Charles Duhigg wrote a feature for *The New York Times Magazine* that contained this anecdote:

> *About a year after [Target data scientist Andrew Pole] created his pregnancy-prediction model, a man walked into a Target outside Minneapolis and demanded to see the manager. He was clutching coupons that had been sent to his daughter, and he was angry, according to an employee who participated in the conversation. "My daughter got this in the mail!" he said. "She's still in high school, and you're sending her coupons for baby clothes and cribs? Are you trying to encourage her to get pregnant?" The manager didn't have any idea what the man was talking about. He looked at the mailer. Sure enough, it was addressed to the man's daughter and contained advertisements for maternity clothing, nursery furniture and pictures of smiling infants. The manager apologized and then called a few days later to apologize again.*

> *On the phone, though, the father was somewhat abashed. "I had a talk with my daughter," he said. "It turns out there's been some activities in my house I haven't been completely aware of. She's due in August. I owe you an apology." There are serious privacy problems with data about what people buy. It's plausible that sexual orientation could be targeted in the same way. These examples are more*

> *fine-grained than you might be able to extract from a blockchain, but nonetheless the issue manifests in a system like Bitcoin.*

A more on-the-nose example is Venmo. For those of you who don't know, Venmo is a service primarily used for payments between friends, to pay for a bar tab or split a restaurant check. By default, Venmo has a public feed of every transaction that its users make. It includes your name, the recipient's name, and a memo field describing why you paid them. That is pretty close to the data on the Bitcoin blockchain.

We've seen the failure cases of Venmo's public feed, including small-time pot dealers being arrested and supposedly lighthearted guides to stalking your ex-boyfriend. That's playful in theory, but actually no, it's creepy and abusive. People should not be okay with any system having these features.

Another threat that's more well-known to the cryptocurrency community, where issues are even cropping up today, is fungibility. We know that for certain cryptocurrencies, freshly mined coins sell for a premium. Exchanges sometimes block customers based on their transaction history; where they've sent their money in the past.

It's important to note that exchanges are powerful. We can't think of them as merely third-party observers. They know more about you than just the transaction graph. Frequently they conduct transactions on behalf of their users. The privacy problem here is akin to trying to maintain privacy from Google while using Gmail and Google Maps on an Android phone. At some level, you're embodying your adversary.

Remember, Bitcoin is Twitter for your bank account. And not the kind of Twitter where you choose what tweets to write and publish. Bitcoin is more like a creepy alternate-universe Twitter that automatically transmits all of your thoughts.

Defenses and Failures

What are the viable defenses?

In a world of massive data collection and machine learning, plausible deniability doesn't work. Typically when I talk about this, someone comes up to me and says, "What if I tell the police, 'Hey, you can't prove it's me!'" That is naivete, insufficient for the real world. The algorithms being deployed don't care about plausible deniability; they operate on probabilities. And when the probability is high enough, that holds up for law enforcement purposes as much as advertising.

Blockchain privacy is not intuitive. Typically people tend to think of passive third-party observers as the main threat. But it's crucial to consider active attackers who can send payments to you, receive payments from you, and interact with third parties. Obvious examples of such attacks are merchants or cartels of merchants who keep track of customers, people who try to identify a payment recipient's real identity, and exchanges that also want to track you. (I'll address these scenarios momentarily.)

The range of supposed solutions to privacy problems is huge, so I won't review all of them individually. However, we can look at the approaches broadly in terms of three different kinds of systems.

First, some systems look like vanilla Bitcoin, where you explicitly identify the origin of your payment. The only protection here is that there are no real names.The base layer doesn't even attempt to obfuscate transaction data, which is now widely understood in cryptocurrency circles. (The general public could still use education on the issue.) Another approach is what I'm going to call decoy-based systems, where you hide what's going on in a given transaction by selecting a certain number of possible payment origins. The strongest approaches are Zerocoin and Zerocash, where no origin at all is identified.

In decoy-based systems — CoinJoin, Monero's RingCT, and others — you are required to explicitly verify the source of your funds, but you try to hide it by including a handful of decoys that aren't your real source. Theoretically, anyone looking at the transaction cannot tell which is which. The actual origin is obfuscated by adding noise.

And again, in systems using the principles of Zerocash, you don't have any identifiers whatsoever.

My position is that we haven't properly examined the downsides of decoy-based systems. It is a significant oversight, because much of the cryptocurrency community is turning to decoys as a source of scalable privacy. Decoy-based systems do not provide the robust, attack-resistant privacy that people assume.

Decoy-Based Deanonymization

Let's say that you're sending a decoy-obfuscated transaction. The protocol identifies the possible source of the funds, along with a handful of decoys. Now an observer or attacker has access to a tree of possibly associated payments that go back in history. They can't pinpoint quite what happened, since it's like a fuzzy family tree, but they can extrapolate some notion of what's going on based on this single transaction. That family tree — what I will call a taint tree — also works going forward.

Overseer Attack

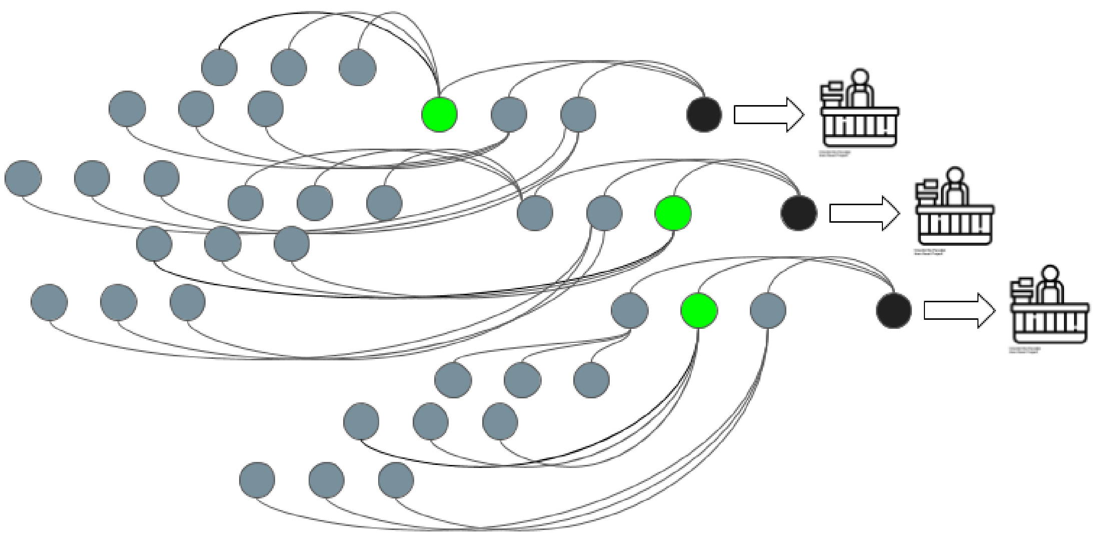
Let's say that your transaction was a payment to a merchant. Where does the money go next? Attackers cannot know precisely because of the systemic use of decoys. But they will be able to trace a finite number of possibilities in terms of where the money might have gone. Next, they can start a process of elimination.

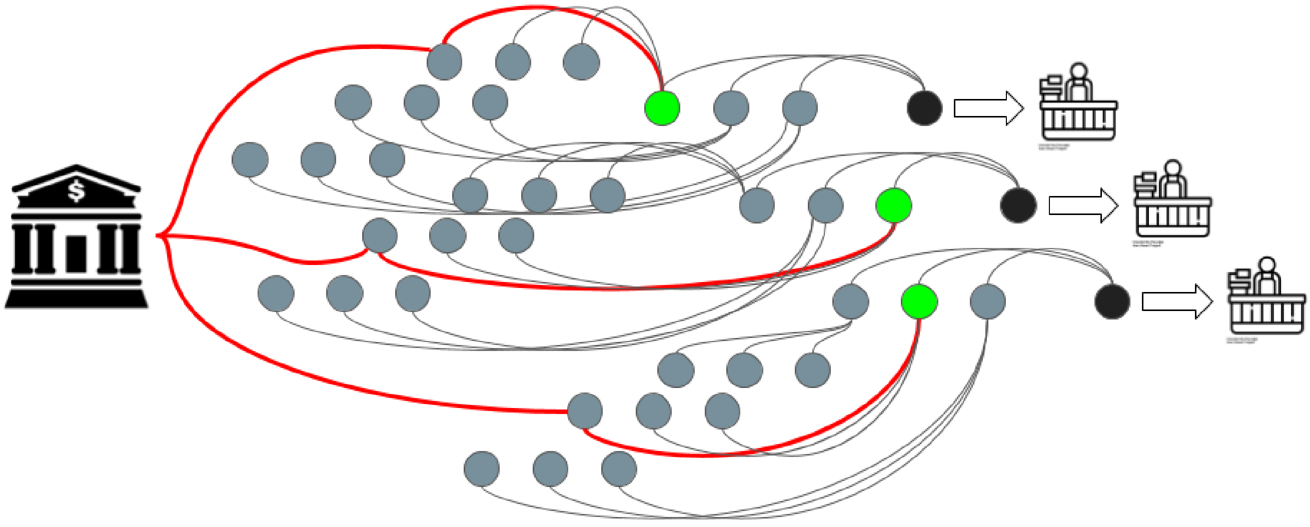
The taint tree gives an attacker a lot of power, especially when tracking analysis is repeated over multiple transactions. One thing you can do if you're a merchant, or a set of colluding merchants, is track customers across repeated payments.

Hypothetically, I'm going into Target on a daily basis and making cash purchases. There should be no way of tracing me — beyond arduous methods like dusting for fingerprints or DNA, which requires already having those biometrics, or knowing in advance the serial numbers of the bills I'm going to use.

What if I start using a cryptocurrency to buy things at Target? (No, large retailers don't accept cryptocurrencies yet, but that's the endgame of these technologies.) Ideally I could make three separate purchases and there would be no way to link them together. A cryptocurrency with true privacy would achieve that.

If you look at decoy-based systems superficially, it seems like that do achieve that. None of these transactions appear to be linked together until you draw the red lines:


Overseer attack: tracking repeat customers


Overseer attack: tracking repeat customers

It gets worse. Again, let’s consider multiple payments that I’ve made to one merchant. I don’t want them to know that I’m the same person, but in a decoy-based system you have taint trees of possible ancestors. Well, what happens if they have a common origin? I went to Coinbase or whatever exchange, and I bought a bunch of cryptocurrency, then I loaded it onto the blockchain.

There’s going to be one source of those funds. If you trace back the taint trees, you can look at the intersections and pinpoint the person making these transaction. That method works not just for one merchant, but also for groups of merchants — or other entities that receive payments. They can collude to figure out who you are, which is a problem when the goal is privacy.

Flashlight Attack

Let’s suppose I want to accept payments online, anonymously. For example, I’m a dissident in an authoritarian country who needs to accept donations, but I cannot reveal my real identity; my life is at risk in the country where I do my activism. But I need to be able to fund my work. Of course, the government of that locale is trying to identify me. They have intel agencies and secret police at their disposal.

If I’m using a privacy-preserving cryptocurrency, it should be safe for me to deposit the donated funds at a local exchange. Even if that exchange is controlled by the government! Ideally the data that could be used to identify me — probabilistically or otherwise — is simply not available. I should be safe regardless of whether the exchange is hacked, corrupt, subpoenaed or otherwise infiltrated. What I’m describing is how it *should* work, not how it actually works.

If the government wants to identify me, they have my cryptocurrency addresses because I’ve exposed them in order to accept donations. Maybe my website is only accessible over Tor; maybe I even use a unique address per donation. And of course I’m relying on a decoy-based cryptocurrency.

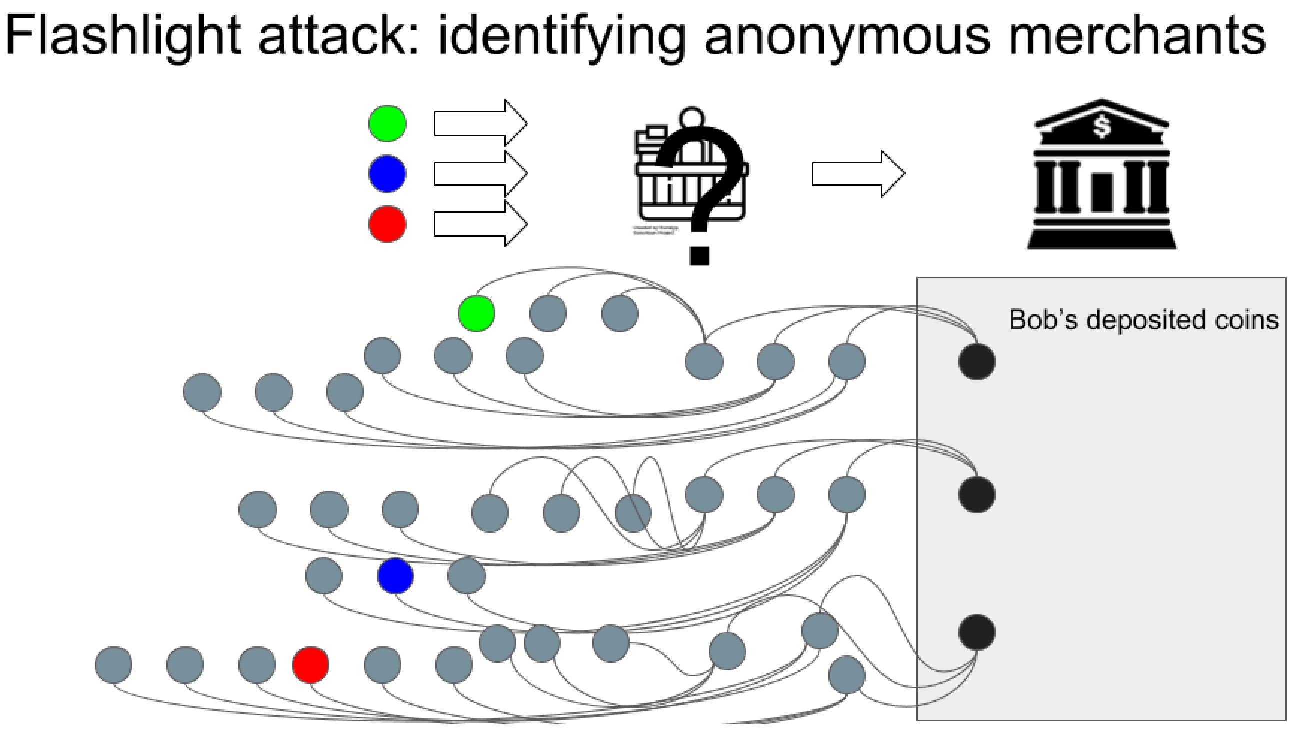
The government realizes that they can send tracking payments to an address of mine. Three of them, perhaps, or 20, or 100. The payments can be very small; size is irrelevant. At some point I’m going to deposit the funds from those payments.

Now I've got a big problem. Anybody who can access the exchange's records is now able to test whether the depositor is the same person as the democracy activist. They can examine the set of coins that I've deposited and reconstruct the taint tree, the possible sets of origins.

For any random person, it would be unremarkable that their deposits involved tainted payments. Decoys are picked at random, so by happenstance, one of the tainted payments could make its way into their deposits. On the other hand, the probability of that happening multiple times is quite low. It's vanishingly unlikely to happen with all of the funds from 100 tainted payments that were sent to this one democracy activist.

The government can look through all of my deposits, and see that my taint tree contains all of the tracking payments that they sent. That evidence links my legal identity to my democracy activism with overwhelming probability.

As you can see, taint trees are viable for deanonymization, and thus decoy-based systems violate people's notions of how privacy should (or does) work in cryptocurrencies. Taint trees allow privacy to be ripped apart in a way that would shock and concern many users.



This is probably the easiest to execute and most immediately troubling attack on decoy based systems.

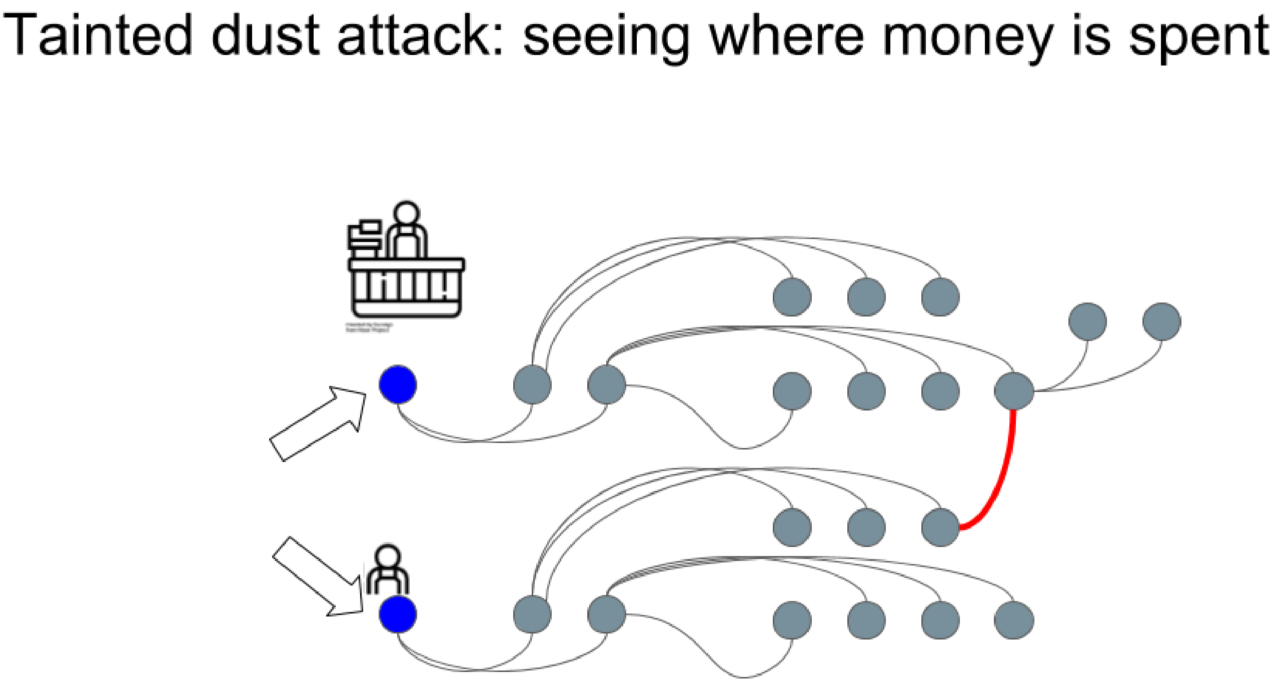
The takeaway is that repeated interactions with a malicious sender or recipient are dangerous. But it keeps getting worse!

Tainted Dusk Attack

Remember when I mentioned that taint trees can be used to trace money going forward? After you make a payment, there's an uncertain cloud of possible transactions that could involve those funds. That can also be abused. For example, an attacker could find out where a friend — or family member, or ex-lover, or anyone else they know a bit about — is spending their money.

Let's say the attacker makes a small payment. It could even be a dust transaction. They make a payment to some merchant, and then to their victim. They keep watching as the taint tree grows out, as possible spends happen.

At some point, there's an interesting crossover. The attacker notices a transaction that seems to involve both the funds they sent to the merchant and the funds they sent to their victim.



Tainted dust attack: seeing where money is spent

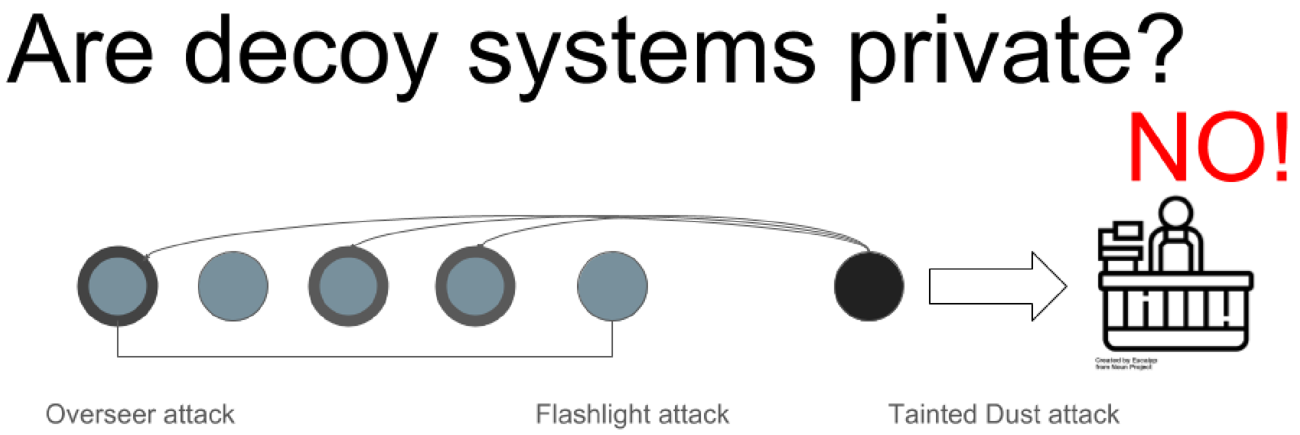
Many plausible explanations exist. The crossover could result from random decoys. Or perhaps the victim who received the attacker's transaction was spending money with the merchant in question. What the attacker now sees is the merchant moving funds out of a hot wallet, or spending them to pay bills, or whatever else. Again, any one instance is not definitive. But if the pattern repeats several times, then you have strong probabilistic evidence that your friend is making recurring payments to this merchant.

Law enforcement could use an analysis along these lines to validate that a particular person does indeed use a particular supplier. Or you could identify that your friend makes purchases on Pornhub. Which would be incredibly embarrassing for them, not because they're paying for porn, but because they're probably doing it using Verge.

In summary, the limitations of decoy-based privacy systems are readily apparent once you threat model how attackers might approach them. You must consider what people can actively do, what they can't, and what goals they're likely to have. Various privacy proposals need this kind of rigorous evaluation or they can't be expected to stand up against clever adversaries (especially well-resourced ones). Cryptocurrency designers have to ask themselves, "If I were going to identify someone via this system, how would I go about it?"

The democracy activist taking donations over Tor might think, "I'm safe, I'm behind seven proxies!" But with a decoy-based system that isn't true. The moment someone can start sending you tracking payments, and then get data from an exchange, you lose any and all privacy.

Solving Decoy Problems



Are decoy systems private?

NO!

The common perception of these various techniques seems to be, "Well, Bitcoin may not be private, but anything above-and-beyond Bitcoin will add meaningful privacy." The reality is that specific techniques and implementations matter. The details are crucial. Users need to understand the tradeoffs afforded by the specific system they're using. Buying modafinil has a different threat model from protesting an authoritarian regime.

I'm not saying that it's impossible for decoy-based systems to provide meaningful privacy. If your decoy set is very large — think five million possible origins to identify, rather than five — that changes the probabilistic evidence that attackers can uncover. On top of that, the decoy sets would have to substantially overlap across all recent transactions. Otherwise you will still see repeated common origins when making multiple purchases with a merchant and so on.

Finally, it's important to sample the decoys carefully. I won't go into it here, but a couple of papers have shown that the distribution from which Monero sampled their decoys didn't line up with the distribution of people's transactions. There was a gap. In previous versions of Monero — this is now somewhat fixed — the last transaction in the decoy set was actually the real transaction, with overwhelming probability, because of recency preferences.

Most decoy-based systems are intended to be practical. In order to get substantial decoy sets, you can't have systems that scale linearly in the number of decoys. Using Monero and bulletproofs as an example, each additional decoy costs you 1-2 kilobytes in transaction size. It should be very clear, with linear scaling, that you're not going to have a transaction with 100 decoys in it, or 500, or a thousand. Proof generation and verification scale equivalently, which ruins the practicality.

What you need is logarithmic size. The transaction size should be logarithmic in your decoy set, and transaction generation and verification time should be at least logarithmic if not constant.

Zero-Knowledge Approach

I’m a little biased, but in my view the solution is a Zerocash-style protocol. Transaction outputs are commitments to the value in the recipient address, and you generate a Merkle tree over some fraction of the UTXO set, whatever you can afford computationally. A zero-knowledge proof is used to show that the origin of your payment exists in the UTXO Merkle tree. It can be verified without revealing the UTXO in question. This is where the privacy comes from. That’s the basic approach of Zerocash, where the entire UTXO set is included in the Merkle tree.

How do you make that scalable? You have to pick a zk-proof technology that you like, and by “like” I mean: You think the cryptography is secure, you think that the assumptions are warranted, and the setup properties work for whatever operational requirements you have. It might be SNARKs, or STARKs, or bulletproofs. After choosing a zk-proof, you can tinker with scalability.

The scheme and parameters have been selected, so now you turn to efficiency. Start benchmarking. As the Merkle tree gets longer the transactions are going to get bigger and the verification times are going to slow down if your not using QAP-based zk-SNARKs like in Zcash.. The goal is finding a depth where efficiency meets your performance requirements. Maybe it’s d=32, which Zcash Sapling uses. Maybe d=4, maybe 8, it doesn’t really matter. Whatever you do, your decoy set is now 2^d, which exceeds most decoy-based approaches.

I should briefly note that the state of the art for these techniques is improving. With respect to zk-SNARKs, it’s gone from taking ~40 seconds to like two seconds to generate a transaction. Huge amounts of memory used to be required, in excess of three gigabytes, and now it’s 40 megabytes. Similarly, bulletproofs keep getting faster and faster.

Conclusion

We need to deeply consider our approaches to privacy. Cryptocurrencies should be built with robust, attack-resistance solutions for protecting financial information. That may happen on-chain, but it’s not a given. The current mantra is that privacy will be ensured off-chain. That’s fine and I hope it works, but it doesn’t absolve you from assessing the default weaknesses of your system. Merely because it's off-chain doesn't mean that information doesn't leak.

It's been interesting to observe the reactions to my talks at Scaling Bitcoin and Devcon. Some projects care giving their users accurate expectations. For example, the Grin project has written up the state of its privacy protections. That's exactly what cryptocurrency developers should do, and the document is excellent. Grin's team took a very conservative position, talking about the privacy that is available now — not hypotheticals or privacy theater. My only concern is that Grin underplays the risks with leaking the transaction graph (what they call "inputs and outputs linking"). But all in all, the "Grin Privacy Primer" is very good, and I wish more groups would strive for equivalent clarity.

Unfortunately, many others have responded with the exact kind of privacy theater that I featured in my talk. It is irresponsible to claim that CT, stealth addresses, or Dandelion provide comprehensive or perfect privacy. None of those technologies address the issues that I've raised. None of them stop the flashlight attack that would allow governments to identify someone's legal identity by interacting with a dark-web site that accepts payments. It is a major concern for some users today, but privacy theater distracts from the real risks.

Finally, a number of people have noted that some of the attacks I mentioned may be hard to mount in practice, because of noise and large volumes of transactions. For the tainted dust attack, that's absolutely true. But it's not true at all for the flashlight attack or the overseer attack.

In general, the attacks that I described are thought experiments. The goal is to make you realize that many systems aren't as private as people think they are, and to guide explorations of the practical levels of privacy. It may be the case that with enough traffic and sufficiently large decoy set, you get viable privacy. However, barring an analysis proving that, we have to think about which implementations pass a basic smell test. Moreover, my examples are the basic attacks that come up when thinking through real-world cryptocurrency usage. Adversaries are clever, creative, and diligent.

Remember, passive third parties are not the only attackers. That's not the main threat that people face with existing technologies on the internet today. It's being tracked by companies, or malicious ex-lovers, or oppressive governments. Also remember that attacks only get better. We are in the early days of cryptocurrency functioning and usage. Compared to the internet or other older systems, we have very little experience with building or protecting cryptocurrencies.

By all means, choose to prioritize scaling over privacy. That is a reasonable choice to make as developers and as a community. But when you do that, understand what you're giving up in terms of privacy, and be transparent about it. Don't pick any random approach and say, "It adds some privacy, ergo the thing is completely private." That's not true; adding some privacy doesn't make a protocol private in totality, and users will still be vulnerable.

It took, what, 20 years for us to understand how bad the privacy problems with the internet were? Progress has accelerated, but a couple of years will not be enough. Five years, or 10, maybe. It's important to lay the groundwork for privacy now.

Previous Next

commercial partnerships or co-promotion.

Suggest an edit or modification on GitHub.