

# Monero Traceability Heuristics: Wallet Application Bugs and the Mordinal-P2Pool Perspective

Anonymous Authors  
Anonymous Affiliation  
Anonymous Location  
authors@example.com

**Abstract**—Privacy-focused cryptoassets like Monero are intentionally difficult to trace. Over the years, several traceability heuristics have been proposed, most of which have been rendered ineffective with subsequent protocol upgrades. Between 2020 and 2023, Monero wallet application bugs “Differ By One” and “10 Block Decoy Bug” have been observed and identified and discussed in the Monero community. In addition a decentralized mining pool named P2Pool has proliferated, and a controversial UTXO NFT imitation has been adapted for Monero, known as Mordinals. In this paper, we systematically describe the traceability heuristics that have emerged from these phenomena, and evaluate their quality based on ground truth, and through pairwise comparisons. We also explore a the temporal perspective, and show which of these heuristics have been applicable over the past years, what fraction of decoys could be eliminated and what the effective ring size is.

Our findings illustrate that most of the heuristics have a high precision, that the “10 Block Decoy Bug” and the Coinbase decoy identification heuristics have had the most impact between 2019 and 2023, and that the former could be used to evaluate future heuristics, assuming they are applicable during that time frame.

We discuss our results in detail, and highlight the need for caution when applying these heuristics, as there is very limited ground truth data available.

## I. INTRODUCTION

In the evolving landscape of digital currencies, Monero has emerged as a significant player, renowned for its strong emphasis on privacy. Originating from the CryptoNote protocol [1], Monero represents a key advancement in blockchain technology by employing stealth addresses, hiding transaction amounts using Ring Confidential Transactions (RingCT) [2] and obfuscating its underlying transaction graph with ring signatures.

As of November 2023, with a market capitalization of approximately 3 billion USD, Monero stands as the foremost privacy-focused cryptocurrency. This prominence can largely be attributed to its enduring commitment to privacy-by-default features, which notably surpass those offered by more mainstream blockchains such as Bitcoin and Ethereum. Unlike these transparent blockchains, where transactions are prone to traceability through heuristic methods that exploit protocol specifics or typical user behaviors [3]–[5], Monero has consistently evolved to address its vulnerabilities.

However, despite its advancements, Monero has not been impervious to challenges. Over the years, various Monero-specific heuristics have been employed to probe its privacy

features, leading to the identification of early protocol weaknesses [6]–[9]. These vulnerabilities, though, have been largely mitigated through a combination of protocol upgrades, improvements in wallet implementation, and ongoing community education efforts.

Yet, new concerns have arisen. Recent discoveries of software bugs and the proposition of projects (likely unintentionally) misaligned with Monero’s privacy efforts, highlight ongoing threats to the currency’s privacy framework. These issues have been disclosed and discussed in the community, but in this paper, we aim to formalize and assess the impact of these developments by studying the new heuristics that are applicable as a consequence.

Our contributions in this study are manifold. We delve into the specifics of the ‘10 Block Decoy Bug’, ‘Differ-by-one’, Coinbase and Mordinal decoy identification heuristics, as well as a P2Pool specific output merging heuristic based on publicly known Miner payouts. Several of these heuristics have previously been discussed in the Monero community, through Github issues and discussions on Reddit. However, we provide a systematic, comparative and combined evaluation, which constitutes the main contribution of this work. We present a comprehensive analysis, measuring the impact of each heuristic’s applicability over time, as well as the effective ring size of Monero’s transaction inputs up until the October of 2023. This combined approach offers a nuanced understanding of the current state of privacy in the Monero ecosystem.

## II. BACKGROUND AND RELATED WORK

Monero is based on the cryptonote protocol [1]. Its three most prominent privacy enhancing technologies of Monero are ring signatures, stealth addresses and RingCT [2]. In contrast to a transparent Unspent Transaction Output (UTXO) blockchain design like in Bitcoin [10], each transaction input is a ring signature, where only one ring member is the truly spent input, and the others are decoys. As of 2023, the ring size in Monero is 16, meaning 15 ring members are decoys. Stealth addresses are one-time addresses generated by the sender of a transaction, derived from a supplied public address. RingCT hides the transaction amounts using bulletproofs [11].

**Definitions.** An input ring  $R = \{pk_1, \dots, pk_n\}$  consists of a set of referenced transaction outputs, identified by their public key  $pk$ . Only one  $pk \in R$  is the truly spent output, also known as the *true-spend*. An input ring of a monero transaction is said

to be fully traceable, if the ring member that was truly spent is identified, and thereby all other ring members are marked as decoys. An input's ring *effective ring size* can be reduced from the original count of ring members, if some members can be identified as decoys. This does not necessarily reveal the true-spend, but reduces the anonymity set.

Early works on traceability heuristics for Monero have primarily focused on true-spend identification heuristics. In the two earliest and well known analyses by Möser et al. [7] and Kumar et al. [6] traceability heuristics exploiting zero-mixins, chain reaction (also known as cascade), output-merging and guess newest have been explored. The zero-mixin and chain reaction heuristics have exploited early weaknesses in the Monero protocol, and yield results that can be treated as ground truth, as they do not produce false positives. Subsequently, Hinteregger and Haslhofer [8] have proposed the intersection removal heuristic, exploiting the observability of the same key image associated with a spent output on forked Monero blockchains with a shared history. Here, the true spend may be identified in the intersection of the input rings. They have also shown that the Guess Newest Heuristic no longer works due to an upgrade of Monero's decoys selection algorithm. At the same time, Yu et al. [9] have proposed the closed set heuristic, which does not directly identify the ring in which a certain output has been spent, but can make a claim that the output must have been spent at a given point in time. It is therefore primarily a decoy identification heuristic.

Wijaya et al. have proposed a restricted version of the output merging heuristic applied to mining pool outputs [12], but did not evaluate with ground truth [12]. They also studied the unforkability of Monero [13], as it allows for the intersection removal heuristic.

A recent work by Vijayakumaran [14] showcased how the Dulmage-Mendelsohn Decomposition [15] can be used to infer the true spends of closed sets in addition to the zero-mixin and cascade attacks, providing a polynomial-time implementation of the closed set heuristic.

Assuming a probabilistic perspective, Aeeneh et al. [16] have proposed two methods to assess the probability of a ring member being the true spend, under the circumstances that a given ring member is only seen once in exactly one ring, and when the ring member age distribution does not correspond with the expected true spend age distribution. The authors did not empirically evaluate their work.

### III. DATA

For our analysis, we've used three types of datasets.

- Monero mainnet blockchain transactions from the blockchain's inception on 2014-04-18 until 2023-10-31.
- Manual Monero testnet transactions conducted by ACK-J [17] between 2022-01-20 and 2022-02-23 consisting of 760,588 transactions and 1,334,693 key images for which the true spend is known.
- 31,759 p2pool mining pool payout transactions with 2,298,927 individual payouts.

### IV. ANALYSIS

In this section, we define and empirically cross-evaluate 6 heuristics. We begin by defining the notation used. To recall,  $TX$  denotes the set of all transactions stored on the blockchain. Each transaction  $tx \in TX$  creates outputs, identified by a public key  $pk$ , which is owned by a Monero user controlling a set of public and private keys. A transaction input is an input ring  $R = \{pk_1, \dots, pk_n\}$ , with one  $pk_t \in R$  as the true spend, and the other ring members are decoys. As output  $pk$  are used as inputs, one can refer to a  $pk$  as an *enote* in the general case.

We categorize Monero traceability heuristics as follows:

- **True-Spend Identification Heuristic:** For an input ring  $R_1$ , identifies the likely true spend  $pk_t$  in  $R_1 = \{pk_1, \dots, pk_n\}$ . All other  $pk \in R_1$  where  $pk \neq pk_t$  are decoys. If  $pk_t$  appears in any other input ring  $R_x$ , it is marked as a decoy. This is based on the unique expendability of an enote in a single input ring. In summary, heuristics of this category identify true-spends and, as a consequence, decoys, which may also appear at a future point in time.
- **Decoy Identification Heuristic:** For an input ring  $R_1$  determines a subset  $D_1 = \{pk_{d1}, \dots, pk_{dm}\} \subseteq R_1$  of public keys as decoys within  $R_1$ . The subset  $D$  can contain multiple enotes, based on specific criteria or be empty. In summary, heuristics of this category only identify decoys within input rings it is applied to.

We evaluate each heuristic against the well known zero-mixin and chain reaction heuristics results (also known as zero-decoy and cascade). While these heuristics have been largely ineffective since 2018 [8], they still constitute one of the main set of labels that can be treated as ground truth, as it has no false positives. We will also evaluate most heuristics with the 10 block decoy bug heuristic that we will introduce in the first subsection IV-A, as it is a high confidence heuristic with results until 2023.

On the basis of labeled ring members, we measure true positives (TP), false positives (FP) and precision  $P$ , with

$$P = \frac{TP}{TP + FP}$$

For decoy identification heuristics, we also measure the self collision rate  $SCR$ . A collision occurs for example when heuristic  $H_1$  claims that the same public key appearing in two different input rings ( $pk_x \in R_1, R_2$ ), is the true spend. It can only be the true spend in one of them. A collision also occurs if a  $H_1$  considers two ring members ( $pk_x \in R_1, pk_y \in R_1$ ) to be the true spend of a single input ring  $R_1$ . There can only be one true spend per ring. If  $C$  is the number of conflicting labels heuristic  $H_1$  produces, and the total number of ring members labeled by  $H_1$  is  $N$ , we calculate the self collision rate  $SCR$  with

$$SCR = \frac{C}{N}$$

Age (blocks):	10	56	61	85	130	332	564	758	1,087	1,199	1,478	1,855	4,143	4,832	10,438	18,869
Ring Member:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Fig. 1. Illustration of the 10-Block-Old Decoy Bug Heuristic: if there exists exactly one ring member that is 10 blocks old and the input ring has been created between October 11, 2018 and April 10, 2023, it is very likely the true spend (highlighted in green).

### A. 10 Block Decoy Bug

On May 23, 2023, a significant vulnerability was disclosed within a wallet library, specifically in the ‘wallet2’ component, as documented in the Monero project’s GitHub issue tracker<sup>1</sup>. This bug, affecting the decoy selection algorithm, was identified in a library that forms the backbone of several widely-used Monero wallet applications, including Monero Wallet GUI/CLI, Feather Wallet, Cake Wallet, and Monerujo<sup>2</sup>. The flaw was an off-by-one error in the decoy selection process, leading to an inability to select decoys that were exactly 10 blocks old. This specific age of 10 blocks is particularly noteworthy because it represents the unlock time for outputs in Monero, marking the earliest point at which a user is able to spend a received output.

This vulnerability was present across multiple versions of the wallet, from version v0.13.0.0 to v0.18.2.1. This period spans approximately 4.5 years, from October 11, 2018, to April 10, 2023, highlighting the extended duration over which this critical issue went undetected in the Monero ecosystem. As a consequence of the vulnerability, we can define a time constrained heuristic that we refer to as the *10 Block Decoy Bug Heuristic*, which is a true-spend identification heuristic that we define as follows:

**Heuristic Definition.** Define  $age(pk, R)$  as a function that determines the age of a ring member  $pk \in R$  in terms of blocks. If there exists exactly one  $pk$  in  $R$  such that  $age(pk, R) = 10$ , then this  $pk$  is highly likely to be the true spend for input rings created between October 11, 2018 and April 10, 2023. An example is visualized in Figure 1.

**Evaluation.** To validate the heuristic, we take two approaches:

- 1) We evaluate the heuristic on the Monero Mainnet against ground truth obtained from the zero mix-in and chain reaction heuristics. The heuristic identifies 1,364,175 spent outputs and consequently 27,308,717 decoys. In comparison with the zero mix-in + chain reaction heuristic, there is an overlap of 3 correctly identified true spends, and a total of 209 true positives in terms of ring members labeled correctly. There are no false positives, leading to a precision of 100%. However, there are self collisions for 6,448 ring members, as 3,112 distinct outputs are identified as the true spend of multiple input

rings. This amounts to a self collision rate  $SCR$  of 0.023%, which is likely due to a larger client diversity. This means the actual precision of this heuristic likely isn’t this high.

- 2) To provide another data point on the precision of the heuristic, we also evaluate on the Monero Testnet, where a large dataset of ground-truth transactions exist. There are 1,334,693 input rings, of which 92,954 have exactly one 10-block old ring member. In all but two input rings, the true spend is the 10 block old ring member, yielding a precision of  $\approx 99.998\%$ .

From the point of the limited ground truth data, the 10 block decoy bug heuristic appears to be highly accurate. Nevertheless, it yields a large, high confidence label set with results between 2018 and 2023, and can be used to assess the quality of other heuristics that are applicable during that time frame. We further discuss these results in Section V and proceed with the Differ By One heuristic in the next subsection.

### B. Differ By One

The Differ-by-One heuristic identifies true spends from pairs of input rings characterized by an identical set of ring members, with the exception of a single element. It was previously referenced in a GitHub repository by Kraviec-Thawyer [18] at the end of 2022, who also provided an implementation to find such instances.

In this heuristic, the distinct ring member is presumed to represent the true spend for that particular input ring. A possible explanation for this pattern is the potential caching of ring members by certain wallet applications, leading to repetitive usage of the same decoys. As this is a true-spend identification heuristic, if the  $pk \in R$  identified as the true spend appears in other input rings, it is consequently treated as a decoy in those rings. The Differ-by-One pattern has been observed across 58,429 transactions ranging from 2014-04-21 until the end of our dataset on 2023-10-31.

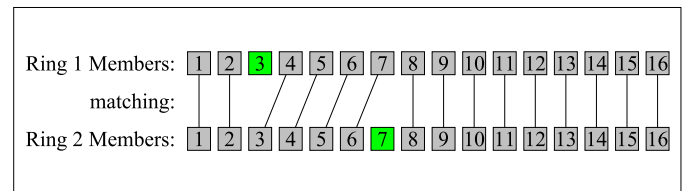


Fig. 2. Illustration of the Differ-by-One heuristic: given two input rings that are almost identical except for one ring member (i.e. all other ring members match between the rings), the differing outputs (marked in green) are likely the true spends.

**Heuristic Definition.** If two input rings  $R_1$  and  $R_2$  are almost identical but differ by exactly one ring member, that unique member is hypothesized as the true spend. See Figure 2 for a graphical illustration.

**Evaluation.** The heuristic tags 360,102 ring members as true spent outputs, and a total of 4,777,246 labeled ring members. For 178 labeled ring members there exists a self collision, as

<sup>1</sup><https://github.com/monero-project/monero/issues/8872>

<sup>2</sup><https://github.com/monero-project/research-lab/issues/99>

### C. Mordinals

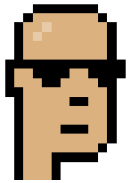


Fig. 3. Mordinal: An image embedded in the tx\_extra field in transaction hash baa3f1fa73942366c19471aac73b78dd2664eefe634bdbc260d58d09d2a0e259

The burned outputs are non-spendable and therefore obviously decoys if included in any input ring. Mordinal outputs in turn are spendable, but are anticipated to be spent in Mordinal transfer transactions, they can therefore with high likelihood be discarded as decoys when referenced by regular non-Mordinal transactions.

**Evaluation.** Between 2023-03-09 and 2023-04-21, a total of 43,099 Mordinals were minted. Subsequent to this period, transfer transactions have been sparse, and there have been no instances of minting. The heuristic marks 474,442 ring

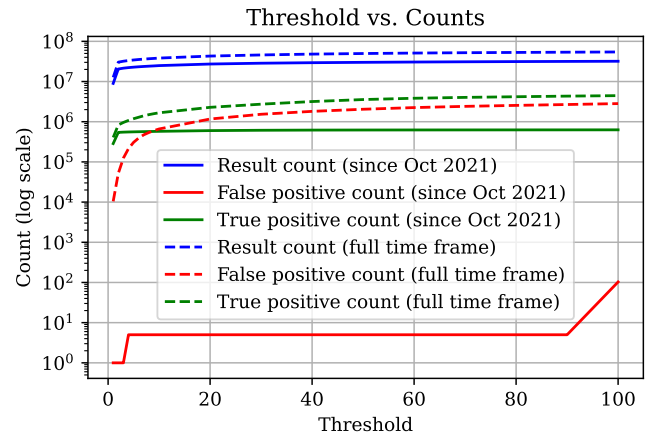


Fig. 4. If we apply the heuristic to all transactions, both the number of false positives and true positives would be high. However, if we only consider transactions that have happened since P2Pool was launched in October 2021, the number of true positives remains high, and the numbers of false positives becomes low for transactions that have  $\leq 90$  inputs.

members as decoys, of which 9,934 are true positives, and 21 are false positives, when comparing against the heuristics zero mixins, chain reaction and 10 block decoy bug. Overall, the heuristic’s precision is therefore 99.79%.

#### D. Coinbase Outputs

A coinbase transaction is a transaction where a reward is being given out to miners. There are three types of recipients for coinbase transactions; solo miners, centralized mining pools, and decentralized mining pools, with the latter prominently represented by a service called P2Pool. Unless a solo miner is equipped with significant hardware resources to generate a high mining hashrate, it is most efficient for most miners to join a mining pool in which they are rewarded proportionally to their contributed hashrate for finding a block.

Coinbase transactions associated with centralized pools usually have one output, while those associated with P2Pool usually have a relatively high number of outputs because multiple miners receive payouts in the same transaction. As a consequence, most coinbase outputs are either spent by centralized pools to send payouts to their miners, or spent directly by miners using a decentralized mining pool. This means P2Pool is responsible for most coinbase outputs. Figure 5 shows a spike in the number of coinbase outputs in October 2021 due to the launch of P2Pool. However, the average number of coinbase outputs decreased in March 2023 when P2Pool launched an upgrade that makes their payouts more efficient.

Normal Monero users are not expected to spend coinbase outputs but the decoy selection algorithm can include them in input rings. As miners tend to receive mining outputs on a recurring basis, they often need to merge their outputs. It is therefore common for coinbase outputs to be spent in transactions that have a relatively high number of inputs. This means that for most small transactions, we should be able

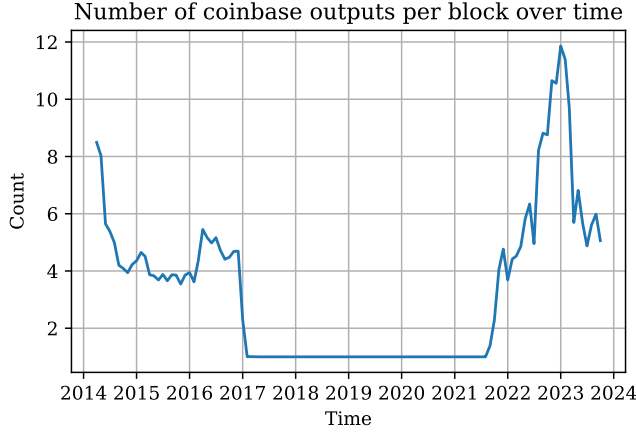


Fig. 5. The number of coinbase outputs was high until 2017, as Monero used to generate outputs of multiple denominations prior to the introduction of RingCT, hiding amounts. In 2021, the number of outputs started increasing again with the emergence of decentralized mining pool P2Pool.

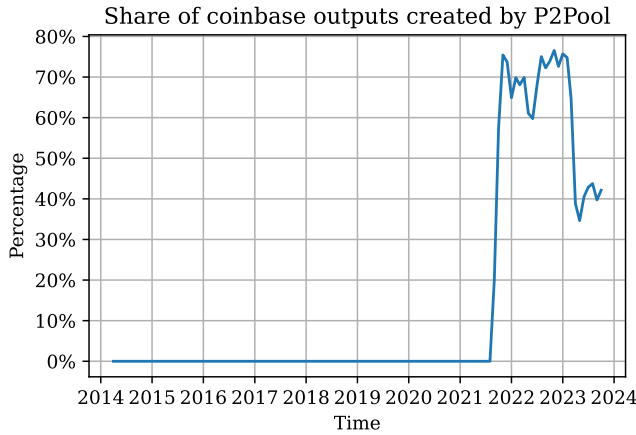


Fig. 6. The share of coinbase outputs created by P2Pool in comparison to all generated coinbase outputs spiked to more than 70% at the end of 2021, and still accounts for about 40% near the end of 2023.

to discard referenced coinbase outputs as decoys. However, it is possible for miners to merge their outputs in small transactions, so we experiment with multiple thresholds for the maximum number of inputs of transactions for which we can discard referenced coinbase outputs.

Figure 4 shows how the number of excluded decoys, false positives, and true positives change as the threshold increases. If we apply the heuristic to all transactions, both the number of false positives and true positives would be high. However, if we only consider transactions that have happened since P2Pool was launched in October 2021, the number of true positives remains high, and the numbers of false positives becomes low for all thresholds that are  $\leq 90$ . Therefore, we apply to all transactions that have happened since October 2021, for which

the number of inputs is  $\leq 90$ .

**Heuristic Definition.** Define  $C$  as the set of all coinbase transactions. The heuristic labels a ring member  $pk$  in the input ring  $R$  of any transaction  $tx$  as a decoy if and only if the number of ring inputs of  $tx$  is  $\leq 90$ , the date of  $tx$  is  $\geq 2021-10-01$  and  $pk$  is an output of a coinbase transaction  $tx_c \in C$ .

**Evaluation.** The heuristic marks 31,650,837 ring members as decoys, of which 626,679 are true positives, and 5 are false positives, when comparing against the heuristics zero mixins, chain reaction and 10 block decoy bug. The heuristic's precision is therefore 99.9%.

#### E. P2Pool Output Merging

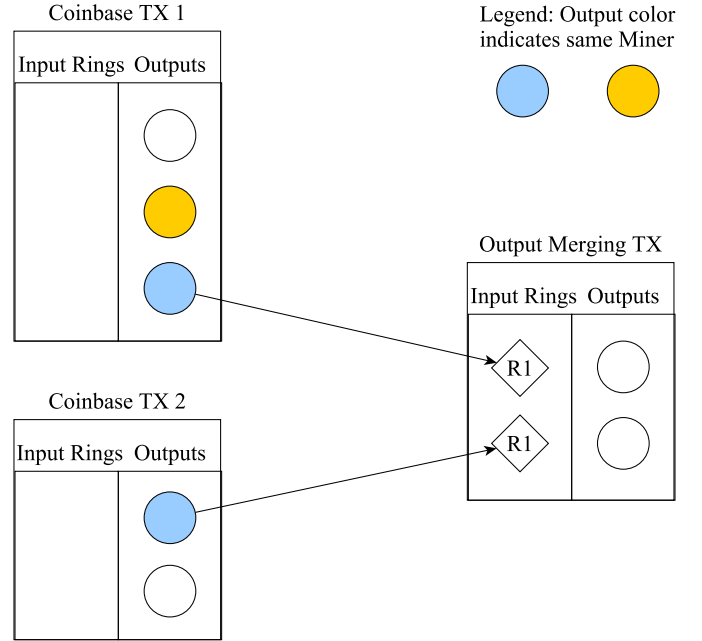


Fig. 7. P2Pool coinbase output merging: P2Pool provides information on which output is owned by which miner (indicated by the colors here) in their coinbase transactions (which have no inputs). Therefore, the output merging heuristic can be applied on the outputs of the same miner.

Many mining pools post their transactions online, often including the associated addresses of the miners. Transactions that reference multiple outputs from multiple known mining pool transactions are likely generated either by a miner consolidating their payouts or by a mining pool merging its change outputs for subsequent payouts. It is therefore reasonable to discard those outputs as decoys from the ring signatures of other transactions. Since P2Pool owns a significant share of coinbase outputs (see Figure 6) and they publish all transaction hashes and miner addresses, we apply this heuristic to their outputs. However, the analysis can be extended to centralized mining pools as well [19]. We consider transactions where all rings reference outputs owned by the same miner. In some cases, a single output is referenced by multiple transactions. For those outputs, we pick the transaction that has the highest number of inputs.



TABLE I

COMPARING HEURISTICS: PAIRWISE COLLISION AND AGREEMENT RATES BETWEEN THE HEURISTICS. COLLISION RATES ARE MOSTLY VERY LOW WITH THE EXCEPTION OF DIFFER BY ONE WHICH HAS A HIGHER COLLISION RATE WITH ZERO MIXINS + CHAIN REACTION OF 10%, AND COINBASE WHICH HAS A COLLISION RATE OF 10.18% WITH P2POOL OUTPUT MERGING. AS FOR AGREEMENTS, THERE IS MOSTLY LITTLE OVERLAP, INDICATING THAT MOST HEURISTICS ARE COMPLEMENTARY TO EACH OTHER. THE HIGHEST AGREEMENT EXISTS BETWEEN P2POOL OUTPUT MERGING AND THE COINBASE DECOY IDENTIFICATION HEURISTIC, WHICH IS PLAUSIBLE, AS THE OUTPUT MERGING LEADS TO COINBASE OUTPUTS BEING MARKED AS DECOYS ELSEWHERE.

	0-Mix + Chain React.	Differ By One	10 Block Decoy	Coinbase	Mordinals	P2Pool Output Merging	
0-Mix + Chain React.	0%	8.76%	< 0.01%	0.04%	0%	0%	Agreement Rate
Differ By One	10.02%	< 0.01%	0.88%	< 0.01%	0.36%	0.07%	
10 Block Decoy	0%	0.39%	0.02%	2.14%	2.09%	1.84%	
Coinbase	0.04%	0%	0%	0%	0%	36.42%	
Mordinals	0%	0.12%	0.21%	0%	0%	0.09%	
P2Pool Output Merging	0%	0%	0.05%	10.18%	0%	0.04%	
Collision Rate							

**Heuristic Definition.** Define  $M$  as the set of all P2Pool miners and  $O_m$  as the set of outputs owned by a miner  $m \in M$ . For every  $o \in O_m$ , consider the set of transactions  $T_o$  for which there exists at least one  $pk$  for every ring member  $R$  such that  $pk \in R$  and  $pk \in O_m$ . If  $T_o$  has multiple transactions, we only consider the one with the highest number of inputs.  $pk$  is likely to be the true spend of  $R$ .

**Evaluation.** The heuristic tags 11,370 ring members as true spent outputs, and a total of 269,194 labeled ring members. For 99 labeled ring members there exists a self collision, as 2 outputs are identified as the true spend in multiple input rings, and 48 input rings have multiple ring members identified by the heuristic as the true spend. This leads to a self collision rate  $SCR$  of 0.036%. To validate the heuristic, we treat the zero-mixin, chain reaction and 10 block decoy bug heuristics results applied to Monero mainnet transactions as ground truth. By comparing the Differ-by-One results to this ground truth, we identify 284 false positives and 4,963 true positives, yielding a precision of 94.59%.

#### F. Combined Analysis

When evaluating heuristics in the previous sections, we focused on the precision. We now turn to a pairwise comparison of the proposed heuristics. Given heuristics  $H_1$  and  $H_2$  that each make a statement about a ring member  $pk$  in  $R$ , an agreement means both heuristics yield the same label, and a collision otherwise. This has the benefit that we can compare the results of a true-spend identification heuristic with a decoy identification heuristic.

Let  $|H_1|$  and  $|H_2|$  be the number of ring members labeled by heuristics  $H_1$  and  $H_2$  respectively. We denote the number of agreements between two heuristics as  $A$  and the number of collisions as  $C$ . This is similar to  $TP$  and  $FP$ , but we intentionally use a different notation as we technically do not have ground truth available for all pairwise comparisons. We measure collision rate and agreement rate that we define as follows:

$$Collision\ Rate = \frac{C}{A + C}$$

$$Agreement\ Rate = \frac{A}{\min(|H_1|, |H_2|)}$$

Table I shows that collisions rates are mostly low, with a couple of exceptions. Differ By One has a collision rate of 10% with Zero-mixin + Chain Reaction, which we have already seen in the heuristic evaluation section. Coinbase also has a collision rate of 10.18% with P2Pool Output Merging. This points to the precision of the Coinbase heuristic being lower than previously estimated. We believe this heuristic is more useful when used with additional context. For example, it is safer to use if we have additional information that confirms the transaction of interest was not made by a miner or a mining pool. The agreement rate is low for most heuristics, which indicates that the heuristics are complimentary to each other. The highest agreement of 36.42% exists between the Coinbase heuristic and the P2Pool Output Merging heuristics. This is expected since the P2Pool Output Merging heuristic leads to coinbase outputs being marked as decoys elsewhere.

Figure 8 shows the percentage of decoys identified by each heuristic over time. Zero-mixin + Chain Reaction can be used effectively to evaluate the impact of heuristics for old transactions, but they have almost no impact on recent transactions. For recent transactions, 10 Block Decoy Bug and the Coinbase heuristic have the highest impact.

By applying the chain reaction heuristic to the results of all previously described heuristics, combined with the results of the zero-mixin heuristic, we identify 40,005 additional true spent outputs and 61,928 additional decoys. The impact is low compared to other heuristics, but it is worth noting that those spent outputs and decoys can only be identified by combining results of all heuristics.

Figure 9 shows how the effective ring size has changed over time. In August 2022, the mandatory ring size was increase to 16, but the effective ring size was lower than 14, primarily because of the spike in the number of coinbase outputs caused by P2Pool. The effective ring size started to go up in early 2023 after P2Pool introduced an upgrade that made their payouts more efficient and reduced the number of coinbase

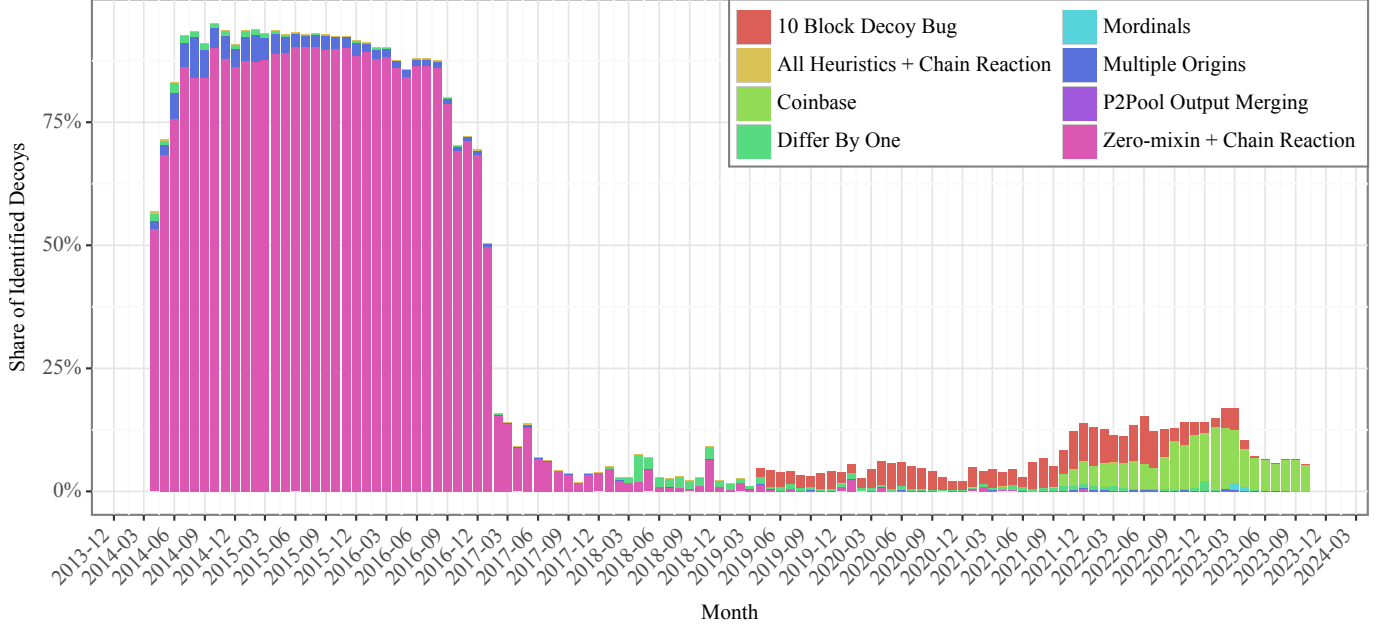


Fig. 8. Share of Identified Decoys per month, colored by heuristic. In alignment with earlier works, the Zero-mixin + Chain Reaction heuristic was very effective between 2014 and 2018. Differ By One has an overlap during that time pointing to this particular bug primarily being present in the early years of Monero. Most interestingly, the 10 Block Decoy Bug heuristic is among the most effective in recent years, abruptly ending with the vulnerability disclosure in May of 2023. Of all heuristics, the coinbase decoy identification heuristic remains the most applicable at the end of 2023.

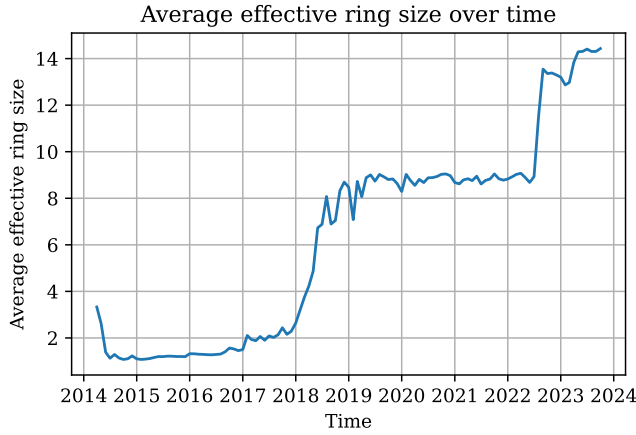


Fig. 9. Throughout the past years, the mandatory ring size of Monero input rings was increased with protocol upgrades. Between 2018 and 2022, the mandated ring size was 11, and we can show that the average effective ring size has been around 9. With the latest upgrade in August 2022, the mandatory ring size was increased to 16. After applying the heuristics described in this paper, the average effective ring size is still above 14 in October of 2023.

outputs they generated. Overall, the figure shows that the heuristics described in this paper have some impact on the effective ring size, but the impact is relatively low since the effective ring size is still higher than 14.

## V. DISCUSSION

We now turn to discussing the results of our heuristic analyses. In general, for most heuristics the collision rate is low. There exists some agreement between several heuristics, but ultimately the size of the ground truth originating from Zero mixins and chain reaction heuristics is very small since 2018, and will diminish even further. The 10 block decoy bug heuristic can therefore be considered as the best alternative among the available options, which is the reason we’ve used it to evaluate some of the other heuristics. Nevertheless, the 10 block decoy bug heuristic only works really well on input rings generated by the wallet2 library. In reality, its precision is likely slightly lower than what we were able to determine, as there is more client diversity on the Monero mainnet.

We could not use the Monero testnet dataset to evaluate most of our heuristics, as it does not contain transactions spending Mordinals, coinbase outputs, contains outputs from P2Pool, or exhibits the Differ By One phenomenon. Regarding the latter heuristic, we stipulate that the origin of this pattern is that there exists one or more wallet applications that do not sample ring members correctly, and instead perhaps cache a list of previously used decoys.

We did not include an analysis of the existing heuristics closed set and intersection sets originating from Monero forks as we wanted to focus on recent developments instead. We could have used the ground truth result of the dulmage-mendelsohn decomposition as proposed by Vijayakumar [14], but the results are nearly identical to the use

of zero mixins + chain reaction, and we decided against the overhead of implementing the underlying algorithm in favor of lower implementation complexity.

We also want to note that apart from the Differ By One heuristic, in particular the P2Pool Output Merging heuristic has had higher collisions with the Coinbase exclusion heuristic. We have been very restrictive in the requirements for performing this heuristic, and in a practical setting we would recommend to apply this heuristic very cautiously as well.

This leads to the final discussion point. As Deuber et al. [3] have recently shown in their work on the underlying assumptions around cryptocurrency deanonymizations, every heuristic is based on some assumptions, and can yield false positives. It is therefore crucial to keep the possibility of false positives in mind, and be very cautious when applying such heuristics in real world cases.

## VI. CONCLUSION

This paper contributes significant new insights to the field of cryptocurrency privacy, particularly within the Monero ecosystem, focusing on developments between 2019 and 2023. Our comprehensive work delves into the intricacies of several key heuristics, including the '10 Block Decoy Bug', 'Differ-by-one', Coinbase, Mordinal decoy identification, and a P2Pool specific output merging heuristic, grounded in the analysis of publicly known Miner payouts. While these topics have been discussed within the Monero community on platforms like Github and Reddit, our study stands out by providing a systematic, comparative, and combined evaluation of these methodologies.

Our findings illustrate that most of these heuristics demonstrate high precision, with the '10 Block Decoy Bug' and the Coinbase decoy identification heuristics having the most significant impact in the period from 2019 to 2023. Notably, the '10 Block Decoy Bug' heuristic can serve as an evaluation baseline for future heuristics applicable within this timeframe.

A crucial aspect of our analysis is the measurement of each heuristic's impact over time, including the assessment of the effective ring size of Monero's transaction inputs up until October 2023. This comprehensive approach has enabled us to provide a nuanced and detailed understanding of the current state of privacy in the Monero ecosystem.

Finally, we want to highlight that every heuristic is based on certain assumptions and can yield false positives. This underlines the necessity of acknowledging the potential for inaccuracies and exercising caution when applying these heuristics in real-world scenarios. Our findings contribute not only to the academic discourse but also offer practical insights that could guide future developments in studying privacy in cryptocurrency transactions.

## REFERENCES

- [1] N. van Saberhagen, "Cryptonote v 2.0 (white paper)," 2013.
- [2] S. Noether, A. Mackenzie et al., "Ring confidential transactions," *Ledger*, vol. 1, pp. 1–18, 2016.
- [3] D. Deuber, V. Ronge, and C. Rückert, "Sok: Assumptions underlying cryptocurrency deanonymizations," *Proceedings on Privacy Enhancing Technologies*, vol. 3, pp. 670–691, 2022.
- [4] S. Ghesmati, W. Fdhila, and E. Weippl, "Sok: How private is bitcoin? classification and evaluation of bitcoin privacy techniques," in *Proceedings of the 17th International Conference on Availability, Reliability and Security*, 2022, pp. 1–14.
- [5] F. Victor, "Address clustering heuristics for ethereum," in *Financial Cryptography and Data Security: 24th International Conference, FC 2020, Kota Kinabalu, Malaysia, February 10–14, 2020 Revised Selected Papers 24*. Springer, 2020, pp. 617–633.
- [6] A. Kumar, C. Fischer, S. Tople, and P. Saxena, "A traceability analysis of monero's blockchain," in *Computer Security – ESORICS 2017*, S. N. Foley, D. Gollmann, and E. Sneekenes, Eds. Cham: Springer International Publishing, 2017, pp. 153–173.
- [7] M. Möser, K. Soska, E. Heilman, K. Lee, H. Heffan, S. Srivastava, K. Hogan, J. Hennessey, A. Miller, A. Narayanan et al., "An empirical analysis of traceability in the monero blockchain," *Proceedings on Privacy Enhancing Technologies*, vol. 2018, no. 3, 2018.
- [8] A. Hinteregger and B. Haslhofer, "Short paper: An empirical analysis of monero cross-chain traceability," in *Financial Cryptography and Data Security: 23rd International Conference, FC 2019, Frigate Bay, St. Kitts and Nevis, February 18–22, 2019, Revised Selected Papers 23*. Springer, 2019, pp. 150–157.
- [9] Z. Yu, M. H. Au, J. Yu, R. Yang, Q. Xu, and W. F. Lau, "New empirical traceability analysis of cryptonote-style blockchains," in *International Conference on Financial Cryptography and Data Security*. Springer, 2019, pp. 133–149.
- [10] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.
- [11] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs: Short proofs for confidential transactions and more," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 315–334.
- [12] D. A. Wijaya, J. K. Liu, R. Steinfeld, and D. Liu, "Transparency or anonymity leak: Monero mining pools data publication," in *Information Security and Privacy: 26th Australasian Conference, ACISP 2021, Virtual Event, December 1–3, 2021, Proceedings 26*. Springer, 2021, pp. 433–450.
- [13] D. A. Wijaya, J. K. Liu, R. Steinfeld, D. Liu, and J. Yu, "On the unforkability of monero," in *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, 2019, pp. 621–632.
- [14] S. Vijayakumaran, "Analysis of CryptoNote Transaction Graphs Using the Dulmage-Mendelsohn Decomposition," in *5th Conference on Advances in Financial Technologies (AFT 2023)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), J. Bonneau and S. M. Weinberg, Eds., vol. 282. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023, pp. 28:1–28:22. [Online]. Available: <https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.AFT.2023.28>
- [15] A. L. Dulmage and N. S. Mendelsohn, "Coverings of bipartite graphs," *Canadian Journal of Mathematics*, vol. 10, p. 517–534, 1958.
- [16] S. Aenech, J. O. Chervinski, J. Yu, and N. Zlatanov, "New attacks on the untraceability of transactions in cryptonote-style blockchains," in *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 2021, pp. 1–5.
- [17] ACK-J, "Lord of the rings: An empirical analysis of monero's ring signature resilience to artificially intelligent attacks," Monero Research Lab, GitHub, Technical Report, 8 2022, final Report for Multidisciplinary Academic Grants in Cryptocurrencies. Available online: [https://raw.githubusercontent.com/ACK-J/Monero-Dataset-Pipeline/main/Lord\\_of\\_the\\_Rings\\_An\\_Empirical\\_Analysis\\_of\\_Monero\\_s\\_Ring\\_Signature\\_Resilience\\_to\\_Artificially\\_Intelligent\\_Attacks.pdf](https://raw.githubusercontent.com/ACK-J/Monero-Dataset-Pipeline/main/Lord_of_the_Rings_An_Empirical_Analysis_of_Monero_s_Ring_Signature_Resilience_to_Artificially_Intelligent_Attacks.pdf).
- [18] M. P. Krawiec-Thayer, "ringxor," <https://github.com/Mitchellpk/ringxor>, 2022.
- [19] D. Wijaya, J. Liu, R. Steinfeld, and D. Liu, *Transparency or Anonymity Leak: Monero Mining Pools Data Publication*, 11 2021, pp. 433–450.