sciendo

Proceedings on Privacy Enhancing Technologies ; 2018 (4):179–199

Steven Goldfeder*, Harry Kalodner, Dillon Reisman, and Arvind Narayanan

When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies

Abstract: We show how third-party web trackers can deanonymize users of cryptocurrencies. We present two distinct but complementary attacks. On most shopping websites, third party trackers receive information about user purchases for purposes of advertising and analytics. We show that, if the user pays using a cryptocurrency, trackers typically possess enough information about the purchase to uniquely identify the transaction on the blockchain, link it to the user's cookie, and further to the user's real identity. Our second attack shows that if the tracker is able to link two purchases of the same user to the blockchain in this manner, it can identify the user's cluster of addresses and transactions on the blockchain, even if the user employs blockchain anonymity techniques such as CoinJoin. The attacks are passive and hence can be retroactively applied to past purchases. We discuss several mitigations, but none are perfect.

DOI 10.1515/popets-2018-0038
Received 2018-02-28; revised 2018-06-15; accepted 2018-06-16.

1 Introduction

Eight years after Bitcoin's introduction, the ability to pay online using cryptocurrencies is common: prominent merchants such as Microsoft, Newegg, and Overstock support it. Cryptocurrency users tend to value financial privacy, and it is a major reason for choosing to pay with digital currencies [33]. Yet, websites including shopping sites are known to be rife with third-party tracking [16]. In this paper, we study the impact of online tracking on the privacy of cryptocurrency users.

First, we show that online trackers are able to see sensitive details of payment flows, such as the identities and prices of items added to shopping carts. Crucially, in many cases they receive sufficient information about a purchase to link it uniquely to a transaction on the Bitcoin blockchain.[1] This core linkage can be expanded in both directions: based on tracking cookies, the transaction can be linked to the user's activities across the web. And based on well-known address clustering techniques [3, 42], it can be linked to their other transactions.

This basic attack can be made worse in several ways. We find that many merchant sites send even more information to trackers, such as the transaction-specific Bitcoin address. This acts as a high-entropy identifier and makes linking to the blockchain trivial. We also show that many merchants additionally leak users' PII (name, email address, etc.) to trackers, allowing trackers to link not only users' web profiles but also blockchain transactions to their identities. Finally, malicious trackers may use JavaScript to extract Bitcoin addresses or PII from web pages even if it is not leaked to them by default. We show that this is possible on the vast majority of merchant sites.

Of course, Bitcoin does not guarantee unlinkability of transactions. But while linking of a user's Bitcoin addresses *with each other* is well known [3, 42, 55, 57], our attack shows how to link addresses to external information, including identity.

The main defense against linkage attacks is mixing [12, 58]. The best known mixing technique is CoinJoin, in which users send coins to each other in a way that hides the link between their old and new coins. Our second main contribution is showing the effectiveness of the *cluster intersection attack*, a previously known attack against mixing. Specifically, we show that a small amount of additional information, namely that two (or more) transactions were made by the same entity, is sufficient to undo the effect of mixing (see Figure 1). While such auxiliary information is available to many potential entities — merchants, other counterparties such as websites that accept donations, intermediaries such as payment processors, and potentially network eavesdrop-

***Corresponding Author: Steven Goldfeder:** Princeton University, E-mail: stevenag@cs.princeton.edu
Harry Kalodner: Princeton University, E-mail: kalodner@cs.princeton.edu
Dillon Reisman: E-mail: dillon@lonlon.io
Arvind Narayanan: Princeton University, E-mail: arvindn@cs.princeton.edu

1 Throughout we study Bitcoin since it has the most support for online payments, but our findings apply to many other cryptocurrencies.

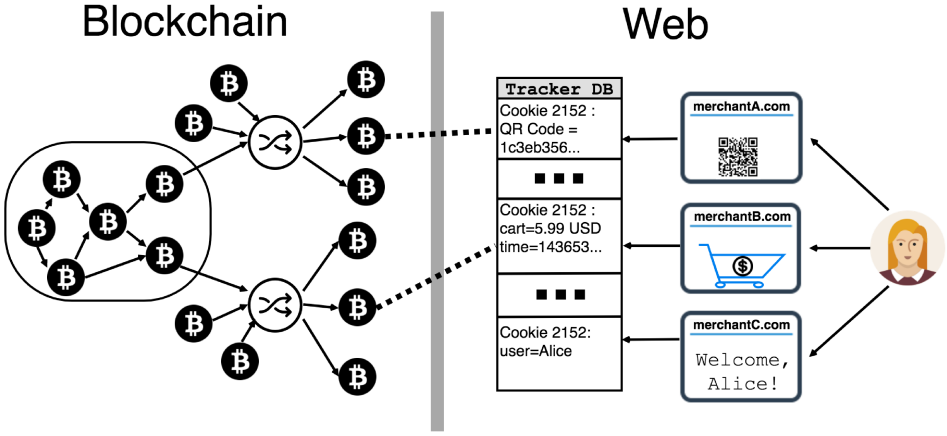

Fig. 1. An illustration of the full scope of our attack.
Consider three websites that have the same embedded tracker. Alice makes purchases and pays with Bitcoin on the first two sites, and logs in on the third. Merchant A leaks a QR code of the transaction's Bitcoin address to the tracker, merchant B leaks a purchase amount, and merchant C leaks Alice's PII. Such leaks are commonplace today, and usually intentional (Section 4). The tracker links these three purchases based on Alice's browser cookie. Further, the tracker obtains enough information to uniquely (or near-uniquely) identify coins on the Bitcoin blockchain that correspond to the two purchases. However, Alice took the precaution of putting her bitcoins through CoinJoin before making purchases. Thus, either transaction individually could not have been traced back to Alice's wallet, but only one wallet participated in both CoinJoins, and it is hence revealed to be Alice's.

pers — web trackers are in the ideal position to carry out this attack.

Note a limitation of the cluster intersection attack: it can only link addresses that were in the same cluster before mixing was employed. Throughout this paper, we assume that the user's wallet consists of a single cluster of addresses. In reality, a user wallet may have multiple disjoint clusters that our attack will not be able to link.

Based on the above two attacks, we present the following findings. We present a taxonomy of information leaks to trackers on e-commerce websites. We focus on leaks that allow linking a payment flow to a blockchain transaction. We compiled a list of 130 online merchants that accept Bitcoin, and analyzed their websites by extending the functionality of the open-source OpenWPM web privacy measurement tool [16]. We find that at least 53/130 of merchants leak payment information to a total of at least 40 third parties, most frequently from shopping cart pages. The vast majority of these represent *intentional* sharing of purchase data with third parties for advertising and analytics purposes. In addition, we find that many merchant websites have far more serious (and likely unintentional) information leaks that directly reveal the exact transaction on the blockchain to dozens of trackers.

Turning to the Bitcoin blockchain, we use empirical measurement to estimate the uniqueness of transactions as a function of the adversary's uncertainty about the transaction's timestamp and value (Section 5). We find that unique linkage is possible in over 60% of cases for realistic values of these parameters, and that in the vast majority of cases, the anonymity set size is 5 or less. The attack degrades gracefully as the adversary's uncertainty increases. Note that in the case of the unintentional leaks mentioned above, there is no uncertainty, and unique linkage is always possible.

Next, we evaluate the efficacy of the cluster intersection attack against CoinJoin (Section 6; in Section 8 we discuss the applicability to other types of mixing). By identifying a corpus of 78,697 CoinJoin transactions on the Bitcoin blockchain over a two-year period, we present realistic simulations of a victim who mixes coins from her wallet and then makes payment transactions that are observed by the adversary. For example, if the victim employs 3 rounds of CoinJoin and the adversary observes two of the victim's payments, he can link them back to her wallet (despite mixing) with 98% accuracy. Multiple rounds of mixing increase privacy, but those gains are quickly stripped away if the adversary observes more than 2 payments.

Finally, we evaluate our attack end to end (Section 7). We made 21 purchases on 20 merchant websites. For 11 of these purchases, we used freshly mixed coins to attempt to deter linkage. There were 25 pairs of purchases

made with mixed coins for which there was at least one tracker that received leaked data about both purchases. We find that in 20 of these 25 cases, the tracker can identify the user's wallet despite the use of mixing.

Our attack highlights the dangers of pervasive web tracking: Bitcoin is often used for sensitive activities, making the compromise of Bitcoin privacy a far more serious threat than targeted advertising. In Section 8 we discuss mitigations that merchants can deploy. None is a complete solution, given the fundamental tension between privacy and the analytics needs of modern e-commerce. Indeed, most of the privacy-breaching data flows we identify are intentional and not accidental (Section 4).

The main self-defense available to users today is to use tracking-protection tools such as Ghostery or uBlock Origin, but we note several limitations. First, since our attack is passive, trackers have *already* accumulated data in their logs that enable them to retrospectively carry out the attack. Second, tracking protection tools aren't perfect and contain both false positives (resulting in broken functionality) and false negatives (resulting in missed trackers). In Section 4 we show that even with tracking protection enabled, 25 merchants still leak sensitive information to third parties. Third, merchants, payment processors, and even network eavesdroppers are potential adversaries for some of the attacks we describe, and tracking protection does not help against these adversaries. Finally, in Section 8 we also discuss how our techniques can aid law enforcement investigations.

We stress that we focus our study on Bitcoin as it is the cryptocurrency with the most adoption on merchant websites. However, our attacks apply to many cryptocurrencies; in Section 5.2, we demonstrate that our attacks hold on the Litecoin blockchain.

2 Background and Related Work

Our work brings together two previously unrelated areas of privacy research: web tracking and anonymity of cryptocurrencies. We describe each in turn.

Online tracking. Since the web's inception, the number of third parties that track and record user activity has exploded. [14, 36, 40, 56]. In this paper we use the terms third party and tracker interchangeably. Some trackers have a substantial view of users' activities across the web: Google, for instance, has a tracking presence on roughly 80% of sites [37]. Tracking methods have also become more sophisticated over time [2, 15, 17, 35, 60]. The effectiveness of tracker-blocking tools has been studied by various authors [19, 43, 66].

Some trackers like Google and Facebook are known to tie their tracking profiles to identities directly disclosed by users, but most trackers have no direct relationship with users. However, even such trackers acquire PII, often accidentally. Various studies have shown that the leakage of PII from first parties to third parties is rampant [31, 32], and the problem remains severe today.

Most trackers are legitimate businesses, but are known to use intrusive means to track users. These include misuse of HTML5 APIs for fingerprinting, such as Canvas, Audio Context, and Battery Status [16]; cross-device tracking [13]; workarounds to browser privacy features [4], and sniffing data from unsubmitted forms [26]. Many trackers have poor security on their servers and are a target for compromise for malvertising and other purposes [50, 61].

The problem of trackers observing shopping and payment flows is unlikely to go away. Consider retargeting, which is the ability to serve ads to users for items they are known to have shown an interest in purchasing. It is one of the most valuable forms of advertising [29]. The farther into a payment flow a tracker can observe a user (cart page, checkout page, etc.) the greater the interest signaled. Another major benefit is conversion tracking of ad campaigns. Having trackers on the payment flows is needed to help analyze whether a user who was served an ad actually follows through with a purchase. Other applications include fraud/abuse detection and consumer insights.

Cryptocurrencies. In Bitcoin-like cryptocurrencies, users pay by broadcasting transactions to a peer-to-peer network. Transactions are signed statements authorizing transfers from one address to another. Addresses are public keys that act as pseudonymous "account" identifiers. Transactions are recorded in an immutable, global ledger called the blockchain [11, 48].

Address clustering and mixing. It is trivial to generate new Bitcoin addresses, and most wallet software takes advantage of this feature to improve user privacy. In the normal course of operation, users end up with coins split between numerous addresses, and it may not be obvious which addresses belong to the same user (or entity). However, there are well known and well understood attacks to infer links between such addresses [3, 42, 55, 57]. These techniques have been improved upon and implemented by companies such as Chainalysis and made available via easily accessible APIs. Address clustering is not perfect, but it is a powerful at-

tack, and wallet addresses must be considered clusterable unless additional privacy-protection techniques are employed to break the link between those addresses.

Many such privacy-protection techniques are known [65]; the ones readily deployable on existing Bitcoin-like cryptocurrencies are all variants of the idea of *mixing*. The best known and most used technique is known as CoinJoin [39, 58], in which different users coordinate in order to jointly create a transaction that spends a coin of equal value from each of them, and from which each of them receives a coin of the same value. The order of outputs is randomly permuted so that the mapping between inputs and outputs cannot be deduced from the public blockchain. Services such as JoinMarket provide the ability for users to coordinate to mix their coins [47].

CoinJoin improves unlinkability by breaking the multi-input heuristic, one of the main heuristics used in address clustering. However, the susceptibility of CoinJoin (and other mixing techniques) to clustering has not yet been rigorously studied. It is known that CoinJoin transactions are at least *detectable* as such, since they involve many inputs and outputs with the same value, a highly unlikely pattern in a regular payment transaction. In other words, CoinJoin improves anonymity but does not provide unobservability [47].

Intersection attacks date back to the communications anonymity literature and are well known. Their applicability to cryptocurrency mixing is also generally understood. At least two papers mention it explicitly [10, 23], but they focus on mix participants and other intermediaries as adversaries. A 2015 blog post also mentions the attack [51]. We introduce the idea that auxiliary information to link different mixed coins is readily available to web adversaries (as opposed to behavioral patterns in earlier work, which is a much less reliable linkage mechanism). Further, we are able to empirically evaluate the attack using recently proposed techniques for identifying CoinJoin transactions on the blockchain [47] (Section 6).

Other research on cryptocurrency privacy and forensics. Gervais et al. present an attack on e-commerce purchases using cryptocurrencies: since prices are denominated in local currencies, usually close to integer multiples of the unit of currency, blockchain transaction amounts could reveal the currency and hence the location of the purchase [20]. Our work is complementary; their attack is stronger than ours in that the adversary can be anyone examining the blockchain, whereas our attack is stronger in the sense that much more information is leaked, and not just the location.

Another major route to compromise of cryptocurrency privacy, orthogonal to ours, is the linkage of transactions to the sender's IP address. An adversary who is well connected to the Bitcoin peer-to-peer network might be able to do so [8, 30]; even users who connect to the Bitcoin network over Tor are potentially vulnerable [9]. In response to these attacks, Bitcoin Core changed the protocol for how transactions are disseminated across the network in 2015. However, recent work showed weaknesses in the updated protocol [18, 27, 49]. A re-designed P2P networking protocol with strong anonymity guarantees has been proposed [64], but not yet adopted by any cryptocurrency.

In concurrent work, Portnoff et al. explore a technique similar to our transaction linkage attack [53]. In their work, linkage is a forensic technique to help identify entities behind illegal activities (sex trafficking). It is enabled by a specific feature of a specific website, backpage.com: classified ads paid for by users are posted on the website along with an accurate timestamp. This allows anyone (e.g., researchers, NGOs, law enforcement) to link an ad to the transaction on the bitcoin blockchain that represents the payment for the ad. In our work, linkage can be carried out only by specific entities, such as trackers, but we extend the linkage via cookies, PII, and blockchain analysis, none of which are applicable to the setting of Portnoff et al. Of course, their work can be viewed as a demonstration of a privacy breach affecting Backpage users, including the majority not engaging in illegal activities; similarly, our attack can be turned into a forensic technique (Section 8).

3 Threat model and attacks

Merchant, payment processor, and trackers. A typical cryptocurrency-based e-commerce flow consists of a user, a merchant, a payment processor, and one or more trackers. The merchant is the website where the user is shopping. Most merchants make use of payment processors such as BitPay and Coinbase to handle the processing of cryptocurrencies. When the user pays with Bitcoin or another cryptocurrency, the transaction is received by the payment processor, who then usually credits the merchant's account with an equivalent amount of dollars or other local currency. Trackers are "third parties" on web pages, often invisible, that track users' actions for purposes of advertising and analytics (Section 2). Doubleclick, Google Analytics, and Facebook are common examples. Merchants, payment processors,

and trackers are all potential adversaries in our attack, although we are most interested in the latter.

Information flows to third parties. Users take actions on shopping sites such as logging in, viewing items, adding items to their cart, checking out, and making a payment. See Figure 2 for a typical payment flow on a merchant site. The more of these actions a third party learns about, the more feasible the attack. The types of information useful to the third party are:

- Payment timestamp: the third party learns the approximate payment time simply by virtue of being embedded on merchant website, especially on pages constituting the checkout process. Checkout pages usually require the user to complete payment (i.e., broadcast the cryptocurrency transaction) within a short time window, typically 15 minutes. Trackers embedded on payment receipt pages are in an even better position, as they learn the payment time to within a few seconds. Note that assuming the user included a reasonable transaction fee, payment processors consider payments received as soon as the transaction is broadcast to the peer-to-peer network and received by the payment processor's node. This involves a latency of only a few seconds. The transaction may not be *confirmed* until it is incorporated into the blockchain, which may take tens of minutes depending on the degree of confirmation that the payment processor requires. The transaction confirmation time is largely irrelevant to our attack.
- Payment address: the payment address is the destination to send coins. Recipients (payment processors) will typically generate a fresh address specific to the transaction — new Bitcoin addresses are trivial to generate. Although there is no business reason for trackers to receive the payment address, we find that this does happen often (Section 4). Since payment addresses are unique, at least within the time scale of interest to us, a leak of the payment address trivially allows the tracker to link the web user to the blockchain transaction.
- Price: Depending on the merchant website, trackers may be able to see the prices of items viewed by users, items added to the cart, or even the final price after shipping and taxes have been calculated. Note that these prices are almost always denominated in USD, EUR, or another fiat currency, even on websites that only accept cryptocurrencies as payment, due to the extreme volatility of cryptocurrency exchange rates. However, once the user checks out, the amount is calculated in BTC (or another cryptocurrency) based on the exchange rate at that instant. In some cases, this BTC-denominated price is also revealed to the tracker, which is more useful for linking than the price before applying the conversion.
- Personally Identifiable Information (PII): By PII we mean any information related to the user's real identity or account on the merchant website, such as name, email address, username, and shipping address. Trackers' access to PII exacerbates the attack. In this paper we analyze leaks of PII from merchant websites to trackers, but we emphasize that since trackers are widely present on the web, the link to PII can be acquired on any website whatsoever. Leaks of PII to trackers are known to be rampant across the web (Section 2).

In our measurements in Section 4, we focus on passive attacks where trackers obtain this information in the normal course of operation. Except for (some) PII leaks, most other information flows to trackers are intentional: trackers use this information for advertising and analytics purposes. However, we note that in many cases, tracking scripts are in a position to carry out an active attack and extract all of the above information from web pages even if they don't obtain it passively. This is because third-party scripts are typically embedded without any isolation, in a way that gives them full access to the content on the page. Sandboxing techniques such as iframes are readily available, but only infrequently employed since they interfere with some of the functionality provided by trackers.

Attack 1: single transaction linkage. In this attack, the adversary (tracker) seeks to link a web user (as identified by the user's cookie or PII) to a transaction on the cryptocurrency blockchain. The merchant and payment processor are not interesting adversaries for this attack, because it is unsurprising that they can carry out this linkage (but see Section 4.3). We assume that the user is aware of this possibility, and potentially takes necessary precautions, such as mixing to unlink the transaction on the blockchain from her *other* blockchain transactions and addresses. Attack 2 seeks to overcome such defenses. But the tracker's ability to link to even a single transaction is a privacy breach, because the user has no business relationship with the tracker and many users are in fact unaware of the existence of trackers (or at least their prevalence and sophistication). It is also worrisome because trackers compile profiles of users' activities across the web.

If the tracker has access to the receiving address, it trivially enables linkage, as noted above. The more

Fig. 2. An illustration of a typical payment flow on a merchant site. Each step of this flow presents opportunities for leaking transaction-relevant information to embedded third-party trackers.

interesting case is when the tracker knows the approximate price and time. Then the tracker's task is to search the logs of transactions that were broadcast to the peer-to-peer network to identify those that fall within the window of uncertainty both in terms of transaction value and time. To quantify the tracker's success, then, we must model the uncertainty in the tracker's knowledge of price and time.

- Price uncertainty: The tracker's uncertainty around price arises primarily from shipping. If the tracker knows the adversary's location (either based on a leak of PII or based on IP address), this uncertainty can be minimized, although there might still be a small number of possible values of the shipping fee based on the shipping speed selected by the user.
- Exchange rate uncertainty: The second source of uncertainty is the exchange rate: the tracker usually sees prices denominated in USD (or another fiat currency) and not in BTC. Most payment processors use exchange rates based on trading data publicly released by an exchange, which means the tracker can always reconstruct the exchange rate at any given point in the past. However, since trades happen several times per second, the exchange rate varies rapidly and hence some uncertainty will still remain.
- Payment time uncertainty: this arises because of the gap between the user checking out, the user's wallet broadcasting the transaction, and that broadcast being recorded by the adversary or another node. The adversary may run his own peer-to-peer Bitcoin node, or may simply obtain the transaction broadcast timestamp from publicly available sources such as blockchain.info. If the tracker is present on the transaction receipt page, then the latency is minimized, and is of the order of the network propagation delay, i.e., a few seconds.

Attack 2: Cluster intersection. This is a complementary attack where the adversary aims to identify the cluster of addresses in the victim's Bitcoin wallet. Recall from Section 2 that wallets can (and do) easily create numerous addresses, but in the normal course of operation these addresses can still be linked together via various heuristics. Mixing techniques such as CoinJoin are thought to protect against such linkage, although this has not yet been studied rigorously. We assume that the victim uses a desktop (local) wallet rather than an online wallet provider. Privacy-conscious users tend not to use online wallets, since that would allow the wallet provider to trivially track all of the user's activities. We also assume that the user employs effective communications anonymity techniques to mask the IP addresses of their wallet, as that is a well-known way for anonymity to be compromised (Section 2).

In our attack, the victim interacts with the adversary multiple times. The adversary could be a merchant, payment processor, or (especially) a tracker who only indirectly observes the victim. Knowing that the adversary might learn one of his addresses, the victim employs mixing to prevent the adversary from learning the rest of his addresses and transactions. He doesn't spend coins directly from his wallet, but only after first mixing them. In Figure 1, after the victim has shopped on *merchantA.com* the adversary is unable to determine which of the three wallet clusters belongs to the victim. But after a second interaction with the same victim on *merchantB.com*, the adversary simply finds the intersection of the two sets of clusters, which leads him to a unique cluster.

Web trackers passively observe users' web purchases and are able to link them together, via cookies or device fingerprinting, even if the merchant and payment processor are different in every case. Thus, this attack is complementary to Attack 1, and would take as input two blockchain transactions identified via Attack 1. Note that even if Attack 1 is imperfect, and returns a set of transactions instead of a single one, Attack 2 will still succeed. The intersection size rapidly decreases as a function of the number of observations, and even if two observations aren't sufficient to uniquely identify

the wallet, it is likely that a small number of additional observations will suffice. We quantify this in Section 6.

4 Web measurement: Leaks of sensitive data

In this section we analyze leaks of sensitive data on merchant sites. In sections 5 and 7 we examine how trackers can actually use this data to identify transactions on the blockchain. We also show in this section how trackers can connect this information to users' identities.

4.1 Method

To identify leaks of sensitive data, we performed a web crawl of popular merchants that accept Bitcoin. To create a list of merchant sites, we began by combining popular community-maintained lists of merchants [54, 62], which gave us 1438 sites. We then pruned the list to those domains that were found in the Alexa top 1 million websites, which left 283 sites. As we crawled the sites, we discovered that about half of the merchants no longer accepted Bitcoin. This left 130 merchants in our crawl that accepted Bitcoin at the time of our measurements, and we focus on these 130 sites. These merchants were geographically distributed over 21 countries, with 64 based in the United States and 20 based in the United Kingdom.

Typical merchant payment flows allow us to complete most of the steps — viewing products, adding them to the cart, initiating checkout, and receiving a payment address and price — before actually having to make a payment. This allowed us to collect data on almost the entire payment flow on a large number of websites. However, to analyze payment receipt pages, we need to actually make purchases. Thus we analyze transaction receipt pages on a smaller scale based on actual purchases. We made purchases from 20 distinct merchants in total.

To collect data on web tracking we used a modified version of the open-source web privacy measurement tool OpenWPM [16]. Using the tool we collected all HTTP(S) requests and responses. We also manually marked any PII and payment-related information that we encountered on the pages we visited; we added functionality to the tool to automatically record this information when marked.

BitPay	**70**
Coinbase	**24**
Coinpayments	**3**
Stripe	**2**
Other	**31**
Total	**130**

Table 1. Prevalence of payment processors in 130-site crawl

Throughout our measurements, we are interested in the privacy risk both for a regular user and for a user who employs tracking-protection tools. Most such tools (e.g., Adblock Plus, uBlock Origin) use standard, community-maintained *filter lists*: EasyList and EasyPrivacy. To measure the privacy-risk for users of tracking protection lists, we simply re-run our analysis after deleting those third-party URLs in our crawl databases that appear in the lists.

4.2 Findings

Based on our measurements of 130 Bitcoin-accepting merchants, we found numerous third parties that receive transaction-relevant information by virtue of their business relationship with the merchant in the normal course of a transaction. We define transaction-relevant information as any information that could help identify the transaction on the blockchain. These potential adversaries could retroactively perform the transaction-linkage attack using data already present in their HTTP logs or databases. We measure information received through either unintentional leakage, via the referer field of an HTTP GET or POST request [32], or through intentional information sharing via an HTTP POST action or a GET URL parameter.

Third parties receiving Bitcoin address or BTC price

Almost all merchants use third-party payment processors; this helps them avoid the security, volatility, and legal risks of receiving and holding bitcoins. Table 1 lists the prevalence of different payment processors in our crawl of bitcoin-accepting merchants. The payment processor either sits in an `iframe` on the checkout page, or on a separate page in the payment flow. The frame or page will display the exact Bitcoin amount the user should send, to an address controlled by the payment processor.

We found that 17 of the 130 Bitcoin-accepting merchant websites send the receiving Bitcoin address or BTC-denominated price to a

Info type	w/o protection	w/ protection
Non-BTC-denominated price, incl. shipping	24	12
Non-BTC-denominated price, pre-shipping	23	5
Non-BTC-denominated price, either	43	16
Bitcoin address	12	12
Bitcoin price	11	9
Bitcoin address or price	17	15
Add-to-cart events	28	2
Total merchants sharing info	53	25

Table 2. Number of merchants sending transaction-relevant information to third parties, with/without tracking protection

Info type	w/o protection	w/ protection
Non-BTC-denominated price, incl. shipping	29	11
Non-BTC-denominated price, pre-shipping	18	3
Non-BTC-denominated price, either	38	13
Bitcoin address	5	4
Bitcoin price	4	2
Bitcoin address or price	9	6
Add-to-cart events	9	2
Total third parties receiving info	40	13

Table 3. Number of third parties receiving transaction-relevant information, with or without tracking protection

third party (Table 2). With this information, linking the payment to the blockchain is trivial. The leaks were found on less-popular payment processors and websites that implement their own Bitcoin payment processing.

We can also break it down by third parties instead of merchants: see Table 3. In both tables, we also present the corresponding measurements for users of tracking-protection tools (via the EasyList and EasyPrivacy[2] filter lists). Those results are presented in the "with protection" columns. See Appendix A.2 for a list of third parties that receive transaction-relevant information despite the use of tracking protection.

On 11 out of the 12 websites that leak the Bitcoin address, the leaks were to third-party services that *render QR codes to facilitate payment*. Providing a QR code encoding the payment recipient's Bitcoin address and the Bitcoin price makes payment easier for the user. QR-code generator services generally work by accepting a GET request with the data encoded as a query parameter, and returning the rendered QR code image. Thus, transaction-relevant information is contained in the GET request (e.g., https://blockchain.info/qr?data=bitcoin://[address]?amount=[price]&size=180). If the QR-code generator service stores HTTP requests in their logs they will have passively collected sufficient information to perform the blockchain analysis attack. We saw three domains providing this service: chart.googleapis.com, qrserver.com, and blockchain.info. Three payment processors in particular use chart.googleapis.com to generate QR codes: coingate.com, litepaid.com, gourl.io.[3]

We made purchases on a subset of 20 merchants websites, which allowed us to examine trackers on payment receipt pages. Table 7 in Appendix A.2 presents the number of third parties found on each merchant's payment receipt page, and the number of third parties that also receive transaction-relevant information in the course of the payment flow. Embedded third parties are common on receipt pages: in total, there were **245 third parties on the 20 merchant receipt pages** we visited.

We found further serious leaks of sensitive information on some of these pages. In particular, the payment processor Coinbase redirects to receipt page URL on the merchant website (such as https://www.overstock.com/bitcoinprocessed/?...), and appends to this merchant URL a long string of query parameters that include the Bitcoin payment address. If the resulting payment receipt page embeds third parties, then the merchant will (likely inadvertently) leak the payment address via the HTTP referer header. We found this to be the case on multiple merchant websites that use Coinbase. **The overstock.com receipt page alone leaked the payment address to 42 distinct third parties via this referer leakage.**

Additionally, we found that many merchant websites leak payment processor invoice page URLs to third parties. This is a different type of leak from the one in the previous paragraph. The URLs themselves do not contain sensitive information, but the contents of the invoice pages do, in the case of both Coinbase and Bitpay. In both cases, the invoice page is not protected by access control and the content can be viewed by anyone

2 https://easylist.to/

3 Google's policy is to retain these log for 2 weeks, for debugging and development purposes [21]. We could not find the retention policies for the other two service providers.

who has the URL. Of the sites from which we made purchases, 12 of the 20 merchants included leaks of these URLs to a total of 25 third parties.

Third parties receiving non-BTC cart prices

The cart page displays each product in a user's shopping cart, along with the non-BTC denominated (e.g., USD or EUR denominated) subtotal of the cart. This subtotal will often exclude taxes and shipping. The user is often directed to enter their shipping address at the following checkout page, which will then calculate the shipping fee and add it to the cart subtotal.

From our crawl data we identify a second set of third parties that receive the non-BTC-denominated cart price. If the received cart price is missing shipping and handling, it will increase the adversary's uncertainty about the BTC-denominated price (see Section 5). As seen in Table 2, **43 out of the 130 bitcoin-accepting merchants we visited send some form of non-BTC-denominated cart price data to third parties** — many more than share BTC price or address with third parties.

Based on the type of HTTP request that sent the transaction-relevant information to a third-party, we can categorize whether the sharing of data was intentional or unintentional. We consider an unintentional data leakage a sharing of data that happens solely via referer leakage. While it is possible that the third party parses the referer for the price information on the backend, we find it is useful to separate these cases from an *intentional* sharing of data. An intentional share with a third party means that the price was sent in the URL of a GET request or the body of a POST request — in that case, the request was intentionally constructed. In our crawl, we found that **the overwhelming majority of requests that shared transaction-relevant data with a third party were intentional**: of the 312 requests we observed on 53 merchant sites sharing transaction-relevant information, 295 intentionally shared data.

To perform the cluster intersection attack, an adversary must have transaction-relevant information for at least two separate purchases. A third party positioned on more than one website is in a better position to have the necessary data. Table 8 in Appendix A.2 contains the prevalence of third parties receiving transaction-relevant information that appeared on at least two merchant sites. As one might expect, Google Analytics and Facebook are pervasive.

Third parties receiving product page visits

At minimum, on an e-commerce site a product page will display the non-BTC denominated price of an item that a user can purchase. A product page will also often embed resources from many third-party domains. As an illustrative example, the product page https://missionbelt.com/collections/solid-color-40mm-belts/products/vader-40, includes resources from *31* distinct third-party domains.

Thus, third-party trackers could infer the user's cart subtotal based on the product pages they visit. In our data we found several examples of third parties that not only see the product pages a user visits, but know exactly when the user adds an item to their cart. In total, **28 bitcoin-accepting merchants in our crawl shared add-to-cart events with third parties.** The most common third party to receive this information was Facebook, which received add-to-cart events on 26 merchants' sites. Being able to receive add-to-cart events is essentially equivalent to being able to see pre-shipping cart prices. Only two of those merchant sites send add-to-cart events to third parties if the consumer uses browser tracking protection.

Third parties receiving transaction timing

As discussed in Section 5, an adversary needs to know the approximate timing of a Bitcoin transaction in addition to its value. There are several ways in which third parties already have this information stored in logs.

As discussed earlier, when a user completes a Bitcoin transaction, the payment processor typically redirects the user back to a receipt page. Any third parties loaded on the receipt page who had previously seen transaction-relevant data can then use the receipt page load-time as the approximate timestamp of the bitcoin transaction (except during periods of anomalously high network load). In our data, 245 distinct third-party domains had resources loaded on merchant receipt pages on sites for which we made a purchase. **16 out of 20 of those merchant sites embedded on the receipt page at least one third party which had previously received transaction-relevant information**. This knowledge makes blockchain linking much easier. A more detailed breakdown of third parties by merchant receipt page can be found in Table 7 in Appendix A.2.

Even without seeing a bitcoin transaction receipt page, third parties could still estimate the time a transaction took place based on whether or not they know when a user starts the checkout process. From our data we measured the extent to which Facebook

PII type	w/o protection	w/ protection
email	32	25
firstname	27	20
lastname	25	19
username	15	12
address	13	9
name	11	4
phone	10	4
company	5	4
Merchants sharing PII	49	38

Table 4. Number of merchants sharing each type of PII, with or without tracking protection

trackers, for instance, explicitly track a user through their API's "`InitiateCheckout`" event. The Facebook `InitiateCheckout` event was found on 15 sites of the 130 bitcoin-accepting merchants we visited in our crawl.

Third parties receiving PII

A leak of transaction-relevant information coinciding with a leak of PII allows the adversary to attach a real world identity to a Bitcoin address. In our crawl, **49 bitcoin-accepting merchants leak some form of PII to a total of 137 third parties**. Table 4 lists the number of merchants that share each type of PII with a third party. Table 6 in Appendix A.2 lists the number of third parties receiving each type of PII.

We also found that **21 third parties that receive PII also receive transaction-relevant information**. Those third parties can conduct the blockchain analysis attack and add a label to the resulting cluster.

4.3 Other adversaries

Some entities do not yet collect the necessary information to conduct the blockchain attack, but could be in the position to collect the information through an active attack.

Active attacks by third parties

Third-party JavaScript has access to the complete DOM of whatever frame the script is embedded in: if that frame contains a piece of transaction-relevant data, then a script that turned malicious could go out of its way to collect the data if it wasn't already collecting it.

OpenWPM allows us to match third-party scripts to the pieces of information they can read. In short,

Bitcoin address	31
Non-BTC-denominated price	104
Bitcoin price	30
Total sites leaking transaction-relevant info	107

Table 5. Active attacks: Number of merchants that allow third-party script access to transaction-relevant information

107 sites in our crawl grant third-party scripts access to transaction-relevant information (Table 5). The most-prevalent third party by far is google-analytics.com, which appears to be widely trusted: for example, a google-analytics.com script is often the *only* script found on Coinbase's payment processing page.

Potential third-party access to PII was even more prevalent: on the 130 bitcoin-accepting merchants we crawled, **125 merchants granted third-party scripts access to some form of PII.** This included the scripts of payment processors that may not otherwise receive PII.

Network adversaries

While most of the merchants we visited in our crawl use HTTPS, some failed to do so. In our crawl, **we found 36 merchants that did not use HTTPS for crucial parts of the payment flow, including cart pages**. A network adversary could thus see the page load in cleartext, and parse transaction-relevant information or PII from the page.

Payment processors

While users are arguably aware that payment processors will receive transaction-related information in the course of a payment, they probably do not expect that payment processors receive PII, since creating an account with the payment processor is generally not required. Yet, we found that **at least 24 merchants share some form of PII with BitPay**, even though BitPay does not require that the merchant send them PII.

Merchants

Conversely, users may expect that while merchants necessarily receive PII, they may not be able to easily identify the transaction on the blockchain. But we made merchant accounts with BitPay and Coinbase, and found that they both share the full details of the Bitcoin transaction with the merchant.

5 Blockchain analysis: Estimation of linkability

Having shown that trackers obtain payment-related information from online purchases, we now present empirical analyses of the Bitcoin blockchain to show that trackers can use this information to uniquely identify the transaction on the blockchain. We reiterate that many trackers will have no need for the techniques in this section since merchants share unique transaction-specific information with them.

For blockchain analysis, we used BlockSci [28], an in-memory blockchain graph database and query interface that supports Bitcoin and various altcoins.

5.1 Method

We seek to answer the question: for a given level of price uncertainty, exchange rate uncertainty, and transaction time uncertainty, what is the distribution of anonymity set sizes of the transaction? The anonymity set contains candidate transactions, one of which represents the actual payment. Based on the anonymity set sizes, we also compute adversary success probabilities. Throughout, we average our measurements over a set of prices (obtained from actual sites, as described below), and a set of random points in time over a two-year period from mid-2015 to mid-2017.

Exchange rate data. Recall from Section 2 that payment processors use price data from exchanges, which is also available to adversaries. In our measurements, we use publicly available historical data from BitStamp made available by bitcoincharts.com.[4] The data contains the prices of all trades executed on the exchange, starting in September 2011. As of June 2017 it contains 11.6 million trades. During the time period of interest to us it contains about 4.4 million trades, or about 4.2 trades per minute.

Sampling prices. To obtain a representative sample of prices of user purchases, we sampled 100 item prices from our dataset of merchants. We sampled merchants randomly and then sampled items randomly from those listed on the homepages of those merchant websites, ensuring a maximum of 10 from any one merchant. When converted to USD, the prices ranged from a minimum of 1.52 to a maximum of 359.00, with a mean of 51.27 and a median of 24.99.

Sampling actual prices is important, because the distribution of values of e-commerce payments is different from that of other transactions on the blockchain. For example, prices are often close to integer multiples of the currency of account (USD, EUR, etc.) [20]. Therefore, if we sample prices directly from blockchain transactions, we might obtain unrepresentative results.

Sampling times. We pick 100 random timestamps from our time period of interest. By picking these randomly instead of periodically, we ensure that there are no patterns such as specific times of day or day of week.

Modeling price uncertainty. By analyzing the behavior of various merchants, we make the following key observation: *if the adversary knows only the country of the user's shipping address, there are only a small number of possible values (typically fewer than 10) for the difference between the cart price and the final price.* For example, there are no instances where each US state has a distinct shipping fee. Based on our observations, the vast majority of merchants do not collect tax. Note that often the shipping address is directly revealed to the tracker (Section 4). Even otherwise, it is easy for trackers to learn the user's country, whether based on IP address, locale, or language. Thus, we model price uncertainty as a small set of possible values. We sample these values from the list of actual shipping rates on the merchants we analyzed. We always include the value 0 in this sample, because free shipping is (unsurprisingly) a popular option.

Modeling time uncertainty. The tracker may observe the user take one or more of the following actions: view the shopping cart, initiate checkout, and view the transaction receipt page. In the first case, the tracker can guess that the user may have initiated payment within a few minutes (though this guess might be incorrect). In the second case, the tracker knows that if the payment was made, it would have to be within a time window set by the payment processor, typically 15 minutes. In the third case, the tracker knows the transaction timestamp to within a few seconds (network latency).

Modeling exchange rate uncertainty. The exchange rate data used by the payment processor is always available to the adversary. However, there is potentially some uncertainty introduced by the lag between the tracker observing the user initiate the checkout process and the user being quoted an exchange rate. We model the adversary's uncertainty as a time interval. If this window is 5 minutes, it means that the exchange rate that was applied to the transaction could take any

4 https://api.bitcoincharts.com/v1/csv/

value from the published time-series of exchange-rate values during a 5-minute period that begins when the adversary observed the user initiate checkout.

Modeling the victim and adversary. We simulate 10,000 payment flows based on all combinations of the 100 prices and 100 timestamps sampled as above. For each flow, we consider two cases: the victim either does, or doesn't, complete the payment within the windows of uncertainty assumed by the adversary.

We posit an adversary that behaves as follows:

– if there is exactly one transaction that falls within the uncertainty intervals, output that transaction.
– if there is more than one such transaction, output a random transaction from that set.
– if there is no transaction, output "no transaction".

The last point is important, because in many cases the adversary observes the user on the shopping cart page or the checkout page, but does not know for sure that the payment was made (some adversaries are also present on transaction receipt pages and will have this additional information).

Quantifying success. We quantify the adversary's success in terms of two numbers: the true positive rate and the true negative rate. A true positive occurs when the victim completed the transaction, and the adversary outputs the correct transaction. A true negative occurs when the victim did not complete the transaction, and the adversary correctly outputs "no transaction".

For each simulated payment flow and each set of uncertainty parameters, we search the log of broadcast transactions ("mempool" log) for transactions that match the price and time within the specified uncertainty windows. Transactions found, *in addition to the payment itself*, constitute the anonymity set. Since the payments are simulated, we do not expect to find them on the blockchain, but in a real attack the payment would be part of the blockchain. In other words, our measurements answer the question: "had a payment of a given value been made at a given time, how many transactions on the blockchain could it be confused with?" The anonymity set size is 1 more than this value.

Additional heuristics. E-commerce payment transactions have several other characteristics that enable the adversary to distinguish them from (some) other transactions on the blockchain. We incorporate several such heuristics in our attack.

– Payments are always made to regular addresses rather than high-security "multisignature" addresses. This is true across almost all 130 merchants that we analyzed. The use of multisignature addresses would make our attack far stronger since the attacker, knowing the type of address used by any given merchant, would be able to limit the set of candidate transactions on the blockchain.
– Payment transactions almost always have two outputs — the recipient's output and the change output — and never more than two. This behavior is consistent across all but one user wallet software that we are aware of; the exception is Samourai Wallet (https://samouraiwallet.com/).
– Fresh addresses are used, both for change and for the recipient's output. This is a conservative assumption; alternative behavior would make our attack stronger. If the user's wallet reuses addresses for change, that would undo the effect of mixing. If the recipient reuses addresses, it would make it easier for the adversary to associate specific addresses with recipients, and thus further filter the set of candidate transactions on the blockchain.

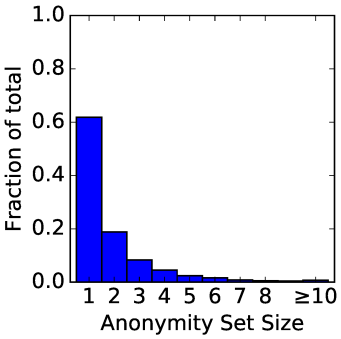

Fig. 3. Anonymity set size of the single-transaction linkage attack that aims to link a web transaction to the blockchain. The price uncertainty set size is 5, the payment time uncertainty is 15 minutes, and the exchange rate uncertainty is 5 minutes.

5.2 Results

Anonymity set size. Figure 3 shows the distribution of anonymity set sizes under default values of various parameters: payment time uncertainty of 15 minutes, exchange rate uncertainty of 5 minutes, and a price uncertainty set size of 5. The most common value of the anonymity set is 1, which shows that the attack is powerful under this default set of parameters. Based on the anonymity set size distribution, the true positive rate is 76% and the true negative rate is 62%.

Impact of uncertainty. Having shown that the attack is successful under a default set of parameters for uncertainty, we examine the impact of each uncertainty parameter. In Figure 4 we see that the accuracy remains high even if there are 10 possible values for the price. As we observed earlier, price uncertainty arises

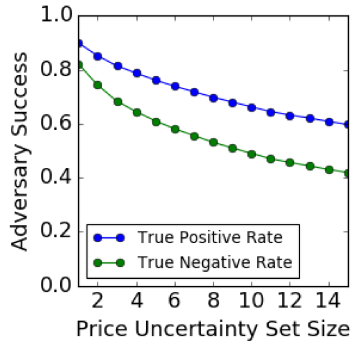

Fig. 4. Effect of price uncertainty on transaction linkage

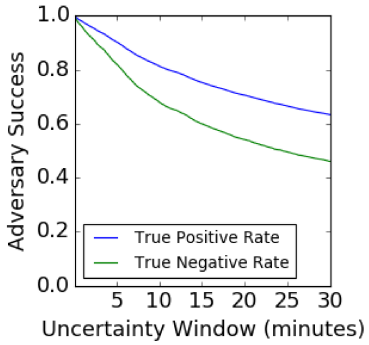

Fig. 5. Effect of payment time uncertainty on transaction linkage

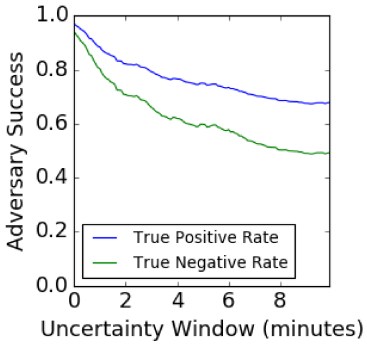

Fig. 6. Effect of exchange rate uncertainty on transaction linkage

due to shipping options, and there are rarely more than 10 possible values for it for any given country.

The attack degrades gracefully when we increase the adversary's time uncertainty or exchange rate uncertainty (Figures 5, 6). Note that if the payment processor automatically redirects to the payment receipt page, and the adversary is embedded on this page, then the time uncertainty is on the order of seconds, and the success rate is extremely high.

More generally, the adversary will have a high success rate if his uncertainty on *at least one* of the three dimensions is low (Figures 4, 5, 6), as this greatly cuts down the number of possible matching transactions.

Robustness of the results. While we took care to sample prices from the actual distribution of prices on merchant websites, we find that our results are robust in terms of the sampling strategy. For example, we repeated our experiments with prices sampled from the distribution of transaction amounts on the blockchain (Figure 10 in Appendix A.2). The results are very similar; the accuracy improves slightly. We also repeated our experiments with all prices doubled, i.e., with each sampled price replaced by twice its value (Figure 11 in Appendix A.2). Again the results are essentially unchanged.

We also repeated our experiments on the Litecoin blockchain instead of Bitcoin. Litecoin is the altcoin with the most adoption for online payments, in terms of support by merchants and payment processors. Again we find that the success rate is high (Figure 12 in Appendix A.2); in fact, it is higher than the success rate for Bitcoin, likely due to Litecoin's lower transaction volume, and therefore smaller anonymity sets. Litecoin had a volume of 3,605,028 transactions in the two-year period of interest, as opposed to Bitcoin's 150,614,721.

Further improvements. So far, we have made conservative assumptions about the adversary's knowledge. The success of the attack in practice may in fact be much higher, either due to idiosyncratic behavior by payment processors or due to additional information available to the adversary.

BitPay, one of the two main payment processors, rounds its transaction amounts (in Satoshis) to a multiple of 100. Since the adversary knows the identity of the payment processor, whenever that processor is BitPay, he can eliminate a large fraction of possible transactions — any transaction amount that is not a multiple of 100 Satoshis. Applying this heuristic, the accuracy improves substantially (Figure 13 in Appendix A.2).

Even if there is no discernible pattern in the transaction amount, the adversary may be able to tell which (if any) payment processor was involved in any given transaction on the blockchain. Such address tagging heuristics are well known [42], and are applied at scale by companies such as Chainalysis. Tagging is not always accurate, but it can help the adversary greatly decrease the anonymity set. This technique was used for Bitcoin forensics in a recent paper [53].

6 The cluster intersection attack

We now turn to our second attack, the cluster intersection attack (Algorithm 2). To recap, the attack is applicable when the adversary has auxiliary information revealing that two (or more) transactions made with mixed coins trace back to the same wallet (address cluster). Web trackers who observe multiple purchases may have this information.

In this section, we present a large-scale simulation of the effectiveness of this attack. An empirical validation,

Algorithm 1: Clustering. One step of the address clustering algorithm. We invoke this function recursively to find all addresses associated with a given coin or address. The algorithm incorporates the multi-input and change-address detection heuristics from [42]. Bitcoin mixing today is dominated by JoinMarket, so we use JoinMarket detection (Appendix A.1) in place of IsMixTx.

```
1:  function ExpandCluster(addr)
2:      C ← {addr}
3:      for all tx in TxsFrom(a) do
4:          if not IsMixTx(tx) then
5:              C ← C ∪ FromAddresses(tx)
6:              C ← C ∪ {ChangeAddress(tx)}
7:      for all tx in TxsTo(a) do
8:          if not IsMixTx(tx) and
9:                ChangeAddress(tx) = addr then
10:             C ← C ∪ FromAddresses(tx)
        return C
```

where we de-anonymize our own wallets, is deferred to the next section.

6.1 Method

Identifying joins. First we identify existing CoinJoin transactions on the Bitcoin blockchain. We focus on JoinMarket, since it is (to our knowledge) the only decentralized mixing service that is currently operating and has a usable level of liquidity. We adapt Möser et al.'s algorithm to identify JoinMarket transactions [47], and it is shown in Algorithm 3 in Appendix A.1. We found 95,239 such transactions, of which 78,697 are during the period of interest to us (mid 2015–mid 2017). The number of coins mixed in one of these transactions has a mean of 3.98 and a standard deviation of 1.72.

Simulating the victim. We consider a victim with a wallet of clusterable addresses who obtains 100 distinct mixed coins over the two-year period of interest. We sample 100 timestamps (block heights) uniformly during this period; at each of these times, the victim initiates mixing of a coin from her wallet and completes $r \leq 5$ rounds of mixing. 5 rounds represents a very high degree of anonymity based on JoinMarket's advice to users [1]. The victim retains the mixed coins until the end of the period of interest. The values of these coins don't matter since this information is not used by our deanonymization algorithm.

Algorithm 2: Cluster Intersection Attack. Step 1 can be amortized over multiple invocations of the algorithm; alternately step 2 can be modified so that join detection can be performed only as needed.

Inputs:

- a set of mixed coins C known to be controlled by the same user.
- an integer r, representing the adversary's (possibly incorrect) assumption that the victim did at most r rounds of mixing.

Output: a wallet cluster.

1. Identify all join transactions on the blockchain.
2. For each coin $c \in C$:
   - Identify all coins x such that there is a directed path from x to c of length at most r consisting only of join transactions. Call this set X_c.
   - For each coin $x \in X_c$, identify the wallet cluster it belongs to (Algorithm 1). Call the resulting set of wallet clusters W_c.
3. Compute the wallet cluster intersection: $\bigcap_{c \in C} W_c$.
4. If it results in a unique wallet cluster, output it. Otherwise output "incorrect assumptions".

To simulate the mixing of one coin (with r rounds of mixing) starting from a given timestamp, we sample from the JoinMarket transactions on the blockchain that have this timestamp. With this node as the starting point in the graph of JoinMarket transactions, we sample a path of length r from among all such paths. If there are no such paths, we repeat the procedure starting from a different initial transaction.

Attack. At this point the victim has 100 mixed distinct coins in her (simulated) wallet. Now we simulate the web tracker's view, that is, we simulate the victim making two transactions in a way that reveal to the adversary that two of these coins trace from the same wallet. Then we execute the cluster intersection attack (Algorithm 2). We repeat the procedure with different values of the number of rounds r and the number of transactions t observed by the adversary.

6.2 Results

Figure 7 shows the adversary's success rate as a function of the number of mixing rounds and the number of transactions observed. By construction of the exper-

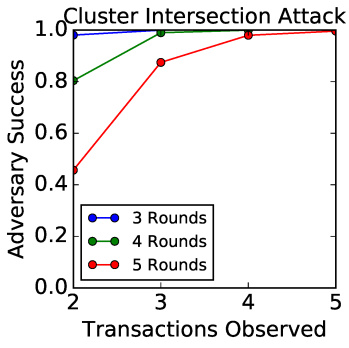

Fig. 7. Effect of rounds of mixing on success rate of cluster intersection attack. For $r < 3$, the success rate is 100%.

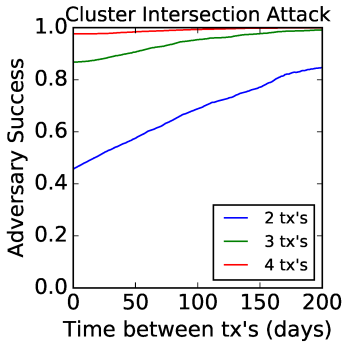

Fig. 8. Effect of age of mixed coins on success rate of cluster intersection attack (with 5 rounds of mixing).

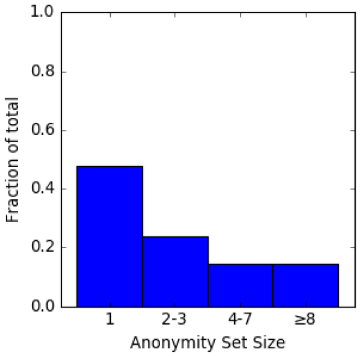

Fig. 9. Observed anonymity set sizes in empirical evaluation of the transaction linkage attack

iment, the cluster intersection attack has the same true positive rate and true negative rate. Thus the graph also represents the probability that, if the adversary is incorrect about the number of rounds of mixing, it will output "incorrect assumptions". With one or two rounds of mixing, just two observed transactions are sufficient for the adversary to identify the wallet cluster. Even with four rounds of mixing, a small number of observations is sufficient for high accuracy.

Figure 8 helps explain why the attack succeeds: the success rate is strongly dependent on the difference in age between the mixed coins. This is intuitive: if the victim mixed a coin a year ago and another coin today, the anonymity sets of the two coins are much less likely to intersect, compared to two coins both mixed today. In other words, users who have a long history of making e-commerce purchases using mixed coins are at a greater risk of deanonymization, not just because of the number of purchases but also because of the gap between them.

7 Empirical validation of attacks

We now describe how we validated our attacks empirically by making actual purchases and participating in CoinJoin transactions. Naturally, the scale of these experiments was more limited than our simulations.

Setup. We began by purchasing bitcoins from an exchange, Coinbase, and routing them to six addresses. We ensured that these six addresses are clearly clusterable by our clustering algorithm (Algorithm 1). This simulates a user with a wallet containing addresses that are linkable to each other before mixing. As discussed in Section 2, we believe that this is a conservative and realistic assumption. Furthermore, the clusterability of the user's wallet is affected by factors not in the user's control. For example, if a payment processor provides a payment address that has been used before, rather than a freshly generated address, then the user's change address will be linked to her wallet.

Next, we participated in JoinMarket CoinJoin transactions to create 11 coins which are not linkable to the main cluster using known techniques. We participated in one round of CoinJoin for 6 of the transactions and two rounds of CoinJoin for the other 5.

Finally, we made a set of 21 purchases on 20 merchant sites. We sampled these sites from among those that leaked transaction-relevant information to at least one tracker (as measured in Section 4). For 11 of these purchases, we used coins that had been mixed in the previous step, and we ensured that these addresses as well as the change addresses for these purchases did not get linked to our cluster. For the other 10 purchases, we used coins directly from our cluster. The final prices of these items ranged from $3.28 to $46.40 when converted to USD, with a mean of $13.67.

Validating transaction linkability. We calculate the anonymity sets of these 21 transactions based on our default values of the adversary's uncertainty: a 5-minute exchange rate uncertainty and a 15-minute payment time uncertainty. The uncertainty windows are centered around the true values of the payment time and the exchange rate determination time. For price uncertainty, we use the actual list of shipping options (and resulting list of final prices) that we recorded while making purchases. The number of possible pre-BTC prices is typically 5 or fewer per purchase. We find that in 10 out of 21 cases, the anonymity set size is 1 (Figure 9).

For 17 of the 21 purchases, there was at least one tracker that received transaction-relevant information

and was also present on the payment receipt page, which means that for these adversaries the payment time uncertainty is very low. In this scenario, the anonymity set sizes are much lower: in fact, it is 1 for 14 of 17 purchases. The adversary behaving as described in Section 6 would identify the correct transaction 90% of the time. The true negative rate is also high: 82%. This means that if the adversary's assumption about the payment time was incorrect for whatever reason — if the network load is too high or the user's wallet software included too low a fee, the receipt may not happen quickly — the adversary will be able to correctly deduce this.

Validating the cluster intersection attack. Next we validate the cluster intersection attack. Out of the 11 purchases we made using mixed coins, we consider adversaries that observe a random t of those purchases and know which transaction on the blockchain corresponds to each of them. We find that for $t = 2$, the adversary described in Algorithm 2 has an 89% chance of correctly identifying our wallet cluster, and for $t = 3$, this goes up to 99%.

If more rounds of mixing are used, the attack will not work as well. On the other hand, we mixed all our coins during a 3-day period, and realistic users who mix coins over a period of months or years will have worse privacy. In Section 5 we evaluated the effect of these factors as well as the number of transactions observed by the adversary. For our experiments in this section, we limited the number of parameters because of the expensive nature of the mixing market.

Validating the attack end-to-end. The two attacks are especially powerful when combined. Even if the single-transaction linkage doesn't produce a unique result, we can run cluster intersection on every possible combination of the candidate transactions produced by it. Most combinations will produce an intersection of size zero, and can be discarded. If exactly one combination produces one cluster and all the rest produce zero clusters, then the adversary outputs that cluster.

From our purchase records, we determined that there were 11 trackers in a position to observe more than one purchase: americanexpress.com, chatid.com, criteo.com, doubleclick.net, facebook.com, google-analytics.com, google.com, monetate.net, revjet.com, steelhousemedia.com, tealiumiq.com. Overall there are 25 pairs of purchases for which there was a tracker that could observe both purchases. We ran the end-to-end attack on all 25 pairs, again using default values of the adversary's payment time uncertainty and exchange rate uncertainty. We found that the attack succeeds in identifying our wallet cluster in 20 cases.

Again, these numbers reflect conservative assumptions about the adversary's knowledge. If the Bitcoin transaction details are directly leaked to the tracker, or if the tracker is present on the receipt page, or observes more than two payments, the success rate will be much higher. Similarly, many Bitcoin users do not employ mixing. We do not know what fraction of e-commerce purchases are made with mixed coins, but we observe that only .05% of transactions on the blockchain over the past two years are CoinJoin transactions. If the user doesn't employ mixing, then it will be straightforward for the adversary to identify her wallet cluster, even with a high degree of uncertainty in the payment amount and time.

8 Mitigation and discussion

Our findings are a reminder that systems without provable (or at least well-defined) privacy properties may have information leaks and privacy breaches in unexpected ways. When multiple such systems interact, the leaks can be even more subtle. For another example of the difficulty of composing systems with complex privacy properties, see [9].

Cryptocurrency anonymity is a new research topic, but it sits at the intersection of anonymous communication and data anonymization, both well-established fields. Unfortunately, it seems to inherit the *worst* of these two worlds. Like data anonymization (and unlike anonymous communication), sensitive data must be publicly and permanently stored, available to any adversary, and de-anonymization may happen retroactively. And like anonymous communication systems (and unlike data anonymization), privacy depends on subtle interactions arising from the behavior of users and applications. Worse, realistic traces of the system may not be available at the time of designing and implementing the privacy defenses.

Turning to defenses, we observe that our first attack exploits the inherent tension between privacy and e-commerce, and our second attack exploits the inherent tension between privacy and the public nature of the blockchain. Thus, all mitigation strategies come with tradeoffs. The available mitigations break down into three categories: self-defense by users, techniques that merchants can use, and alternative cryptocurrencies or cryptocurrency-based payment methods.

Mitigation by merchants. There are a few straightforward mitigations that merchants could de-

ploy: (1) enabling HTTPS on all shopping (and especially payment-related) pages — this would protect against network adversaries, but not third-party trackers, our main adversary of interest (2) generating Bitcoin-address QR codes internally instead of outsourcing it to a third party; (3) avoiding leaks of the Bitcoin address from payment receipt pages; and (4) avoiding unintentional PII leaks. As to the last point, however, note that the attack succeeds as long as *some* first party website visited by the user leaks PII to third parties, and at least some PII leakage is for cross-device linking purposes [13], and thus intentional. Beyond these obvious steps, merchants could share less data with third parties, and with fewer of them, but this would come at the expense of their advertising and analytics objectives.

Self-defense. Web tracking is a well-known privacy threat, and the main defense is to use browser extensions such as uBlock Origin, Adblock Plus, or Ghostery to block trackers. Such defenses can be quite effective, but they are far from perfect [16, 43]. Our measurements in Section 4 confirm the partial effectiveness of these tools. Note that these tools do not help when the adversary is a network eavesdropper (for either attack) or the payment processor (for the cluster intersection attack).

On the cryptocurrency side, the main self-defense is to use improved mixing techniques, especially multi-round mixing. We showed in Section 6 that this is effective (but not perfect) as long as the adversary observes only 2 or 3 transactions. In our end-to-end evaluation in Section 7, we carried out only 1 or 2 rounds of mixing, and this a limitation of our experiments. Increasing the number of rounds comes at the expense of cost (transaction fees and mixing fees) and convenience (due to transaction confirmation time). A more through evaluation of the trade-offs is a topic for future work.

While we have focused on CoinJoin or decentralized mixing, in principle the cluster intersection attack should also work against centralized mixes. If a mixing service introduces a delay of (say) up to 6 blocks, then for a given coin that was mixed at a given block height, all mix outputs produced in the next 6 blocks can be considered part of its anonymity set. The main complication is the extent to which mix transactions are distinguishable from other transactions, which is likely highly implementation-dependent. Of course, centralized mixing is additionally vulnerable to the mix operator colluding with the adversary or stealing the funds. Evaluating the attack against centralized mixes (as well as other anonymity techniques including TumbleBit [22]) is an avenue for future work.

Alternative cryptocurrencies and payment mechanisms. Our attacks apply to Bitcoin, Litecoin, and any other cryptocurrency with Bitcoin-style transactions. However, unlike the Bitcoin approach of anonymity as an overlay, there are some cryptocurrencies that bake privacy into the protocol, and promise untraceability of transactions. The most well known of these are Zcash, based on the Zerocash protocol [7, 44], and Monero, based on the Cryptonote protocol [63]. Zcash is more computationally expensive but comes with more rigorous security properties. Of the two, Monero has more vendor support at the time of writing, but still far less than Bitcoin or even Litecoin, and primarily on hidden-service sites merchandising illicit goods. While some anonymity weaknesses have recently been revealed in Monero [34, 45], we believe that it is not susceptible to the cluster intersection attack.

The lightning network [52] is a proposal for a fast micropayments. It is a network of two-party bidirectional payment channels. If Alice wants to pay Bob, she finds a path of such channels that link her to Bob, through which she can route the payment. Although the lightning network relies on Bitcoin (or another underlying cryptocurrency) for its security, the vast majority of transactions are off-chain. There is no global ledger of all lightning payments, rendering our attack ineffective. While the lightning network would be an effective defense against our attacks, it introduces other privacy concerns [6, 38], and the issues that arise are analogous to communications anonymity [25].

Finally, like virtually all deanonymization attacks on cryptocurrencies, our techniques could be used to build forensic tools for law enforcement use. In past investigations, agents have sought to find the identity behind specific blockchain transactions that were known to represent thefts, funding of unlawful activities, or earnings from unlawful activities, as in the case of ransomware. Alternatively, agents may have an identified person of interest and may wish to scrutinize their cryptocurrency dealings for evidence of money laundering or other financial crimes. Thus, both blockchain $\rightarrow$ web and web $\rightarrow$ blockchain linking techniques are of potential interest to law enforcement. Agents might subpoena a tracker or payment processor for information that might allow such linkage, or even use network surveillance techniques.

Acknowledgements. We are grateful to Malte Möser and Ben Burgess for feedback on a draft. This work is supported by NSF awards CNS-1421689, CNS-1526353, CNS-1651938, and an NSF Graduate Research Fellowship under grant number DGE-1148900.

References

[1] Step by step running the tumbler. Github, https://github.com/JoinMarket-Org/joinmarket/wiki/Step-by-step-running-the-tumbler, 2017.

[2] Gunes Acar, Christian Eubank, Steven Englehardt, Marc Juarez, Arvind Narayanan, and Claudia Diaz. The web never forgets: Persistent tracking mechanisms in the wild. In *Proceedings of ACM Conference on Computer and Communications Security*, 2014.

[3] Elli Androulaki, Ghassan O Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. Evaluating user privacy in bitcoin. In *Financial Cryptography and Data Security*, 2013.

[4] Julia Angwin and Jennifer Valentino-Devries. Google's iphone tracking. *Wall Street Journal*, 2012.

[5] K Atlas. Weak privacy guarantees for sharedcoin mixing service, 2014.

[6] Kristov Atlas. The inevitability of privacy in lightning networks. https://www.kristovatlas.com/the-inevitability-of-privacy-in-lightning-networks/, 2017.

[7] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized Anonymous Payments from Bitcoin. In *IEEE Symposium on Security and Privacy*, 2014.

[8] Alex Biryukov, Dmitry Khovratovich, and Ivan Pustogarov. Deanonymisation of clients in bitcoin p2p network. In *Proceedings of ACM Conference on Computer and Communications Security*, 2014.

[9] Alex Biryukov and Ivan Pustogarov. Bitcoin over tor isn't a good idea. In *IEEE Symposium on Security and Privacy*, 2015.

[10] George Bissias, A Pinar Ozisik, Brian N Levine, and Marc Liberatore. Sybil-resistant mixing for bitcoin. In *Proceedings of WPES*. ACM, 2014.

[11] Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A. Kroll, and Edward W. Felten. Research perspectives and challenges for bitcoin and cryptocurrencies. In *IEEE Symposium on Security and Privacy*, 2015.

[12] Joseph Bonneau, Arvind Narayanan, Andrew Miller, Jeremy Clark, Joshua A. Kroll, and Edward W. Felten. Mixcoin: Anonymity for bitcoin with accountable mixes. In *Financial Cryptography and Data Security*. 2014.

[13] Justin Brookman, Phoebe Rouge, Aaron Alva Alva, and Christina Yeung. Cross-device tracking: Measurement and disclosures. 2018.

[14] Ceren Budak, Sharad Goel, Justin Rao, and Georgios Zervas. Understanding emerging threats to online advertising. In *Proceedings of the ACM Conference on Economics and Computation*, 2016.

[15] Peter Eckersley. How unique is your web browser? 2010.

[16] Steven Englehardt and Arvind Narayanan. Online tracking: A 1-million-site measurement and analysis. In *Proceedings of ACM Conference on Computer and Communications Security*, 2016.

[17] Steven Englehardt, Dillon Reisman, Christian Eubank, Peter Zimmerman, Jonathan Mayer, Arvind Narayanan, and Edward W Felten. Cookies that give you away: The surveillance implications of web tracking. In *Proceedings of the Conference on World Wide Web*, 2015.

[18] Giulia Fanti and Pramod Viswanath. Anonymity properties of the bitcoin p2p network. *arXiv preprint arXiv:1703.08761*, 2017.

[19] Arthur Gervais, Alexandros Filios, Vincent Lenders, and Srdjan Capkun. Quantifying web adblocker privacy. *IACR Cryptology ePrint Archive*, 2016.

[20] Arthur Gervais, Hubert Ritzdorf, Mario Lucic, and Srdjan Capkun. Quantifying location privacy leakage from transaction prices. *ESORICS*, 2016.

[21] Google. Google charts faq. https://developers.google.com/chart/interactive/faq.

[22] Ethan Heilman, Leen Alshenibr, Foteini Baldimtsi, Alessandra Scafuro, and Sharon Goldberg. Tumblebit: An untrusted bitcoin-compatible anonymous payment hub. NDSS, 2016.

[23] Ethan Heilman, Foteini Baldimtsi, and Sharon Goldberg. Blindly signed contracts: Anonymous on-blockchain and off-blockchain bitcoin transactions. In *Financial Cryptography Workshops*, 2016.

[24] Matthias Hellwig and Alexander Souza. Approximation algorithms for generalized and variable-sized bin covering. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 194–205. 2012.

[25] Jordi Herrera-Joancomartí and Cristina Pérez-Solà. Privacy in bitcoin transactions: new challenges from blockchain scalability solutions. In *Modeling Decisions for Artificial Intelligence*, pages 26–44. Springer, 2016.

[26] Kashmir Hill and Surya Mattu. Before You Hit 'Submit,' This Company Has Already Logged Your Personal Data. https://gizmodo.com/before-you-hit-submit-this-company-has-already-logge-1795906081, 2017.

[27] Péter L Juhász, József Stéger, Dániel Kondor, and Gábor Vattay. A bayesian approach to identify bitcoin users. *arXiv preprint arXiv:1612.06747*, 2016.

[28] Harry Kalodner, Steven Goldfeder, Alishah Chator, Malte Möser, and Arvind Narayanan. Blocksci: Design and applications of a blockchain analysis platform. *arXiv preprint arXiv:1709.02489*, 2017.

[29] John Koetsier. 90% of marketers say retargeting now as good as search ads, email marketing. VentureBeat, https://venturebeat.com/2014/12/16/90-of-marketers-say-retargeting-now-as-good-as-search-ads-email-marketing/, 2014.

[30] Philip Koshy, Diana Koshy, and Patrick D. McDaniel. An analysis of anonymity in bitcoin using P2P network traffic. In *Financial Cryptography and Data Security*.

[31] Balachander Krishnamurthy, Konstantin Naryshkin, and Craig Wills. Privacy leakage vs. protection measures: the growing disconnect. In *W2SP*, 2011.

[32] Balachander Krishnamurthy and Craig E Wills. On the leakage of personally identifiable information via online social networks. In *Proceedings of the ACM workshop on Online social networks*, 2009.

[33] Katharina Krombholz, Aljosha Judmayer, Matthias Gusenbauer, and Edgar Weippl. The other side of the coin: User experiences with bitcoin security and privacy. In *Financial Cryptography and Data Security*, 2016.

[34] Amrit Kumar, Clément Fischer, Shruti Tople, and Prateek Saxena. A traceability analysis of monero's blockchain. *IACR Cryptology ePrint Archive*, 2017.

[35] Pierre Laperdrix, Walter Rudametkin, and Benoit Baudry. Beauty and the beast: Diverting modern web browsers to build unique browser fingerprints. In *IEEE Symposium on Security and Privacy*, 2016.

[36] Adam Lerner, Anna Kornfeld Simpson, Tadayoshi Kohno, and Franziska Roesner. Internet jones and the raiders of the lost trackers: An archaeological study of web tracking from 1996 to 2016. In *Proceedings of the USENIX Security Symposium*, 2016.

[37] Timothy Libert. Exposing the invisible web: An analysis of third-party http requests on 1 million websites. *International Journal of Communication*, 9:18, 2015.

[38] Giulio Malavolta, Pedro Moreno-Sanchez, Aniket Kate, Matteo Maffei, and Srivatsan Ravi. Concurrency and privacy with payment-channel networks. 2017.

[39] Gregory Maxwell. CoinJoin: Bitcoin Privacy for the Real World, 2013.

[40] Jonathan R Mayer and John C Mitchell. Third-party web tracking: Policy and technology. In *IEEE Symposium on Security and Privacy*, 2012.

[41] Sarah Meiklejohn and Claudio Orlandi. Privacy-enhancing overlays in bitcoin. In *Financial Cryptography and Data Security*, 2015.

[42] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M Voelker, and Stefan Savage. A fistful of bitcoins: Characterizing payments among men with no names. In *Proceedings of ACM IMC*, 2013.

[43] Georg Merzdovnik, Markus Huber, Damjan Buhov, Nick Nikiforakis, Sebastian Neuner, Martin Schmiedecker, and Edgar Weippl. Block me if you can: A large-scale study of tracker-blocking tools. In *IEEE Symposium on Security and Privacy*, 2017.

[44] Ian Miers, Christina Garman, Matthew Green, and Aviel D Rubin. Zerocoin: Anonymous distributed e-cash from bitcoin. In *IEEE Symposium on Security and Privacy*, 2013.

[45] Andrew Miller, Malte Möser, Kevin Lee, and Arvind Narayanan. An empirical analysis of linkability in the monero blockchain. *Proceedings on Privacy Enhancing Technologies*, 2018.

[46] Malte Möser and Rainer Böhme. Anonymous alone? measuring bitcoin's second-generation anonymization techniques.

[47] Malte Möser and Rainer Böhme. Join me on a market for anonymity. In *Proceedings of WPES*. ACM, 2016.

[48] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System, 2008.

[49] Till Neudecker and Hannes Hartenstein. Could network information facilitate address clustering in bitcoin?

[50] Nick Nikiforakis, Luca Invernizzi, Alexandros Kapravelos, Steven Van Acker, Wouter Joosen, Christopher Kruegel, Frank Piessens, and Giovanni Vigna. You are what you include: large-scale evaluation of remote javascript inclusions. In *Proceedings of ACM Conference on Computer and Communications Security*, 2012.

[51] Esteban Ordano. We need more coinjoin. https://medium.com/@eordano/we-need-more-coinjoin-c7fefd12dc5e, 2015.

[52] Joseph Poon and Thaddeus Dryja. The bitcoin lightning network: Scalable off-chain instant payments. 2015.

[53] Rebecca S. Portnoff, Danny Yuxing Huang, Periwinkle Doerfler, Sadia Afroz, and Damon McCoy. Backpage and bitcoin: Uncovering human traickers. In *Proceedings of the Conference on Knowledge Discovery and Data Mining*, 2017.

[54] r/Bitcoin. r/Bitcoin Bitcoin Websites. Reddit, https://docs.google.com/document/d/1pFHJ34pZ_5Umfmlk_eAIcBSscSAA-3xd6qWYzeEYhec/edit, 2017.

[55] Fergal Reid and Martin Harrigan. An analysis of anonymity in the bitcoin system. In *Security and Privacy in Social Networks*, pages 197–223. Springer, 2013.

[56] Franziska Roesner, Tadayoshi Kohno, and David Wetherall. Detecting and defending against third-party tracking on the web. In *USENIX Symposium on Networked Systems Design and Implementation*, 2012.

[57] Dorit Ron and Adi Shamir. Quantitative analysis of the full bitcoin transaction graph. In *Financial Cryptography and Data Security*, 2013.

[58] Tim Ruffing, Pedro Moreno-Sanchez, and Aniket Kate. Coinshuffle: Practical decentralized coin mixing for bitcoin. *ESORICS*, 2014.

[59] Jan-Willem Selij. Coinshuffle anonymity in the block chain. 2015.

[60] Ashkan Soltani, Shannon Canty, Quentin Mayo, Lauren Thomas, and Chris Jay Hoofnagle. Flash cookies and privacy. In *AAAI Spring Symposium: Intelligent Information Privacy Management*, 2010.

[61] Aditya K Sood and Richard J Enbody. Malvertising–exploiting web advertising. *Computer Fraud & Security*, 2011.

[62] Spendabit.co. Merchants Accepting Bitcoin - Spendabit. https://spendabit.co/merchants, 2017.

[63] Nicolas van Saberhagen. Cryptonote v2.0. https://cryptonote.org/whitepaper.pdf, 2013.

[64] Shaileshh Bojja Venkatakrishnan, Giulia Fanti, and Pramod Viswanath. Dandelion: Redesigning the bitcoin network for anonymity. *arXiv preprint arXiv:1701.04439*, 2017.

[65] Danny Yang, Jack Gavigan, and Zooko Wilcox-O'Hearn. Survey of confidentiality and privacy preserving technologies for blockchains. https://z.cash/static/R3_Confidentiality_and_Privacy_Report.pdf, 2016.

[66] Zhonghao Yu, Sam Macbeth, Konark Modi, and Josep M Pujol. Tracking the trackers. In *Proceedings of the Conference on World Wide Web*, 2016.

A Appendix

A.1 JoinMarket identification

CoinJoin transactions have a distinct structure, and JoinMarket transactions especially so. Whether or not JoinMarket can be modified to operate in a way that the transactions are not as distinguishable from other transactions is an open question. Here we describe our algorithm for identifying JoinMarket transactions, and evaluate its effectiveness. We adapt the algorithms from several previous works [5, 41, 46, 47, 59].

Algorithm 3: JoinMarket Identification.

```
1:  function IsJoinMarketTransaction(tx)
2:      if ContainsOpReturn(tx) then
3:          return false
4:      p ← ⌈|Outs(tx)|/2⌉
5:      if p < 2 then
6:          return false
7:      v ← MostCommon(Val(O) : O ∈ Outs(tx))
8:      if |{O ∈ Outs(tx) | Val(O) = v}| ≠ p then
9:          return false
10:     A ← {Addr(x) : x ∈ Ins(tx)}
11:     V ← {}
12:     for all a in A do
13:         s ← 0
14:         for all I in Ins(tx) do
15:             if Addr(I) = a then
16:                 s ← s + Val(I)
17:         V ← V ∪ {s}
18:     q ← MaxFee(v)
19:     B ← Array of length p with all entries v − q
20:     i ← 0
21:     for all O in Outs(tx) do
22:         if Val(O)! = v then
23:             B[i] ← B[i] + Val(O)
24:             i ← i + 1
25:     for all P in Partitions(V) do
26:         t ← 0
27:         for all S_i in P do
28:             if ∑ S_i ≥ B_i then
29:                 t ← t + 1
30:         if t ≥ p then
31:             return True
32:     return False
```

The algorithm is a series of heuristics to filter transactions based on the following observations:

- Transactions should only contain spendable addresses (lines 2–3)
- There must be at least two participants. If there are n participants, there could be either $2n$ or $2n-1$ outputs because JoinMarket has "sweep transactions" where the taker obtains no change (lines 4–6)
- There must be an output of value v for each participant, v being the most common output value (lines 7–9)
- There must be enough inputs to cover all of the outputs (lines 10–32). Specifically, for each change address, there must be a distinct set of inputs that add up to at least the output value v plus the change value minus the max fee (q) that might have been paid to the liquidity providers. For our calculations we set this to be the maximum of .0001 satoshis or 1% of the CoinJoin output.

One limitation of this algorithm is that it is slow when the number of inputs is large. This is unavoidable as the problem is NP-complete (variable-sized bin-covering in the unit supply model [24]). The listing shows a brute-force implementation for simplicity; our actual implementation is optimized, but nevertheless exponential. For our analyses, we ran it on transactions with at most 17 inputs; it takes about 30 minutes to process 150 million transactions. Based on the work of Möser et al., who don't use this heuristic, the vast majority — 92% — of JoinMarket transactions have no more than 17 inputs.

When we run this algorithm on our two-year period of interest (May 2015 – April 2017; block 354416–block 464269), it results in 78,697 transactions. The algorithm has low false negatives, and thus we regard this as a near-superset of JoinMarket transactions for this period. The criteria used in Algorithm 3 for filtering transactions are necessarily true of all JoinMarket transactions, except for any transactions where liquidity providers charged so high a fee that they were rejected by our max-fee heuristic. But based on the empirical analysis of [47], virtually all offers posted on the market by makers have a fee that is significantly less than the threshold we used. As a further sanity check, all CoinJoins that we performed in our experiments (Section 7) are identified by this algorithm.

CoinJoins and especially JoinMarket transactions tend to connect to each other, and thus we can expect to find large connected components among the identified transactions. Indeed, among the 78,697 transactions, we find a single giant component of size 60,187. We regard these as a near-subset of CoinJoin transactions during this period. While it is possible that non-CoinJoin transactions may sometimes accidentally satisfy the criteria in Algorithm 3, it is unlikely that they will cluster with the JoinMarket transactions.

The fact that our near-superset and our near-subset are similar in size gives us further confidence in the method. Depending on the application, one or the other version may be more suitable. We make use of both versions in our analyses: when simulating the victim, we use the near-subset, because we want to have high confidence that the transactions we use for simulation are indeed CoinJoins. When simulating the adversary, we

use the near-superset version because the cluster intersection attack is more robust to false positives than false negatives.

A.2 Additional tables, figures, and lists

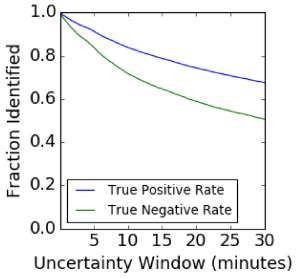

Fig. 10. Effect of payment time uncertainty on success rate when prices are sampled from the blockchain instead of from merchant websites. Compare to Figure 5.

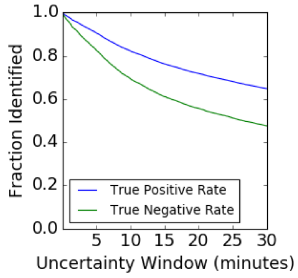

Fig. 11. Effect of payment time uncertainty on success rate when prices are doubled. Compare to Figure 5.

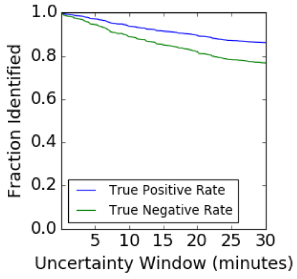

Fig. 12. Effect of payment time uncertainty on success rate when performed on Litecoin. Compare to Figure 5.

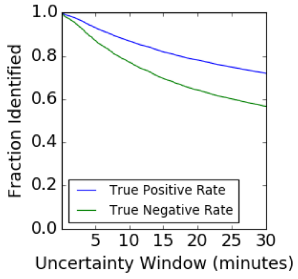

Fig. 13. Effect of payment time uncertainty on success rate against Bitpay transactions. Compare to Figure 5.

PII type	w/o protection	w/ protection
username	76	39
email	63	30
firstname	41	19
lastname	29	17
address	15	9
phone	13	7
name	7	4
company	5	3
Third parties receiving PII	137	70

Table 6. Number of third parties receiving each type of PII, with or without tracking protection

Merchant	Receipt page third parties	Receipt page TPs w/ tx-relevant info
adafruit.com	13	5
baronfig.com	35	7
digitalrev.com	20	9
fancy.com	0	0
giftoff.com	10	1
givemethedirt.com	33	5
healthmonthly.co.uk	19	1
jenshansen.com	59	7
newegg.com	46	22
opendime.com	5	5
overstock.com	42	42
petspyjamas.com	36	11
pi-supply.com	0	0
readytogosurvival.com	32	3
reddit.com	1	1
reeds.com	0	0
somethinggeeky.com	0	0
thepihut.com	44	8
thisisground.com	50	4
tightstore.com	32	4
Third parties on receipt pages	245	88

Table 7. Number of third parties on each merchant's payment receipt page, and the number of those third parties that also received transaction-relevant information

Third-party domain	w/o protection	w/ protection
google-analytics.com	27	0
facebook.com	16	0
doubleclick.net	8	0
google.com	8	8
chart.googleapis.com	8	8
segment.io	3	0
steelhousemedia.com	3	0
criteo.com	3	0
blockchain.info	3	3
hits.io	2	0
monetate.net	2	0

Table 8. Prevalence of third parties receiving transaction-relevant information on at least two websites, with or without tracking protection

List of third parties that receive transaction-relevant information despite the use of tracking protection: monkeykingcode.com, chatid.com, blockchain.info, google.com, revjet.com, qrserver.com, bootstrapcdn.com, americanexpress.com, schutzklick.de, chart.googleapis.com, nosto.com, gopollen.com, exponea.com