

Report Update June 2023; Epistemology of Decoy Systems; Probing the Attacks on the Privacy of the Monero Blockchain

Nathan Borggren

CompDec

(Dated: July 7, 2023)

In an effort to address adversarial attacks on the privacy of the Monero blockchain, in which an adversary utilizes leaked or privileged information to gain information about the global network, a mathematical and accompanying computational environment will be developed that is amenable to analytical, empirical, and simulated analysis. The proposed work will quantify, measure, and analyze the structure and sensitivity Monero has with respect to the fungibility of the Monero coins, the connectivity of merchants and their customers, and how this structure can change as the microeconomics of repeated transactions is correlated and associated. Results, code, and recommendations will be shared with the community with the goal of quantifying, qualifying and improving its privacy capabilities.

The report update is included as additional comments marked in red for the sake of checking and maintaining context.

I. INTRODUCTION

This proposal seeks financial and technical support to provide analysis in abstracto, in silico, and ultimately in situ of Monero’s privacy features. We seek to address the concerns raised about decoy systems in [1]. The particular attacks raised in that work are the

- **Overseer Attack** - In which a merchant (receiver) uses their privileged transaction information to deduce repeat customers and build up a consumer model of their customers. This may already be a severe problem given the fraction of transactions that involve an exchange as one of the counter-parties.
- **Flashlight Attack** - A sender repeats transactions to an address or collection of addresses, providing them the means to link stealth addresses in the blockchain, and follow the taint tree forward to identify funds being aggregated (perhaps at a subpoena-able exchange).
- **Tainted Dust Attack** - Similar to a Flashlight Attack, but may involve a large number of candidate addresses, with an effort of identifying which address leads to a particular user-pattern.

though other related information leaks can also occur:

- **Anomalous or unique wallet configurations.** Users may inadvertently identify themselves by making unique choices for RingCT or number of outputs, use of flags, additional information etc. In previous machine learning blockchain analysis, features involving these choices were a primary source of leaks, as evidenced by their preference by random forests, neural networks, and other classification schemes.
- **Anomalous or unique decoy selection.** If a wallet or user uses non-standard decoy selection, the distributions used can often be teased apart from standard decoy selections by common statistical means.

- **Broadcasting of private information.** ShapeShift’s API leaked a mass of information for a number of years. Mining pools tend to broadcast information about coinbase transactions.
- **Mordinals and anomalous *tx_extra*.** Some users are indifferent or otherwise misaligned from the privacy use-case of Monero and seek to immortalize themselves on the blockchain. This behavior has a network effect of reducing the effective ring size of subsequent transactions.

All of these memory leaks share the common properties of leaking information about the ring inputs, the tx outputs, or both. If the decoy system is working ideally and as intended then these effects should be localized and protect both senders and receivers from each other as well as third parties. However, much work needs to be done to quantify, qualify, and improve the systems. This proposal seeks to do so with a Three-Pronged Approach: Theoretical, Computational, and Experimental.

The theoretical framework will build upon previous graph theoretical work which is largely concerned with asymptotic and/or stationary behavior. The computational work proposes to use statistical machinery like pymc, to efficiently explore a vast number of scenarios in a controlled manner, allowing information to be incorporated into the models in a Bayesian sense and simulate the effect of introducing dramatic changes. The empirical framework will take advantage of the test network and if deemed prudent, the real network to analyze the transaction graphs as they appear on actual blockchains.

An advantage of the multi-pronged approach is to incorporate other interested parties into the analysis as will be discussed in the final *Proposal Request* section.

A. Summary

Progress has been made on each of the three fronts: Theoretical, Computational, and Experimental described at the end of each of their respective sections. A technical document has been started and a computational package, called *MoneroAna* has also been started. These will be migrated to git in the coming weeks and shared with the working group. I will also be migrating and reproducing the work in the computing environment provided by

Rucknium and Gingeropolous.

Progress includes:

- monerod running and synced
- onion-monero-explorer instantiated and serving json
- *MoneroAna* python package started
 - taint tree generation - a depth/breadth first search algorithm modified to yield the taint tree by likeliness
 - taint tree sampling - A Monte Carlo tree search was designed to sample the tree quickly
- Analysis note started

II. THEORETICAL FRAMEWORK

Previously there has been graph theoretical work into graph-based attacks that have provided some comfort into the viability of decoy systems [2, 3], but more effort needs to be made to see this established in the wild. In particular the privacy loss from repeated transactions with another party needs to be quantified and measured. Additional privacy features beyond decoys might be necessary.

Topological Data Analysis (TDA) is an approach to algebraic topology that prioritizes the analysis of real world data. [4, 5] As graphs are the '1-skeleton' of the simplicial structures of algebraic topology a connection can be made with the graph theory of previous work. In addition to the vertices and edges present in graph theory, TDA allows for higher order simplices to be constructed. For example faces consisting of three vertices (V_1, V_2, V_3) can aid in incorporating the additional information gained in situations like the Overseer Attack and the Eve-Alice-Bob-Eve (EABE) Attack and related situations.

To be more clear consider a merchant who has received payment from two different stealth addresses. Only the merchant can know that the public blockchain info (t_1, v_1) and (t_2, v_2) can be extended to include (t_1, v_2) , (t_2, v_1) , (t_1, t_2, v_2) , (t_1, t_2, v_1) , (t_1, v_1, v_2) , (t_2, v_1, v_2) , and (t_1, t_2, v_1, v_2) . This additional structure can shorten path lengths, change connectivity and

alter other topological variants non-trivially. All of this additional information advantages the merchant seeking to draw inferences about their clients.

The advantage is a technical one, opening up connections to mathematical formalisms, as well as computational tools. These tools include further graph analysis [6, 7] and multiple python packages [8, 9]. It is our belief that Graph Theory, (including bipartite graphs and hyper graphs), Topological Data Analysis, Optimal Transport, and probabilistic methods are the most capable of revealing any cracks that may be present in the Monero facade.

Additionally these structures will be able to serve as *domains* for functional calculations. Probability and statistics can be introduced in various ways. One such approach is that of Optimal Transport. In this context the transport is of the coins themselves from their genesis in a coinbase transaction to the space of users. The coin values play the role of mass in the theory. We are imagining the equiprobable choice of priors, where the decoy systems are behaving optimally, to be an important reference frame to evaluate the enhancement to the probabilities that the information gains bring. We also expect that although there are a vast number of paths, the paths still cluster around this reference frame. This situation occurs in statistical mechanics out of equilibrium and is the analog of Maximum Entropy in the non-equilibrium context, known as Maximum Caliber.

Taint trees have been constructed and towards this end a generator of random paths was created. Below shows a typical path to a coinbase. Every transaction in Monero is characterized uniquely by the collection of all such paths to the coinbase. To find an expectation value for the actual value of a transaction, we sample the taint tree, giving successive approximations to the number of paths and the value associated with that path, and summing up all of the contributions. A typical path to coinbase is shown below.

```

: for i in get_random_paths(txo):
    print(i, i.block_height, i.xmr_outputs)

28b67861c9080cb22152b635d9675e57365b9d17dc9defe96e5177acc3fa5e48 2495960 0
75266b75ff2edf0c37cb7e65a4f4a3fa54daf5441b4d15e2a9ca5295bdd98f1c 2495892 0
1590692c38d9930ccd276915c5152e538fa88a2de52c71fa1b17e2a3230a3cb3 2324675 0
f17861a450a6c6df5131afae42e717a5464fd7f4df9a1d3a7adc14fd79ec16d4 2323441 0
f719655671bb4b905f072bfff7ddcb0627a096cee3ec78650653834087c3558c9 2323329 0
176b3664073386ed497110b26ada141101b5be5fa90bffa82041e90df66a0183 2323308 0
7d2019126566e73147940df238524b27110c28b748cc3400b0eb6d6d71932f2a 2323296 0
bab70ebfb5dc32919632afd5b1362f206655841a9a206ccde0f64ddbde9bfffcc 2312135 0
62a20468a55feb9b11d28775f24a01cfb49a4bafb12dd6e23632aa2e7b0cd670 2298261 0
cdd13aa9040cb750bcbdf642c3788b16d67948400409ba9b9e38e34d68b6ee97 2297590 0
85b52e66eb00c49f2ad010ea780840175280b40e0906246cabe4bf69938a8dfc 2292469 0
c898222d9e602aefc28ff9517e94f3229f266c8d35b18d315457bf5ea6f9875a 2291767 0
3d8efb63cal1f49e23b405ec9a600619e379c6675c5fd28e296e9d28056473224 2288920 0
1d7286120f93338866d8398cd7bb602bf3514c74579f8757acf75c95351b9c42 2284775 0
10a445c34d5da68832ed3cd1367a35e558b2a07e861547e1c0d9ab3ca19ba9cc 2284694 0
0cfcfdca1b13c318575829e73c4fd4e7d28aee7731dea4e994b455041542d185 2284425 0
2b46a0d3a36ee6cbc4a720433eb6f3cb3136e5a1f883a7f23db645513728931f 2281217 0
f678e6c4860fc3233347fd4aef9893ad84eb4acfeaf6697abe73d6c20ab9a8 2270970 0
feb78908eebfa18df84e7c30f00c2f0b9d98d45f70a30f52cd8d7b8de84674b1 2269579 0
4967eabc14c56f79130642c0b85326f7441ca8022463d54099632874ceab7e5a 2269568 0
9efb46def136bcb55e17e1212e9b9b7401ba5fd0e8410cb681459810bde4346f 2269107 0
b74e2231bf2b6183c8988ee61b45b01376f157bc68d4ba9b6eec57e6cadb7807 2268903 0
f1c8abcf14f707734bf7d41ee4688cb241d6474f6592b857bb51b85a1b679b7c 2268863 0
bbd2ed7c5fcd19eee76d06087b47268dd6257bb60465e96007448c7e0834d1a2 2268513 0
a254c972f9f824b7266e4e2f962c3b07b6222836c73f28469eb4d284a2b56403 2160113 0
77905615882dcd7b430c5df159fb18eccb3a7ccff1336dd163c7903b8be2fee6 2160038 0
f83eb013ded2b79d faa0abf2260655a9fcaeea05a3d2029f8109000a1f344db7 2160019 0
170d24959e47cb439a06c8169236b06b88ebcc18200a0197d74bbcd6f0faa0b8 2159980 0
d2b2e8d4911792495f2a573a954ba32e352738a5bea856fde4945fcd02bb1670 2159664 0
b78ae85cc6043b3ac1370d98ed9676fabfcf5ac184d6df3764427cb1f7d83c09 2159468 0
14dc867e722c51314826a7f2ed220f0ff41d8e04b8fc3b31674b924b243e7cf2 2156008 0
69f0d4a3cb9f71ac3521de1c23fbf2dac596a6db174c00e2390b10d4acd03208 2155820 0
845d625e208a99b3c748a01babf95998da369cad129027a55d5925ff158651cd 2155086 0
f3cb28f217ea7ede9d0812869e59d0986bc526b0bf9ef86d277c2add8cb4d668 2154575 0
c8079952c21ff2b90432ec41e352994934ab8b0d406404460db1065e3ec09ca1 2142958 0
086bd8d7d6c97da49abe19e2ef917dacf48d919851281bc5e6271c7e72fe6295 2120674 0
d097b1ceaad434fa238fb65fe4503bfa30de175564a08839fb40676071bd19a5 2120600 0
37a68e93575fedb09a4490c320256791818f314ca35cc83e96b4ecd1129ecdde 2118979 0
3fff58f37f7a1ee9af39b5b7a12af53930d381bf1d5535a204f0b7aea913b8e3 2117610 0
df14ace56c030ca93098c2c8cce565324091e964de8da2a6908bf556a073a7ec 2116784 0
5822df2c8b310e40c3dc9c62cc28b12a2017fcb545398133218a6ba0e0844738 2116594 0
800981bc97e88922068db62507a654baa4b00591d6a228f7ccdbce0194b35625 1769108 0
5dd54c87dbdb8ab669ed0f64a7c1122692a835f643a899c0bb2fa51de995b132 1747995 0
4b676c4c614c05e45aa0fbe0c0b5159297b39b866ac70b2306b295d58e47d07 1747984 0
6e285eb85777653a2b765311402757c8bd68a46c0f1547c948ef07ae801a1abc 1701093 0
be5cdf284fca825c19a30dfbd05346adbbb8e2649b7744ed595c66a3c1dec6968 1684870 0
02ba78e0c1ac75b5c823d6099f6c435b7a5e09135ac15c24c33aa7a5aaf74ba7 1377040 6839436614589

```

121

122 An additional statistical choice is that of Bayesian Networks, where distributions are
123 defined over the various structures: like the edges, vertices, and faces. For example, although
124 a transaction is of unknown value on the Monero blockchain, this very transaction value
125 could be the subject of a leaked transaction. Even without the leak the transaction value
126 is constrained by the number of coinbase transactions that eventually lead to its inputs,
127 as decoys or otherwise. The Bayesian framework is a natural choice to incorporate the
128 information gain from this conditioned information. We will elaborate in the computational
129 framework section where we introduce the computational tools to actually perform these
130 tasks [10].

131

Although I haven't completely converged on selecting the exact perspective, progress has

been made on both the Bayesian and TDA theoretical fronts. I have been able to do without the ‘bipartite’ nature of the graphs, insisting on vertices of all the same varietal, namely just the transactions.

The main idea of TDA is the notion of ‘persistence.’ A parameter is swept and the topological features are tracked as that filtration takes place. In this setting there are a number of ways to consider this filtration. First off, a parameter can be swept from top to bottom or from bottom to top. The two natural choices for filtration are by probability or by time (block height), and seems to make the most sense sweeping from high probability to low probability or from current time back towards the genesis block. These two filtrations are largely the same here since it happens to be the case that Monero chooses from recent blocks moreso than old blocks for the decoy selection.

I’ve been exploring the effect of treating the RingCT as a simplicial complex of transactions. Consider a typical ring of a typical transaction:

ring members	blk	ring size
- 00: c6d9d7574905ef16d39a0a972cf97f967a7e0882cff126161d2b1f72aeb9cef3	02665615	11
- 01: 57b6decf1e0f086dd5ec1618cfa1d082c9dc77f667799728cb2f11ae01829ed3	02846310	16
- 02: a14dbc60ed6f36c0a4140d3ffe7d6bf7cc330d57c2c892d5c4ad7de27cbaa4ad	02878392	16
- 03: 5a970bc2c217513c6d1741c4d4dc5a15940f97080381961bed37a9d66b74b3a5	02888110	16
- 04: 1149103f8ee3c8a0b1ca9ae6face0e67448faa112c8a1ef2922b04fee11340fa	02891844	16
- 05: f4706576d39837be55be44925113fc79973f1ee5cf0a0d999e7c01a22be00ebb	02896312	16
- 06: 8f224199c7ea7a851dae545abce8acb3b4bfc1c14f5d3a7e218e7f216dc799d8	02901237	16
- 07: 4e361030c5c4e2929aa06d43d1b22449ca3b9c55c3d717052c353822959c5f32	02901238	16
- 08: 5c316e39cf967f57f7c6a0c78eee45e5db27350849c0a4832ef21f4851a66d95	02903960	16
- 09: 6f9d14979e4d10e695eda73149bf879ca6dee7fd39373e5d86e207b10b5a3660	02904229	16
- 10: c00290222132211c4a2637ed942836405c53eeff6a36d332b2240c734c4a1a8f	02904293	16
- 11: 2b69ffa5ad924f1fef70577666a17003699e98c79332fe0f09c72c18ce72305a	02904310	16
- 12: ae7a1a7d92823ca87547f0413926ed67fbd54a226fa359c404ca4867fb6cdae0	02904610	16
- 13: 9bdfa6c8ffef802949441b82ed585f448c01a9c6a5657258d7eca465ca6d8103	02904716	16
- 14: 0bc472ffae5925e15f22f03b031e49e8d14b6774ea679c838d9ca5f382492192	02904736	16
- 15: 9b7a524b331c13e1958edda435cc150789e807f170bfced93729d208fe254d3c	02904842	16

The full simplicial complex for this ring includes the 16 vertices, tx_{15}, \dots, tx_0 , the edges $(tx_{15}, tx_{14}), \dots, (tx_1, tx_0)$, the planes $(tx_{15}, tx_{14}, tx_{13}), \dots, (tx_2, tx_1, tx_0)$, the tetrahedron $(tx_{15}, tx_{14}, tx_{13}, tx_{12}), \dots$ and so on until you have the 16 dimensional simplex $(tx_{15}, tx_{14}, tx_{13}, \dots, tx_0)$. The simplex value to assign is the youngest block height in the simplex, for e.g. $(tx_{15}, tx_{14}, tx_{13}, tx_{12})$, would have the value 2904610.

There might be little motivation to keep all 16 dimensional simplices; the main advantage is it would allow for the presence of non-trivial persistent homology of up to order 15 while the main disadvantage is the additional bookkeeping. The two most important being the

zeroth order, which identifies the connected components, and the first order which identifies the circuits. Strictly speaking we only need up to two orders of persistent homology to detect the birth and death of the components and cycles needed for the EAE attacks, but the higher order homologies might be informative in unknown ways.

I am also using TDA in the construction and the analysis of the Taint Tree. For a given transaction, the collection of all possible histories can be enumerated, or at least sampled. I am interested into how the likelihoods of the paths cluster. Plotting all these likelihoods on a number line forms a spectrum of sorts; do these probabilities cluster at all into highly likely, possible, or rare? Ideally the decoy selection process would make this spectrum fairly flat, but my initial investigation suggests there is some structure to it.

The large scale remaining to do here is to calculate the intersections of these taint trees via topological methods

III. COMPUTATIONAL FRAMEWORK

The power of TDA comes through its use of Persistent Homology. A filtration is introduced to enable the calculation of a persistence diagram which quantifies noise and structures at all hierarchical levels. In the setting of Monero the scalar parameter behind this filtration can be the 'probability of being the real transaction.' As this probability parameter is swept from zero to one various topological features come and go. The interval of the parameter where these features 'persist' is the subject of persistent homology. These persistence diagrams reside in a metric space where the distance is given by the Wasserstein distance. It will be the object of the proposed research to formalize these ideas but we would like to sketch our initial approach.

Although the graph of the entire Monero blockchain is large, it may be within the realm of computational feasibility that a graph analysis is performed on the entire blockchain. Furthermore the subgraphs that arise out of the aided information in the case of the overseer attacks would be tractable computationally and we can presume readily available to an adversary, thus it is necessary to understand what information could have been leaked.

To move some of the theoretical work into the realm of a messy reality various packages are available, some of which will also be useful and appropriate in the Empirical Framework section. In my previous explorations of the subject I felt as though having rings composed

of entirely decoys (removing the ability to know for certain that some input was being spent) or having decoy outputs (red-herring outputs where the entire subsequent taint tree is obfuscating the truth) would be very problematic for blockchain analysts. With two outputs, both parties are of microeconomic interest; both were involved in the transaction. Adding additional decoy outputs not only provides more choices in subsequent transactions, but insures that these choices are working towards the obfuscation of the transaction graph. These and other hypothesis can be explored.

A python library has been started to interface with the blockchain, construct the various simplices, and compute the various homologies. The main dependencies are *pymc*, *pot* (*python optimal transport*). The TDA has been performed in pure python at the moment, trying to identify the circuits using intersections, will be migrating to some tda package (probably giotto-tda).

IV. EMPIRICAL FRAMEWORK

In a previous study [11], [12] we staged a Monero blockchain with 10-50 agents interacting with each other to probe connectivity of the resulting transactions. The machine learning therein was followed up on in another Magic Grant funded project [13]. I think the similar research going forward should utilize the test network rather than building a new blockchain from scratch. I would also like to draw attention to the correlation results therein, it signifies a new attack not addressed by the decoy system in use.

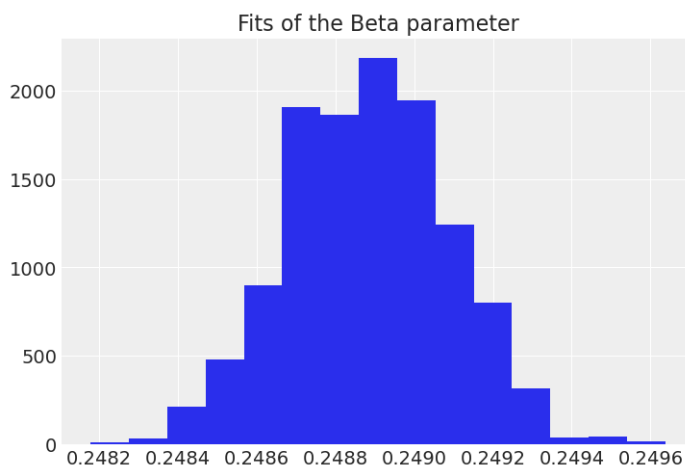
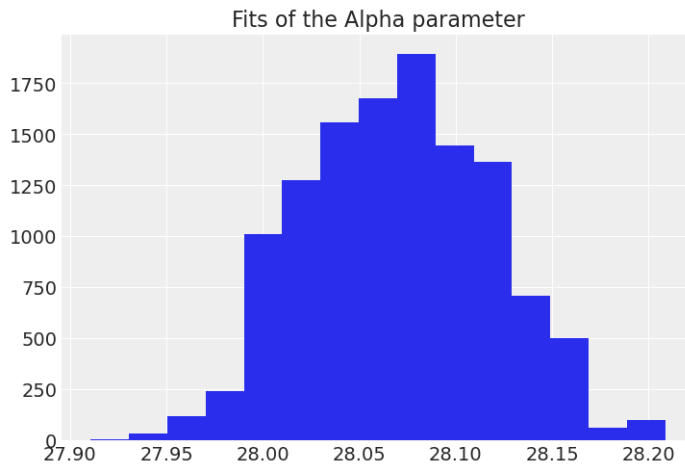
The Monero test network and stage net will be used to perform an empirical analysis of example Overseer attacks to avoid transaction costs and other considerations. Cat-and-Mouse games will be setup to mimic various microeconomic considerations in analogy to the attacks:

- Sender repeats transactions to a Receiver on a schedule.
- Receiver aggregates a collection of previous transactions on a timed schedule
- Receiver aggregates a collection of previous transactions when a threshold amount has been reached.
- Sender sends transactions to a collection of addresses in hopes to identify which addresses belong to a particular user.

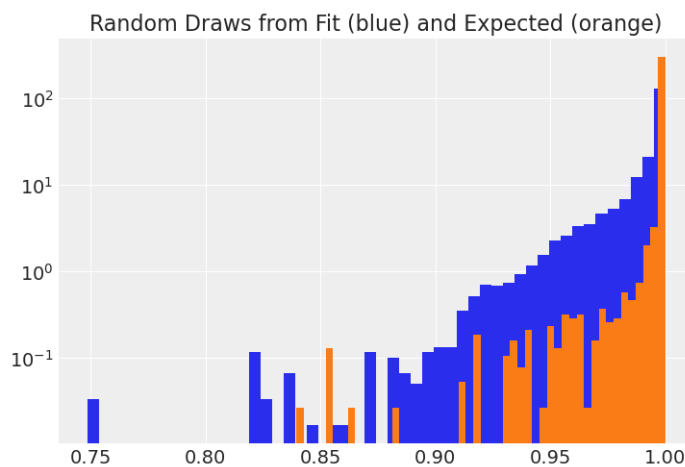
the resulting transaction graphs can then be analyzed and algorithm performance evaluated. It is possible that the decoy selection process may require such graph analysis, drawing from specific clusters in order to increase the fungibility between the clusters. We aim to connect and contribute to other ongoing Monero Research Lab research as fitting and able.

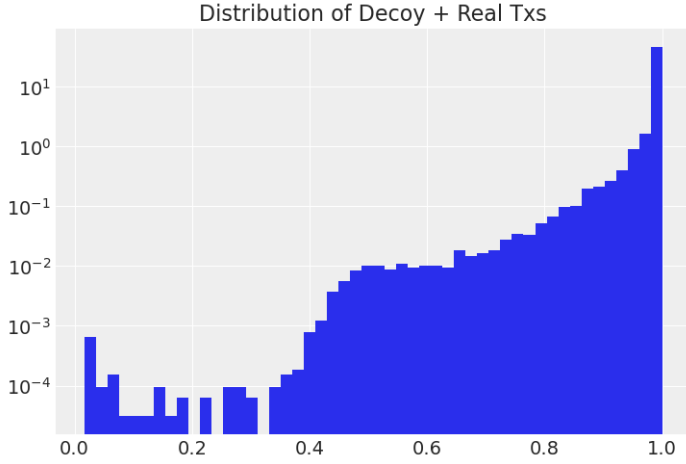
If the effort can be done safely, so as not to compromise existing consumers, the real Monero blockchain can also be analyzed. Motifs can be identified and histogrammed and previous leaks and residue from previous forks can be analyzed to see to what extent the blockchain has already been compromised. Churning strategies can be recommended or decoy selection can be motivated to diffuse this information loss over the continuing blockchain.

Bayesian stats were used in a variety of ways, starting with the fitting of the decoy distributions. A number of updates have been made to Monero since the last time I worked on it. The RingCT has gone from 11 to 16 and the decoy algorithm has improved. To evaluate the various probabilities of paths a clear understanding of how the decoy selection is made is required. I am still after this clear understanding, but this is the picture I've put together so far. Using the python package pymc (formerly pymc3) a (log of) 'Beta' distribution was fit to historic data. A couple of observations were made understanding the decoy selection. First note that the support for the Beta function is 0 to 1, so there is some subtlety into how the decoys are selected, the 0 corresponds to the genesis block and the 1 corresponds to the current block. I'm pretty sure the net effect of the 'log' beta remark in the codebase is that a draw from the beta distribution returns a number between 0 and 1, multiplying that by the current block height and rounding to an integer gives the decoy block height. The time stretching reminds one of the expansion of the universe. An adjustment that has been made in recent years is that the actual decoy selection used isn't a simply beta function, but there seems to be a window of recent transactions, where the beta distribution switches to a flat distribution. From my initial analysis it seems that this is doing well at the desired effect of mixing among the recent transactions. The results from the empirical fits gave values consistently above the 19.38, 1./61 factors suggested in the codebase. Typical fits over 2000 blocks are shown here. We will use these empirical fits to evaluate our probabilities for the time being, but have as a lower priority to-do to also fit the additional parameter of when the distribution turns flat. This cursory examination suggests choosing very old transactions is exceedingly rare.



We also show what the measured values look like, and what the distributions expected from the fit and expected distributions look like.





V. INTEGRATING THE FRAMEWORKS

Although it is the subject of the research to fully formalize these ideas, we can begin to see how the research may unfold and how the insights from the different approaches integrated. The information garnered from these analyses are of different varieties, deterministic, Bayesian, frequentist. Some aspects of which are more important depending on the questions being asked and the motives of those asking them. For example some level of de-anonymization might be sufficient to get a picture of what the microeconomic activity occurring on the blockchain is like but insufficient to identify or link the specific entities involved.

The graph theoretical and TDA approaches, in their basic form, are deterministic theories, they can potentially reveal certainties with respect to connectivity. The Monero blockchain itself, after many statistical choices like decoy selection, stealth address creation are made become part of Natural History, a fossil record of the specific things that occurred. It is a basic premise of Monero that ‘the one true subgraph’ consisting of all real transactions between real users can be sufficiently obfuscated with decoys, hidden in plain sight, so as to be rendered undiscoverable. The performance of this hypothesis in real world conditions versus powerful mathematical and computational machinery is what we are seeking to address. These conditions involve zooming into a potentially very small subset of transactions and diligently using all available resources to prune this tree into a clear picture of economic activity.

Indeed the fear behind the Overseer Attack is that these ‘red-handed’ connections can

be made given enough leaked information. In the same way a user has certainty over the transactions involved in their own activity, can an exchange achieve that same level of certainty on the origins of some funds through the systematic pruning of the edges their privileged information enables? This question can be asked deterministically and addressed through theoretical and empirical means. In which case the graph/simplex decompositions results in either a unique explanation of the scenario or a subgraph which still has a host of decoys present. This latter case is the more general case and the one anticipated by myself and others.

As mentioned in [1] this level of certainty isn't obligatory for the purveyors of these attacks and they may proceed with their queries at the level of probability of their choosing. At this stage we introduce probabilities and ask how the weights have changed on the decoy sets with the conditioned information. In the setting of Overseer Attack this investigation may take on similarities of the motif-hunting in [6] with the added calculation of likelihoods that these motifs and their subgraphs originated from the decoy selection process. A narrative of accuracy also begins when these probabilities get introduced. A bias towards recall over precision can be present in the case where False positives are of little importance; if the unscrupulous-attack-purveyor is not as much concerned with busting down the wrong door so much as the time wasted in the process. It is thus important to evaluate algorithms over the spectrum of relative priority between false positives and false negatives; algorithms with a high rate of false positive can still be quite dangerous for victims of these attacks.

It is most natural to think of how that evaluation process would proceed in the empirical framework section. We will actually try to execute a statistically significant number of these attacks on our own test network. Having the blockchain data as well as the private wallet data will allow us to evaluate the performance of yet-to-be-determined algorithms and mathematics and considering the false positive and false negative rates. Simulating the scenarios as well will allow us to establish a ground truth off of which our recommendations can be tested and evaluated.

VI. PROPOSAL REQUEST

The author is seeking a rate of 2/3 of 1 XMR/hour or approximately \$100/hour and has the means to put up to a full time effort (30-40 hours/week) in June-July 2023 and a part

time effort thereafter of approximately 15-20 hours/week. This rate is slightly less than the author's usual rate for data science work at the PhD level, but commensurate with previous funding from the Magic Fund. The rate is also less than comparable work at Chainalysis, Elementus, TRM Labs, and US Government whose ongoing work on Monero analysis does not necessarily feedback into the improvement of the protocol and indeed can be antithetical to the currency's objective. The total request is **220 XMR** over three months. Although this total is larger than a typical reward perhaps it can be combined with CCS or other mechanisms and is a competitive bid for the requested work.

This timeline is short compared to usual peer-reviewed research timescales but I think is sufficient amount of time to understand the breadth and depth of the issues and inform the Monero community of a path forward. It should also be clear at the end of this time if the Monero Foundation is interested into continuing the research, at which time new proposals can be submitted and further funding arrangements made as it sees fit.

I'd like to acknowledge the anonymous Monero community members that reached out to others and myself in order to create a small community of Monero/Graph Theory/Statisticians who have expressed interest and capabilities towards the problems these adversarial attacks present. Although this proposal does not seek particulars of the funding arrangements for those researchers, I will be grateful and open to their continuous feedback throughout the research process and contributing feedback to their efforts. My research efforts for the entire duration of the proposed work will be transparent to this working group to facilitate this feedback loop.

Part of the partitioning into *Theoretical*, *Statistical*, and *Empirical* components is to be able to scale the effort in a multitude of directions based on the needs of the community as well as the interests and capabilities of the other interested researchers. Having lead teams of data scientists from prestigious universities with skill levels from undergraduate, graduate, post-doc and even professorial levels, I feel equipped and available to lead the effort of scaling these analysis to whatever level the community deems appropriate.

VII. UPDATE

Versions of this proposal were released to the working group mentioned as well as the Monero Research Lab matrix channel. Isthmus, Xmrack and Rucknium responded with

feedback. There seems to be a preference for using the transaction graph of the real network as opposed to the test network. My worry of compromising other transactions through the analysis of our own seemed not to be shared. Doing the tests on the real network would be fine but would take cooperation and some Monero to provide those transactions. I don't personally want to steward Monero beyond dust transactions for this purpose, but perhaps the foundation or users who are willing to share transaction history from their wallets can help play the role of E, A, as well as B in the EABE attacks. It is my belief at this stage that the patterns created by these attacks will be consistent on either the test network or the real network, but that the patterns may be more 'hidden' with the added transactions on the real network.

My assertion about the timeline being short was also seconded, and third-ed. Analogous government contracts would likely be for a year, and indeed 6 months to 1 year might be more reasonable for true resolution of these issues. I am just trying to be consistent with the previous award amounts, but would be delighted to continue the work for longer at the same rate. I would certainly agree to **440 XMR** over 6 months as well, but wanted to give the foundation the option to make an assessment early in the research, and also an opportunity for the researchers that are enthusiastic about helping me with this work to apply and be compensated for their efforts as well.

Continuing I will be trying to formalize and measure the various EAE attacks which I will organize and enumerate. So far I have been trying to detect candidate EAE patterns in the blockchain, given labels I will try to mimic an exchange trying to identify their particular customer out of a host of possible patterns.

[1] Blockchain Privacy; Equal Parts Theory and Practice, 2023. URL <https://zfnd.org/blockchain-privacy-equal-parts-theory-and-practice/>.

[2] Christoph Egger, Russell WF Lai, Viktoria Ronge, Ivy KY Woo, and Hoover HF Yin. On defeating graph analysis of anonymous transactions. *Cryptology ePrint Archive*, 2022.

[3] Saravanan Vijayakumaran. Analysis of cryptonote transaction graphs using the dulmage-mendelsohn decomposition. *Cryptology ePrint Archive*, 2021.

[4] R. Rabadan and A.J. Blumberg. *Topological Data Analysis for Genomics and Evolution*:

Topology in Biology. Cambridge University Press, 2019. ISBN 9781107159549.

- [5] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2): 255–308, 2009.
- [6] Stephen Ranshous, Cliff A Joslyn, Sean Kreyling, Kathleen Nowak, Nagiza F Samatova, Curtis L West, and Samuel Winters. Exchange pattern mining in the bitcoin transaction directed hypergraph. In *Financial Cryptography and Data Security: FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, April 7, 2017, Revised Selected Papers 21*, pages 248–263. Springer, 2017.
- [7] Aric Hagberg and Drew Conway. Networkx: Network analysis with python. URL: <https://networkx.github.io>, 2020.
- [8] Guillaume Tauzin, Umberto Lupo, Lewis Tunstall, Julian Burella Pérez, Matteo Caorsi, Anibal M Medina-Mardones, Alberto Dassatti, and Kathryn Hess. giotto-tda: A topological data analysis toolkit for machine learning and data exploration. *The Journal of Machine Learning Research*, 22(1):1834–1839, 2021.
- [9] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, et al. Pot: Python optimal transport. *The Journal of Machine Learning Research*, 22(1):3571–3578, 2021.
- [10] Anand Patil, David Huard, and Christopher J Fonnesbeck. Pymc: Bayesian stochastic modelling in python. *Journal of statistical software*, 35(4):1, 2010.
- [11] N. Borggren, H.-Y. Kim, L. Yao, and G. Koplik. Simulated blockchains for machine learning traceability and transaction values in the Monero network, 2020.
- [12] N. Borggren and L. Yao. Correlations of multi-input Monero transactions, 2020.
- [13] An empirical analysis of monero’s ring signature resilience to artificially intelligent attacks, 2022. URL <https://github.com/MAGICGrants/Monero-Fund/issues/15#issuecomment-1086122008>.