UNIFORMLY MOST POWERFUL TESTS FOR AD HOC TRANSACTIONS IN MONERO

B G GOODELL, RIGO SALAZAR, AND FREEMAN SLAUGHTER

Abstract. We introduce a general, low-cost, low-power statistical test for transactions in transaction protocols with small anonymity set authentication (TPSASAs), such as Monero. The test classifies transactions as *ad hoc* (spontaneously constructed to spend a deterministically selected key) or *self-churned* (constructed from a probability distribution very close to that of the default wallet software, and with the same sender and receiver). The test is a uniformly most powerful (UMP) likelihood ratio tests (LRT) from the Neyman-Pearson Lemma, and makes no assumptions about user behavior. We extend these tests to expoit prior information about user behavior. We discuss test parameterization, as well as how anonymity set cardinality and user behavior impact test performance. We also describe a maximum-likelihood de-anonymization attack on Monero based on our test.

1. Introduction

Output-based transaction protocols (TPs) verifiably authenticate transactions with digital signatures ([11], [2]). With digital signatures, public keys are directly linkable with corresponding signatures, so it is simple to directly trace the chain of custody of funds. On the other hand, some TPs instead introduce *spender ambiguity* by authenticating transactions with anonymity sets of public keys, e.g. via ring signatures as in the original CryptoNote protocol ([15]) or zero-knowledge proofs ([6], [5]). Tracing the chain of custody of funds through anonymity sets with cardinality n requires computing correct paths through a directed acyclic graph where each node has n children, which is thought to be more difficult as n increases (and $n = 1$ is the non-ambiguous case).

Unfortunately, when these anonymity sets are small, transactions leak information about user behavior, e.g. via so-called EAE attacks ([9], [1]), allowing attackers to gain advantage at tracing funds. *Self-churn*, a folklore mitigation to these leaks, is the practice of punctuating real transactions with sequences of simulated transactions where the sender and receiver are the same ([7], [4], [14], [16], [13]). We call these intermediate simulated transactions *churn transactions*. On the other hand, real transactions are typically made *ad hoc* by spending one or more deterministically selected outputs.

The distributions of *ad hoc* transactions and churn transactions are different enough to build statistical tests exploiting the gap. We introduce a general statistical test which classifies transactions in TPs with small anonymity set authentication (TPSASAs, e.g. as in Monero via ring signatures) as *ad hoc* or churned. We call this the *basic test* for *ad hoc* transactions. The basic test is a uniformly most powerful (UMP) likelihood ratio test (LRT), due to the Neyman-Pearson lemma [12]. Assuming a prior distribution describing user behavior and/or more sophisticated

default wallet software, similar approaches yield *extended tests* for *ad hoc* transactions. The existence of these tests represent a tremendous security threat for users of TPSASAs, even when these tests have low power.

Our basic test is low cost and has variable, but generally low, power, and is related to previous attacks. Our extended tests are similarly low cost but can attain greater power by violating uniformity of the basic test. Test parameterization is generally very expensive, must be re-computed in the event of a blockchain re-organization, and must be re-computed regularly as the blockchain grows. Prototype python code for parameterizing our tests by estimating critical values of harmonic means of sets of random variables sampled from mixed distributions is available at this paper's repository on GitHub at `https://www.github.com/cypherstack/churn`.

Test power responds strongly to variations in the statistical distance between user-selected distributions and default wallet distributions, explaining practical security improvements since [10], [8]. Test power responds weakly to variations in anonymity set size, so increasing anonymity set sizes is not an effective mitigation. Our tests have no practical utility in protocols with very large anonymity sets (e.g. ZK-SNARKs in [6] or full-chain membership proofs in in [5]). Our tests have no practical utility when user behavior is very well-described by the default wallet distribution, whether due to low prevalence of *ad hoc* transactions or similarity between user behavior and the default wallet distributions. Unfortunately, if user behavior is very different from the default wallet distribution, prevalence of *ad hoc* transactions must be low before the basic test loses its practical utility.

1.1. **Organization.** This paper is organized as follows. In Section 2, we set out preliminary notation and background. In Section 3, we derive our basic test and describe its extensions. In Section 4, we discuss the parameterization of our basic test. In Section 5, we discuss tracing attacks against TPSASAs using our tests. In Section 6, we discuss our results. In Section 7, we draw some conclusions and make notes on directions for future work.

Changes

This paper is part of an open, iterative research process and will be regularly updated, with each revision accompanied by a detailed changelog. Future versions will include expanded analyses, new findings, numerical parameterization, additional experimental results, refinements to our formal models and their practical applications, corrections, and more. We use version control via GitHub at `https://www.github.com/cypherstack/churn`. The following describes changes made to this document since 21 October 2024.

- 21 October 2024. Initial draft.

2. Preliminaries and Notation

We use $\approx$ informally as "approximately equal," not to indicate statistically negligible differences.

Let $\epsilon > 0$. Let $n, m, d \in \mathbb{N}$. For any subset $T \subseteq S$, define the set complement $\overline{T} = \{s \in S \mid s \notin T\}$ as usual. Let I be the boolean indicator function, such that $I(P) = 1$ when proposition P is true and $I(P) = 0$ otherwise. For some real numbers $x_1, \ldots, x_n$, denote the tuple $\underline{x} = (x_1, \ldots, x_n)$ and, for any function f with

domain $\mathrm{dom}(f) \subseteq \mathbb{R}^n$, denote $f(\underline{x}) = (f(x_1), \ldots, f(x_n))$. Define the harmonic mean $H(\underline{x}) = n\left(\sum_{i=1}^n x_i^{-1}\right)^{-1}$ and the log harmonic mean $h(\underline{x}) = \ln(H(\underline{x}))$.

A probability mass function (PMF) on a finite set S is a function $f : S \to [0,1]$ such that $\sum_{s\in S} f(s) = 1$. We say a random variable X has PMF f if $\mathbb{P}[X = s] = f(s)$. Note that, generally, if f is a PMF on S, the function we denoted $f(\underline{s}) = (f(s_1), \ldots, f(s_n))$ is not a PMF.

We follow usual likelihood principles ([3]) as follows. For a parameter space Θ, subset $\Theta_0 \subseteq \Theta$, an element $\theta \in \Theta$, and a random variable X which takes on values in some set $\mathcal{X}$, and let $R \subseteq \mathcal{X}$ be a region. Then R corresponds to the rejection region of an inferential statistical test for a hypothesis that $\theta \in \Theta_0$.

A type I error occurs when $\theta \in \Theta_0$ yet $x \in R$. The probability of a type I error is the *size* $\alpha = \mathbb{P}_\theta[x \in R \mid X = x, \theta \in \Theta_0]$. The complement of the size $1 - \alpha$ is the *specificity* of the test. An upper bound on the size α a *level* of the test.

A type II error occurs when $\theta \notin \Theta_0$ and $x \notin R$. The probability of a type II error is the *miss rate* $\beta = \mathbb{P}_\theta[x \notin R \mid X = x, \theta \notin \Theta_0]$. The complement of the miss rate $1 - \beta$ is the *power* or *sensitivity* of the test.

Lemma 2.1 (Neyman-Pearson)**.** *Let $A \subseteq \Theta$ be a negligible set, $\theta_0, \theta_1 \in \Theta$ such that $\theta_0 \neq \theta_1$, $\alpha \in (0,1)$, and $k \in (0,\infty)$. For an unknown $\theta \in \Theta$, let $\mathcal{H}_0$ be the hypothesis that $\theta = \theta_0$, and let $\mathcal{H}_1$ be the hypothesis that $\theta = \theta_1$. If R is the rejection region for $\mathcal{H}_0$ corresponding to this test such that*

- *$x \in R \setminus A$ implies $\mathbb{P}[X = x \mid \theta = \theta_0] < k\mathbb{P}[X = x \mid \theta = \theta_1]$, and*
- *$x \in \overline{R} \setminus A$ implies $\mathbb{P}[X = x \mid \theta = \theta_0] > k\mathbb{P}[X = x \mid \theta = \theta_1]$,*

then the test has size α, is UMP in the set of level α tests, every UMP test in the set of level α tests also satisfies these conditions with the same k (but possibly different sets A), and every UMP test in the set of level α tests agrees with the others except possibly on their corresponding negligible sets A.

We say that the test is *parameterized* by k. Given a distribution for a random variable X with parameter space Θ, we define the usual *likelihood function* as

$$\mathcal{L}(\theta' \mid x) = \mathbb{P}\left[X = x \mid \theta = \theta'\right]$$

and, given subset $\Theta_0 \subseteq \Theta$, we define the *likelihood ratio* as follows.

$$\Lambda(x) = \frac{\sup_{\theta' \in \Theta_0} \mathcal{L}(\theta' \mid x)}{\sup_{\theta' \in \Theta} \mathcal{L}(\theta' \mid x)}$$

The likelihood ratio is a metric for the likelihood that $\mathcal{H}_0$ is true. Thus, rejection region R in the Neyman-Pearson lemma satisfies the following.

- Small likelihood is necessary to reject $\mathcal{H}_0$: $x \in R \setminus A$ implies $\Lambda(x) < k$.
- Large likelihood is necessary to accept $\mathcal{H}_0$: $x \in \overline{R} \setminus A$ implies $\Lambda(x) > k$.

Let $\mathcal{Y}$ be a finite set of public keys for a TPSASA (the entire key space). Let $y^* \in \mathcal{Y}$ be an unknown, fixed element. In the sequel, let f be a PMF on $\mathcal{Y}$ from default wallet software, which is used to sample anonymity set members.

We assume that the wallet distribution f is not dependent upon y^*. This assumption is not valid, in general, but in practice, f is usually only dependent upon the time at which the transaction was constructed, which we can approximate by proxy with the time the transaction appeared. Moreover, our approach below can be immediately modified so that f depends on y^*; we describe these modifications as we go. This, also, is not a valid assumption, in general, as users may sign a transaction

offline and wait a period of time before relaying it on the network. However, this is a rather unusual occurrence given the typical use-case of permissionless e-cash, which we describe below.

3. Basic Test for Churn

3.1. **Ad Hoc Anonymity Sets.** The usual use-case for permissionless e-cash is an *ad hoc* transaction. The user owns some fixed, unknown $y^* \in \mathcal{Y}$ and decides that y^* must be spent, so they do the following.

(1) Sample a distinguished index $1 \leq i^* \leq d$ with the default wallet software.
(2) Set $y_{i^*} = y^*$.
(3) For each $1 \leq i \leq n$ such that $i \neq i^*$, independently sample $y_i \in \mathcal{Y}$ with PMF f employed by the default wallet software, re-sampling in the case of collisions. Then the tuple $\underline{y} = (y_1, \ldots, y_n)$ can be interpreted as an n-set.
(4) Use $\underline{y}$ as the anonymity set to authorize the transaction.

Consider observing $\underline{y}$ conditioned upon the event that this anonymity set comes from an *ad hoc* transaction spending $y^* = y_{i^*}$ for an unknown index i^*. The probability of this occurrence can be computed with the law of total probability and modeling the index i^* corresponding to y^* with a uniform random variable J on $\{1, \ldots, n\}$.

$$\mathbb{P}[\underline{y}] = \sum_{j=1}^{n} \mathbb{P}[\underline{y} \mid J = j]\mathbb{P}[J = j] \tag{3.1}$$

$$= n^{-1} \sum_{j=1}^{n} \mathbb{P}[\underline{y} \mid J = j] \tag{3.2}$$

$$= n^{-1} \sum_{j=1}^{n} \mathbb{P}[\underline{y} \mid y_j = y^*] \tag{3.3}$$

$$= n^{-1} \sum_{j=1}^{n} \prod_{\substack{i=1 \\ i \neq j}}^{n} f(y_i) \tag{3.4}$$

If the distribution f depends on y^*, there exists some family of PMFs f_y on $\mathcal{Y}$ and parameterized by $y \in \mathcal{Y}$ such that $\mathbb{P}[\underline{y} \mid y_j = y^*] = \prod_{i=1, i \neq j}^{n} f_{y_j}(y_i)$. The same approach works without the assumption that f does not depend on y^* (but our resulting basic test no longer is a UMP). In this case, we just obtain the following.

$$\mathbb{P}[\underline{y}] = n^{-1} \sum_{j=1}^{n} \prod_{\substack{i=1 \\ i \neq j}}^{n} f_{y_j}(y_i) \tag{3.5}$$

We only use Equation 3.4 in the sequel for simplicity of our analysis.

3.2. **Churn Anonymity Sets.** Churn transactions, on the other hand, are not just *ad hoc* transactions with the same sender and receiver, since y^* may be unlikely to appear in a random transaction. Indeed, churn transactions have anonymity sets $\underline{y}$ distributed like a sample of n elements without replacement from the default wallet PMF f. In particular, we must have $\mathbb{P}[\{y_1, \ldots, y_n\}] = f(y_1)f(y_2)\ldots f(y_n)$. However, $f(y^*)$ is not entirely within the user control. In practice, f and $\mathcal{Y}$ evolve over time, and anonymity sets in transactions are typically sampled only a (relatively)

brief time before the transaction is broadcast. Thus the equality may be relaxed slightly to approximate equality.

$$\mathbb{P}\left[\underline{y}\right] \approx f(y_1)f(y_2)\dots f(y_n) \tag{3.6}$$

Default wallet distributions f are not unimodal in practice, because they are locally sensitive to blockchain density. However, f does tend to be approximately unimodal with respect to the age of elements of $\mathcal{Y}$. So, if y^* is sufficiently young, then the user can wait a random period of time such that the observed $f(y^*)$ is not unusual-looking before constructing an otherwise *ad hoc* transaction. In this case, the transaction does not appear to have a real signer, and therefore the simulation cannot reveal anything about the true signer i^*.

To approximate a churn transaction, beginning at a secret time t and ending at time t', users may employ the following approach.

(1) Independently sample some $\tilde{y}$ with PMF f and compute secret $u = f(\tilde{y})$.
(2) Discard $\tilde{y}$ safely and keep u secret.
(3) Wait until a future time such that $f(y^*) \approx u$, then immediately construct the transaction with the *ad-hoc* approach from the previous section.
(4) Discard u safely.

The resulting sample $\underline{y}$ appears to be sampled from the wallet PMF. If an observer learns of u and/or the starting time t, they gain an advantage in distinguishing that the transaction is a churn transaction. If f is unimodal with respect to the age of elements of $\mathcal{Y}$ and y^* is not sufficiently young for this churn procedure to work, the user cannot spend y^* in a churn as described. Unfortunately, this forces the user to spend y^* as an *ad hoc* transaction. However, if this *ad hoc* transaction is to oneself, the user receives a new y' which is sufficiently young to be subsequently spent in a churn transaction.

Just as before, we can consider the case that f depends on y^*. We again use the law of total probability, an independent uniform random variable J on $\{1, 2, \dots, n\}$.

$$\mathbb{P}[\underline{y}] = \sum_{j=1}^{n} \mathbb{P}[\underline{y} \mid y_j = j^*]\mathbb{P}[J = j^*] \tag{3.7}$$

$$= n^{-1} \sum_{j=1}^{n} \prod_{i=1}^{n} f_{y_j}(y_i) \tag{3.8}$$

However, in the sequel, we use Equation 3.6 for simplicity of our analysis.

3.3. **Distribution of Anonymity Sets.** We model the distribution of anonymity sets with the following PMF g with parameter space $\Theta = \{0, 1\}$ with parameter $\theta \in \Theta$.

$$g_\theta(\underline{y}) = \mathbb{P}_\theta[\underline{y}] = (1-\theta)\prod_{i=1}^{n} f(y_i) + \frac{\theta}{n}\sum_{j=1}^{n}\prod_{i\neq j} f(y_i) \tag{3.9}$$

$$g_0(\underline{y}) = \mathbb{P}_{\theta=0}[\underline{y}] = \prod_{i=1}^{n} f(y_i) \tag{3.10}$$

$$g_1(\underline{y}) = \mathbb{P}_{\theta=1}[\underline{y}] = \frac{1}{n}\sum_{j=1}^{n}\prod_{i\neq j} f(y_i) \tag{3.11}$$

where $\theta = 0$ corresponds to a churn transaction and $\theta = 1$ corresponds to an *ad hoc* transaction.

3.4. **Test Statistic for Basic Test.** Now to test the null hypothesis $\mathcal{H}_0$ that $\theta = 0$ against the alternative hypothesis $\mathcal{H}_1$ that $\theta = 1$ with the Neyman-Pearson Lemma, we have the following likelihood function and likelihood ratio.

$$\mathcal{L}(\theta \mid \underline{y}) = g_\theta(\underline{y}) = \mathbb{P}_\theta[\underline{y}] \tag{3.12}$$

$$= (1-\theta)\prod_{i=1}^{n} f(y_i) + \frac{\theta}{n}\sum_{j=1}^{n}\prod_{i\neq j} f(y_i) \tag{3.13}$$

$$\Lambda = \frac{\mathcal{L}(0 \mid \underline{y})}{\max\left\{\mathcal{L}(0 \mid \underline{y}), \mathcal{L}(1 \mid \underline{y})\right\}} = \frac{g_0(\underline{y})}{\max\left\{g_0(\underline{y}), g_1(\underline{y})\right\}} \tag{3.14}$$

$$= \begin{cases} 1; & g_0(\underline{y}) > g_1(\underline{y}) \\ \frac{\prod_i f(y_i)}{n^{-1}\sum_j \prod_{i\neq j} f(y_i)}; & g_0(\underline{y}) \leq g_1(\underline{y}) \end{cases} \tag{3.15}$$

$$= \begin{cases} 1; & g_0(\underline{y}) > g_1(\underline{y}) \\ H(f(\underline{y})); & g_0(\underline{y}) \leq g_1(\underline{y}) \end{cases} \tag{3.16}$$

Thus, our test statistic is just the harmonic mean of probabilities of the sample under f, $H(f(\underline{y}))$.

That the harmonic mean appears here may be surprising. The harmonic mean is biased towards the smallest element of a sample. If a sample of probabilities has a small harmonic mean, the sample must therefore have at least one small probability. In this way, the Neyman-Pearson lemma just formalizes the notion that the test statistic is small when at least one element of the sample $\underline{y}$ does not appear to have been sampled independently from f, i.e. $\mathcal{H}_0$ appears to be unlikely.

3.5. **Computing PMF.** To compute our test statistic requires computing $f(y)$ for various values of $y \in \mathcal{Y}$, where f is the PMF implied by the default wallet decoy selection algorithm. The function f is generally not easy to derive in a closed-form solution, even with access to default wallet source code. However, we can follow a few general rules to obtain a model for $f(y)$ which is sufficient for practical testing.

Recall $\mathcal{Y}$ is a description of the keys available for use as anonymity set members on the blockchain, $\mathcal{Y}$ has a structure imposed by the blockchain. In particular, there is a partition of $\mathcal{Y}$, one part for each block. The parts of this partition are linearly orderable by block height, and these different blocks have different cardinalities. The linear order can be interpreted as a clock, and for a given block B_i, we refer to the average number of keys per block in blocks close in time to B_i as the *blockchain density* at height i. Generally, for any $y \in B_i$, $f(y)$ is inversely proportional to blockchain density at height i.

Also, $f(y)$ is also dependent on the age of y, in terms of block height when y was mined. Since [10], it has been popular to empirically estimate the distributions of ground-truth spend-times, modeling these as Gamma-distributed random variables (which can be thought of as sums of exponentially-distributed random variables). Spend-time is a continuous random variable, whereas a sampled block on the blockchain is a discrete random variable, however, and the discrete analogue of the exponential random variable is the geometric random variable. Since a sum of geometric random variables is a negative binomial random variable, we can model

$f(y)$ as proportional to a negative binomial PMF, $\mathbb{P}[k] = \binom{k+r-1}{k}(1-p)^k p^r$, for some parameters $r > 0$ and $p \in (0,1)$. If these parameters are not specified directly in the default wallet software, they can be estimated using standard statistical inference techniques and samples from the default wallet software.

Hence, if we denote the local blockchain density near y with $\texttt{density}(y)$, the age of y in $\mathcal{Y}$ in terms of block height with $\texttt{age}(y)$, we have $f(y) \approx \frac{\binom{\texttt{age}(y)+r-1}{\texttt{age}(y)}(1-p)^{\texttt{age}}(y)p^r}{\texttt{density}(y)}$.

3.6. **The Basic Test.** The Neyman-Pearson Lemma intuitively rejects the hypothesis that the data was generated with parameter $\theta = 0$ if the likelihood that $\theta = 0$ is much smaller than the likelihood that $\theta = 1$. Note that the only negligible subset of $\{0,1\}$ is the empty set.

Corollary 3.1. *Let $\alpha \in (0,1)$, $c \in (0,\infty)$, $\underline{y} \in \mathcal{Y}^n$. For an unknown $\theta \in \{0,1\}$, let $\mathcal{H}_0$ be the hypothesis that $\theta = 0$ and let $\mathcal{H}_1$ be the hypothesis that $\theta = 1$. If R is the rejection region for $\mathcal{H}_0$ corresponding to this test such that*

- *$\underline{y} \in R$ implies $\mathbb{P}[\underline{y} \mid \theta = 0] < k\mathbb{P}[\underline{y} \mid \theta = 1]$, and*
- *$\underline{y} \notin R$ implies $\mathbb{P}[\underline{y} \mid \theta = 0] > k\mathbb{P}[\underline{y} \mid \theta = 1]$,*

then the test has size α, is UMP in the set of level α tests, every UMP test in the set of level α tests also satisfies these conditions with the same k, and every UMP test in the set of level α tests agrees with the others.

A rejection region which satisfies these conditions is $R = \{\underline{y} \mid kg_1(\underline{y}) > g_0(\underline{y})\}$. This region has a corresponding significance $\alpha = \mathbb{P}\left[\underline{y} \in R \mid \theta = 0\right] = \sum_{\underline{y} \in R} g_0(\underline{y})$, the resulting test has size α and is UMP in the set of tests with level α. Rewriting $kg_1(\underline{y}) > g_0(\underline{y})$ as $\Lambda = \frac{g_0(\underline{y})}{g_1(\underline{y})} < k$, we have the following rejection region

$$R = \{\underline{y} \mid H(f(\underline{y})) < k\} = \{\underline{y} \mid h(f(\underline{y})) < h_\alpha\}$$

where $h_\alpha = \ln(k)$ parameterizes the test at size α. This provides a one-sided test which rejects $\mathcal{H}_0$ whenever the (log) harmonic mean of the probabilities of occurrence is sufficiently small.

To run the basic test, the following is sufficient.

- Evaluate f n times.
- Compute n inversions of floating point numbers.
- Compute a mean of these n floating point numbers.
- Invert a floating point number.
- Comparing two floating point numbers.

If evaluating f takes time at most t_f, inverting a floating point number takes time at most t_{inv}, computing a mean of n floating point numbers takes time at most $t_{n,\text{mean}}$, and comparing two floating point numbers takes time t_{comp}, then the test takes time $O(n \cdot t_f + (n+1)t_{\text{inv}} + t_{n,\text{mean}} + t_{\text{comp}})$.

As we shall see, evaluating the wallet distribution f is, itself, a nontrivial challenge; see Section **??** for a discussion.

3.7. **Extensions.** Attackers may utilize additional knowledge about user behavior to develop non-uniform tests with better performance than our basic UMP LRTs. Indeed, exchanges and large economic actors with detailed ground-truth knowledge of user identities and locations can leverage these data. Such an actor could use the distribution g_1 from our basic test as a Bayesian prior, and uses their ground-truth knowledge to compute an update to develop a new prior $\hat{g}_1$, one for each user. Of

course, for every user, most of $\hat{g}_1$ is fixed by the default wallet distribution, and we have $\hat{g}_1(\underline{y}) = \sum_{j=1}^{n} \hat{f}(y_j) \prod_{i \neq j} f(y_i)$ for some $\hat{f}$ associated with the user. Thus we obtain the distribution

$$\hat{g}_\theta(\underline{y}) = (1-\theta) \prod_i f(y_i) + \frac{\theta}{n} \sum_j \hat{f}(y_j) \prod_{i \neq j} f(y_i) \tag{3.17}$$

$$= \hat{\mathcal{L}}(\theta \mid \underline{y}) \tag{3.18}$$

$$\hat{\Lambda} = \frac{\hat{\mathcal{L}}(0 \mid \underline{y})}{\max\left\{\hat{\mathcal{L}}(0 \mid \underline{y}), \hat{\mathcal{L}}(1 \mid \underline{y})\right\}} = \frac{\hat{g}_0(\underline{y})}{\max\left\{\hat{g}_0(\underline{y}), \hat{g}_1(\underline{y})\right\}} \tag{3.19}$$

$$= \frac{\prod_i f(y_i)}{n^{-1} \sum_j \hat{f}(y_j) \prod_{i \neq j} f(y_i)} \tag{3.20}$$

$$= H\left(\frac{f(y_1)}{\hat{f}(y_1)}, \frac{f(y_2)}{\hat{f}(y_2)}, \ldots, \frac{f(y_n)}{\hat{f}(y_n)}\right) \tag{3.21}$$

Following the same reasoning as for the basic test, we reject the null hypothesis $\mathcal{H}_0$ that $\theta = 0$ whenever $\hat{\Lambda}$ is sufficiently small. We have the following rejection region

$$R = \left\{\underline{y} \mid H\left(\frac{f(y_1)}{\hat{f}(y_1)}, \ldots, \frac{f(y_n)}{\hat{f}(y_n)}\right) < k\right\} = \left\{\underline{y} \mid h\left(\frac{f(y_1)}{\hat{f}(y_1)}, \cdots, \frac{f(y_1)}{\hat{f}(y_n)}\right) < h_\alpha\right\}$$

where $h_\alpha = \ln(k)$ parameterizes the test for size α, providing a similar one-sided test to the basic test.

To run the extended test, the following is sufficient.

- Evaluate f and $\hat{f}$ n times each.
- Compute n inversions of floating point numbers.
- Compute n products of floating point number pairs.
- Compute a mean of these n products.
- Invert a floating point number.
- Comparing two floating point numbers.

If evaluating $\hat{f}$ takes time $t_{\hat{f}}$, computing a product of two floating point numbers takes time t_{mul}, then the extended test takes time $O(n \cdot (t_f + t_{\hat{f}} + t_{\text{mul}}) + (n+1)t_{\text{inv}} + t_{n,\text{mean}} + t_{\text{comp}})$.

4. Basic Test Parameterization and Power

To apply this test with a given α, we require a value h_α such that $\alpha = \mathbb{P}_{\theta=0}\left[\underline{y} \in R\right] = \mathbb{P}_{\theta=0}\left[h(f(\underline{y})) < h_\alpha\right]$. An estimate of $h_\alpha$is possible without any additional assumptions or ground-truth data. We explain the general approach in this section.

See Figure 4 for a diagram showing the idea for an arbitrary unimodal distribution on $h \in \mathbb{R}_{<0}$. The shaded region is the rejection region for the size α UMP LRT basic test for churn, where the dashed line is the critical value $h = h_\alpha$. The harmonic mean of elements from $(0, 1)$ is an element from $(0, 1)$, so $h < 0$, explaining the domain of Figure 4. Since anonymity set members are sampled from a finite set, our test statistic is a discrete random variable, but the general picture also applies for densities.

In the case of densities, the critical value h_α is the exact value such that $\alpha = \mathbb{P}_{\theta=0}\left[h(f(\underline{y})) < h_\alpha \mid \mathcal{H}_0\right]$; equivalently, the area of the shaded region is exactly α.

In the case of mass functions, $h_\alpha = \sup\left\{\ln(k) \mid \mathbb{P}_{\theta=0}\left[h(f(\underline{y})) < \ln(k) \mid \mathcal{H}_0\right] \leq \alpha\right\}$; equivalently, the area of the shaded region is as close to α as the distribution allows without exceeding α.

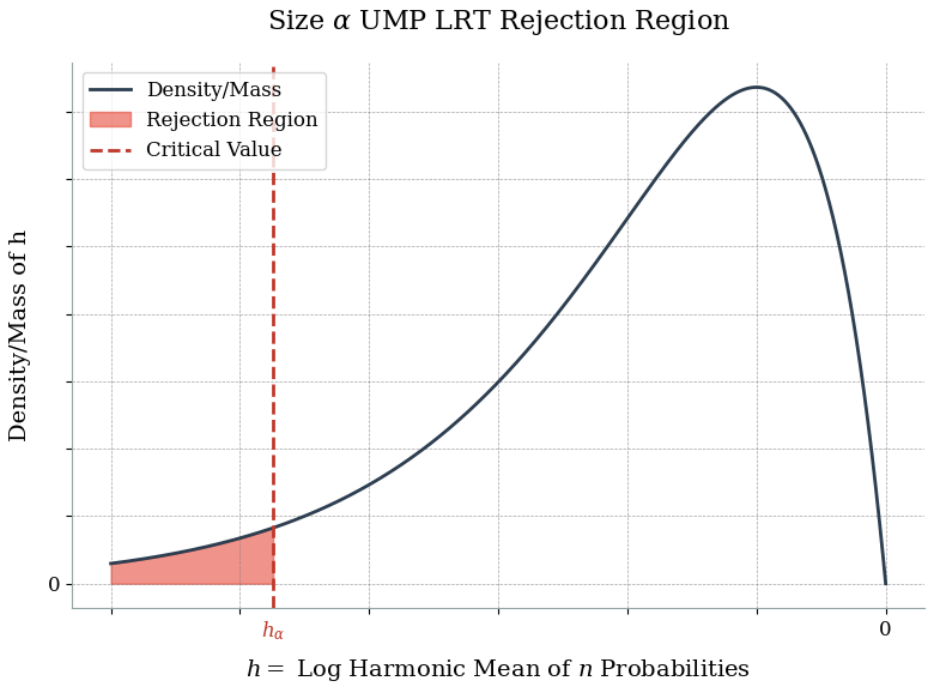

FIGURE 1. Illustrative sketch of a possible PMF (or PDF) for the log harmonic mean of n-tuples of probabilities under the null hypothesis $\mathcal{H}_0$; this is only a sketch, and should not be expected to resemble the ground-truth.

4.1. **Suggested Empirical Approach.** The distribution of $h(f(\underline{y}))$, conditioned upon $\underline{y}$ being sampled from f, is sensitive to the distribution of $f(\underline{y})$ under this same condition. No model of the distribution of the values $f(y)$ conditioned upon y being sampled from f is immediately obvious. An empirical distribution of $f(y)$ from Monte Carlo approaches is easily obtainable, though, from which we can estimate the 100α-th percentile critical value h_α as follows.

We use the fact that the ℓ^{th} order statistic in a sample of m observations is an unbiased estimate of the $100 \cdot \frac{\ell}{m+1}$-th percentile critical value, and compute the estimate for $\alpha = \frac{\ell}{m+1}$. For example, for $\alpha = 0.05$, the classic critical value used in mathematical statistics, set $\ell = 1$ and $m = 19$.

However, a point estimate for h_α alone does not provide a good description of the precision to which we can represent h_α. To rectify this, we use interval estimation, taking large enough sample sizes to guarantee a decimal representations with some $\kappa \in \mathbb{N}$ digits of precision in using scientific notation.

A popular option for interval estimation is the $100(1-\alpha^*)\%$ confidence interval.

$$\left(\overline{h} - t_{\frac{\alpha^*}{2},L-1}\frac{S}{\sqrt{L}}, \overline{h} + t_{\frac{\alpha^*}{2},L-1}\frac{S}{\sqrt{L}}\right) \tag{4.1}$$

where $\overline{h}$ is the sample mean, L is the sample size, $t_{\frac{\alpha^*}{2},L-1}$ is the $\frac{\alpha^*}{2}$-th critical value for Student's t-distribution with $L-1$ degrees of freedom, and S is the sample standard deviation. This interval contains an independently sampled $\overline{h}$ with probability around $1-\alpha^*$. However, our task is to find h_α such that, for an independently sampled $\underline{y}$, $\alpha = \mathbb{P}\left[h(f(\underline{y})) < h_\alpha \mid \mathcal{H}_0\right]$. Thus, the $100(1-\alpha^*)\%$ prediction interval in Equation 4.2 is more appropriate

$$\left(\overline{h} - t_{\frac{\alpha^*}{2},L-1}S\sqrt{1+\frac{1}{L}}, \overline{h} + t_{\frac{\alpha^*}{2},L-1}\sqrt{1+\frac{1}{L}}\right) \tag{4.2}$$

where $\overline{h}$, L, $t_{\frac{\alpha^*}{2},L-1}$, and S are all as in Equation 4.1. We suggest the following procedure with multilevel sampling.

(1) Input digits of precision $\kappa \in \mathbb{N}$, tuple length $n \in \mathbb{N}$, desired sizes $\alpha, \alpha^* \in (0,1)$, inner sample size $m \in \mathbb{N}$ such that $\alpha = \frac{\ell}{m+1}$ for some integer $1 \leq \ell \leq m$, and outer sample size $L \in \mathbb{N}$, and the $\frac{\alpha^*}{2}$-th critical value for the t distribution with $L-1$ degrees of freedom, $t_{\frac{\alpha^*}{2},L-1}$.
(2) For each $1 \leq i \leq L$, do the following.
   (a) For each $1 \leq j \leq m$, sample anonymity set $\underline{y}_{i,j} = \{y_{i,j,1}, \ldots, y_{i,j,n}\}$ and compute $h_{i,j} = h(f(\underline{y}_{i,j}))$.
   (b) Compute $h_{i,(\ell)}$, the ℓ^{th} order statistic of $\{h_{i,1}, \ldots, h_{i,m}\}$, and set $h_i = h_{i,(\ell)}$.
(3) Now we have a sample $h_1, h_2, \ldots, h_L$ of ℓ^{th} order statistics; we use the fact that the ℓ^{th} order statistic is an unbiased estimate of the $100\frac{\ell}{m+1}$-th critical value. Compute the following:

$$\overline{h} = L^{-1}\sum_i h_i \tag{4.3}$$

$$S = \sqrt{\frac{\sum_i (h_i - \overline{h})^2}{L-1}} \tag{4.4}$$

$$h_{\texttt{low}} = \overline{h} - t_{\alpha'/2,L-1}S\sqrt{1+\frac{1}{L}} \tag{4.5}$$

$$h_{\texttt{high}} = \overline{h} + t_{\alpha'/2,L-1}S\sqrt{1+\frac{1}{L}} \tag{4.6}$$

(4) If rounding the significands of the scientific notation representations of $\overline{h}$, $h_{\texttt{low}}$, and $h_{\texttt{high}}$ to κ digits does not yield the same result, increase L (say, by doubling the sample size) and repeate the process again.
(5) Otherwise, set h_α to be this rounded value and terminate.

We make no assumptions about user behavior, using only honestly sampled anonymity sets. Unfortunately, f evolves as the blockchain evolves, so the critical values h_α are sensitive to the state of $\mathcal{Y}$ at the time the transaction appeared. Thus, the same empirical sampling procedure needs to be repeated at different block

heights. Possibly, this may be optimized by binning blocks into epochs without sacrificing much performance.

This process is guaranteed to terminate, thanks to the law of large numbers and the central limit theorem. However, by doubling sample sizes repetitively, we risk wastefully oversampling, and by performing test statistic computations which are not rolling computations in between each doubling, we introduce a lot of unnecessary overhead.

While a small optimization to this basic algorithm is to compute $\overline{h}$ in a rolling fashion, we cannot do this with S. Another small optimization is to avoid wastefully oversampling and wasting time on intermediate computations which are not relevant to the final output. Do this with a small initial sample L' to get initial observations $\overline{h}'$, S', $h'_{\texttt{low}}$, and $h'_{\texttt{high}}$, then estimating an improved sample size L such that a follow-up sample of L elements is very likely to satisfy κ digits of precision.

Running this parameterization procedure only needs to be done once for each confirmed block height, and executed again in the case of a block reorganization. However, the procedure requires taking $L \cdot m \cdot n$ samples of from the wallet distribution f, evaluating f on each observation from this sample, computing the corresponding log harmonic means of the n-tuples from these samples, and finding the ℓ-th order statistics from the m-tuples of these log harmonic means. Lastly, we compute the mean and sample standard deviation, but this is done once and is very fast compared to all the previous.

Thus, if it takes time t_{sample} to sample an churned anonymity set with cardinality n from f, time t_f to evaluate f, time t_h to compute the log harmonic mean of an n-tuple, and time $t_{\text{ord},k,m}$ to compute the k-th order statistic from a sample of m log harmonic means, then it takes about time $O(Lmt_{\text{sample}}+Lmnt_f+Lmt_h+Lt_{\text{ord},k,m})$ to parameterize the scheme. Thus, once the test is parameterized, the test is very low-cost to execute, and the test only needs to be parameterized once. However, test parameterization takes $O(Lmn)$ time, and L generally grows quadratically with required precision. This, coupled with the fact that test parameterization must be occasionally done as the blockchain grows, test parameterization as described here is much more expensive than running the test itself.

4.2. **Model Selection.** In addition to empirically estimating h_α, we can perform goodness-of-fit tests on the empirical distribution of the values $v = f(y)$ to determine useful families models and infer parameters. Beta-distributed random variables, in particular, are capable of modeling many unimodal distributions on $(0, 1)$, and are a good candidate. Then, numerical integration or Monte Carlo techniques from the model can be used to determine h_α similarly as in the previous section, but from the inferred model instead of the empirically observed data. Indeed, in the previous section, we essentially used Monte Carlo sampling of anonymity sets from f to estimate h_α directly, whereas this approach first determines a model for $v = f(y)$ and uses numerical integration or Monte Carlo techniques from the model instead. As model selection improves, we expect these two approaches to produce similar results, providing a useful "reality check."

Prototype python code for estimating critical values of harmonic means of sets of random variables sampled from mixed distributions is available at this paper's repository on GitHub at `https://www.github.com/cypherstack/churn`. This code focuses on moding these v with beta distributions.

4.3. **Relating Parameterization to Power.** In this section, we relate β of our basic test with rejection region R to the test specificity $1-\alpha'$ of a *related* test with a similar rejection region R'.

Recall the following. A type I error occurs when a churned transaction is classified as *ad hoc*; this occurs with probability α and is called a *false positive*. Test specificity is the true negative rate, which is $1-\alpha$. These probabilities are selected during parameterization. A type II error, on the other hand, occurs when an *ad hoc* transaction is classified as churned; this occurs with probability β, and is called a *false negative*. Test sensitivity is the true positive rate, also known as test power, which is $1-\beta$. These probabilities are difficult to estimate without access to ground truth data, generally, but we have the following.

In the case that $\theta = 1$, every y_i for $i \neq i^*$ is sampled via f, and the final $y_{i^*} = y^*$ is fixed and unknown. For this fixed unknown y_{i^*}, we have a fixed and unknown $v = f(y_{i^*})$. Then, given that $\theta = 1$, $H(f(\underline{y})) > k$ if and only if

$$\frac{n}{v^{-1} + \sum_{i \neq i^*} f(y_i)^{-1}} > k.$$

Equivalently, $n > \frac{k}{v} + k\sum_{i\neq i^*} f(y_i)^{-1}$. Since $H(f((y_i)_{i\neq i^*})) = \frac{n-1}{\sum_{i\neq i^*} f(y_i)^{-1}}$, we have that $\sum_{i\neq i^*} f(y_i)^{-1} = \frac{n-1}{H(f((y_i)_{i\neq i^*}))}$, so $H(f(\underline{y})) > k$ if and only if $n > \frac{k}{v} + \frac{k(n-1)}{H(f((y_i)_{i\neq i^*}))}$. Thus, $n - \frac{k}{v} > \frac{k(n-1)}{H(f((y_i)_{i\neq i^*}))}$, or $H(f((y_i)_{i\neq i^*})) > k\frac{n-1}{n-\frac{k}{v}}$. For convenience, set $k' = k\frac{n-1}{n-\frac{k}{v}}$ and $\underline{y}' = (y_i)_{i\neq i^*}$. Note that, where $\underline{y}$ is an *ad hoc* transaction and is therefore sampled with $g_{\theta=1}$, $\underline{y}'$ is, on the other hand, sampled entirely with f.

Thus, we have the following key observation: k' parameterizes a rejection region R' for a test which is similar to our basic test, but for anonymity sets with cardinality $n-1$ instead of n. The corresponding test has rejection region

$$R' = \left\{\underline{y}' \mid H(f(\underline{y}')) < k'\right\} = \left\{\underline{y}' \mid H(f(\underline{y}')) < k\frac{n-1}{n-\frac{k}{v}}\right\}$$

and some corresponding probability of a type I error α' determined by the critical test statistic value $k' = k\frac{n-1}{n-k/v}$. Thus, the probability of a type II error for our basic test is

$$\begin{aligned}
\beta =& \mathbb{P}\left[H(f(\underline{y})) > k \mid \theta = 1\right] && (4.7)\\
=& \mathbb{P}\left[H(f(\underline{y}')) > k' \mid \theta = 0\right] && (4.8)\\
=& \mathbb{P}\left[\underline{y}' \notin R' \mid \theta = 0\right] && (4.9)\\
=& 1 - \alpha' && (4.10)
\end{aligned}$$

The size of the related test $\alpha' = \mathbb{P}\left[H(f(\underline{y}')) > k' \mid \theta = 0\right]$ coincides with the power of our basic test $1-\beta$.

Note $0 < \frac{k}{v}$ always, but $\frac{k}{v} < 1$ or not. So, we have two cases. In the first case, $\frac{k}{v} < 1$, so $n-1 < n - \frac{k}{v} < n$, so $\frac{1}{n} < \frac{1}{n-\frac{k}{v}} < \frac{1}{n-1}$, and $\frac{n-1}{n} < \frac{n-1}{n-\frac{k}{v}} < 1$. Of course, if n is sufficiently large, $\frac{n-1}{n} \approx 1$, so we must have $\frac{n-1}{n-\frac{k}{v}} \approx 1$. In particular, when $\frac{k}{v} < 1$ and n is sufficiently large, $k' \approx k$ so $R' \approx \left\{\underline{y}' \mid H(f(\underline{y}')) < k\right\}$. Also, the harmonic mean is weighted towards the smallest elements, so when $k < v$ and n is sufficiently large, $H(f(\underline{y}')) \approx H(f(\underline{y}))$ with high probability. Thus, in the first

case, if our basic test has significance α, then our related test will have significance approximately α, indicating that our basic test also has power α.

In the other case, $1 < \frac{k}{v}$. As $\frac{k}{v}$ gets larger, $\frac{n-1}{n-\frac{k}{v}}$ gets larger, leading to larger rejection regions R' associated with these related tests. Larger rejection regions $R' = \left\{\underline{y}' \mid H(f(\underline{y}')) < k'\right\}$ are associated with related tests which more aggressively reject the null hypothesis. These related tests have increased sizes α, and therefore the associated basic tests should have increased power.

Thus, we ought to expect test power to roughly begin at our significance α when $v = f(y^*)$ is large and to increase as $v = f(y^*)$ decreases. In particular, we ought to expect that test power is greatest against the least-likely $y^* \in \mathcal{Y}$ to be sampled under the default wallet distribution f.

4.4. **Parameterization of Extensions.** The extended test uses the test statistic $H\left(\frac{f(y_1)}{\hat{f}(y_1)}, \ldots, \frac{f(y_n)}{\hat{f}(y_n)}\right)$. Finding h_α, then, requires some knowledge of $\hat{f}$, the distribution selected by the user. Given this additional information, parameterization can proceed nearly identically as in the previous sections.

Unfortunately, while $v = f(y)$ may be well-described by Beta distributed random variables, the corresponding value from the extended test $r = \frac{f(y)}{\hat{f}(y)}$ generally cannot be, even if both $\hat{f}(y)$ and $f(y)$ are Beta distributed. However, using ratios of two random variables with the Monte Carlo approaches described in Sections **??** through **??** is very straightforward; our prototype Python code can be easily modified to find these associated critical values.

5. Test-Based Attacks

5.1. **Basic Attack.** Attackers apply one of the tests from the previous section to determine if a transaction has been churned. If they reject the null hypothesis, they conclude that the data is consistent with the notion that we may determine the true signer with better performance than guessing randomly.

Specifically, for each $1 \leq j \leq n$, we let $\mathcal{H}'_j$ be the hypothesis that $\theta = 1$ and $i^* = j$, i.e. there is a true signer at index j. These hypotheses partition the hypothesis $\mathcal{H}_1$ that $\theta = 1$. The distribution of the anonymity set conditioned upon this event is exactly

$$\mathbb{P}[y_1, \ldots, y_n \mid \theta = 1, i^* = j] = \prod_{i \neq j} f(y_i) I(y_j = y^*) \tag{5.1}$$

and therefore we have the j^{th} likelihood function

$$\mathcal{L}((\theta, i^*) = (1, j) \mid \underline{y}) = \prod_{i \neq j} f(y_i) \tag{5.2}$$

and the j^{th} likelihood ratio

$$\Lambda^*_j = \frac{\mathcal{L}((\theta, i^*) = (1, j) \mid \underline{y})}{\max_{1 \leq i \leq n} \mathcal{L}((\theta, i^*) = (1, i) \mid \underline{y})} \tag{5.3}$$

yielding an index ι which maximizes Λ^*_ι. Call this ι the *maximum likelihood estimate* for i^*. With this strategy, an attacker performs the following, beginning by setting $m = 1$.

(1) Apply one of the tests in the previous section to each non-singleton anonymity set on the blockchain, labeling the anonymity sets for whom the null hypothesis was rejected as $\texttt{accused}_m$ and labeling the ones for whom the null hypothesis was accepted as $\texttt{churned}_m$.
(2) For each non-singleton anonymity set $\underline{y}_i = \{y_{i,1}, \ldots, y_{i,n_i}\} \in \texttt{accused}_m$, compute the likelihood ratio $\Lambda_{i,j}$ for each member $y_{i,j}$ of the anonymity set. Store the maximum likelihood ratio $\Lambda_i = \max_j \Lambda_{i,j}$ of this anonymity set, the index $\iota = \text{argmax}_j \Lambda_{i,j}$ corresponding to that maximum, and the cardinality n_i. For the greatest cardinality $n > 1$, compute the grand maximum likelihood ratio $\Lambda^{(n)} = \max_i \Lambda_i$ and the corresponding index $\iota' = \text{argmax}_i \Lambda_i$. Accuse $y_{\iota',\iota}$ of being the true signer of $\underline{y}_{\iota'}$ by doing the following.
   - Strike out each $y_{\iota',j} \neq y_{\iota',\iota}$ from $\left\{y_{\iota',1}, \ldots, y_{\iota',n_{\iota'}}\right\}$.
   - For each anonymity set $\{y_{i,1}, \ldots, y_{i,n}\}$ such that $y_{\iota',\iota}$ is an element and $i \neq \iota'$, strike out $y_{\iota',\iota}$.
(3) Increment $m \leftarrow m + 1$ and go back to step 1.

Following this procedure, each accusation is taken into account in future tests. The sets $\texttt{accused}_m$ are absorbing sets, in the sense that some transactions labeled as $\texttt{churn}_m$ may find themselves in $\texttt{accused}_{m+k}$ for some $k \geq 1$ later, but no elements can exit $\texttt{accused}_m$ for some $\texttt{churn}_{m+k}$. Once the sets satisfy $\texttt{accused}_{m+1} = \texttt{accused}_m$ and $\texttt{churn}_{m+1} = \texttt{churn}_m$, the process can terminate.

Note that we proceed by the anonymity sets with greatest cardinality first, decreasing in cardinality as we go, forcing the algorithm to make accusations based on the most information possible and to eventually terminate. However, this attack is not particularly efficient.

5.2. **Similarity to previous attacks.** With the "guess newest" heuristic, the distribution f was heavily weighted toward old outputs. Any time a user spent a young output, that output had $v = f(y^*) < c$, so the harmonic mean of the probabilities was very small. This allowed the harmonic mean to enter the rejection region more often, and the maximum likelihood estimate of the true spender was, with high probability, the youngest output.

5.3. **Other Related Attacks.** The basic attack can be leveraged in other ways.

5.3.1. *Detecting Periodicity.* Presume an attacker receives money from you and hypothesizes that you churn your transactions exactly k times separated by T day intervals. The attacker can step back through the transaction history of the corresponding outputs and apply the basic churn test. For example, an attacker is given a TPSASA transaction history. The attacker is told that exactly one user is churning their transactions and all other transactions are not churned. The attacker is told the churning user always churns 5 times in between real transactions. The attacker is asked to identify whether the churning user made any such chain of churned transactions within this subset of the transaction history.

The attacker sets $\alpha = 0.05$ and applies the basic test for churn to all transactions. Each churned transaction is independently flagged with probability 0.95, and $(1 - \alpha)^k = 0.95^5 \approx 0.77$, so every chain of 5 churned transactions within this connected subset is flagged as such about 77% of the time, regardless of the amount of time separating them. If no such chain is flagged, the attacker can conclude it is more likely that the target user has been absent.

If many such chains are flagged, the attacker has more work to do. Indeed, if the default wallet distribution is very close to *ad hoc* user behavior (i.e. only one user is churning, but *ad hoc* transactions are still very similar to churned transactions), the test has power around 0.05. Thus, about 95% of *ad hoc* transactions are flagged as churned. For this reason, as default wallet distributions approach user behavior or as sample sizes increase, more and more chains of 5 *ad* hoc transactions will be incorrectly flagged as churned. This is the sense in which we say this test is all but useless if users behave similarly to the default wallet distribution. However, even if many chains of 5 transactions are flagged as churned regardless of their ground-truth classification, it is unlikely that chains have transactions separated by exactly T days each. Thus, despite low test power, the additional temporal information can be more than enough to identify the target user transactions.

Note that if users are behaving even a little differently from the default wallet distribution, then test power increases a little. If test power attains 0.15, then only 85% of *ad hoc* transactions are flagged as churned. Then, since $(1 - 0.85)^5 \approx 0.44$, only about 44% of sequences of 5 *ad hoc* transactions are flagged as being churned, improving test utility significantly.

5.3.2. *Weighted Matching Algorithms.* Instead of the approach in the previous section, attackers can arrange the set of keys $\mathcal{Y}$ as one part in a bipartite graph, with ring signatures as the other part. An edge connects a ring signature in the part with signatures to every key implicated as an anonymity set member. Then every matching in this graph is a plausible transaction history implied by the state of the blockchain. An attacker can use our tests to determine which ring signatures appear to be *ad hoc*, and to assign likelihood ratios as weights to graph edges. Since matching algorithms can be made very efficient, this is approach is likely much more efficient than the naïve attack presented in Section **??**, and with similar performance.

6. Discussion

Users should be advised that unless churn rates on the network as a whole are sufficiently great, the test-based attack presented herein remains a threat to user privacy, even for individual users who churn their own funds on a randomized schedule.

6.1. **Low Sensitivity to Anonymity Set Size.** When if n is sufficiently large, $H(f(\underline{y})) \approx H(f(\underline{y}'))$, so our test performance is resistant to changes in anonymity set sizes. Thus, increasing anonymity set sizes is not a particularly effective solution to defusing our attack. However, our tests are essentially useless if small anonymity sets are replaced with full anonymity sets, as in [5].

6.2. **Effect of Binning.** It is sometimes suggested that anonymity set members are sampled from bins to improve security against tracing. Our basic test can be modified to work under a binning paradigm. In this case, the basic test is somewhat less expensive to parameterize, and no less powerful, indicating that binning will not change how mustard tastes.

6.3. **Violate Assumptions.** Our attack is based on certain assumptions which can be violated. In our work above, we assume f is independent of y^*. This is not valid in general, but our tests can be modified to account for this as described above. We also use the time the transaction was first relayed as a proxy (alternatively, the block height the transaction was mined) to determine the PMF f, which we assume is independent of y^*. This proxy is not generally valid, since users can sign transactions offline some time before broadcasting those transactions on the network.

Thus, users may obtain a small degree of obfuscation by not only separating churned transactions by random wait times, but also waiting a random period of time between computing a transaction and relaying it on the network. Unfortunately, this may have a trade-off in security, because if the user waits too long, then f will evolve. Then, the anonymity set will not be distributed according to g_θ any longer, and y^* may look suspicious. Thus, this random time cannot be too long compared to, say, block arrival times.

6.4. **Avoid "Gaming" Tests and Big Data.** If the rejection region R corresponds to $H(f(\underline{y})) < k$, why not construct an anonymity set such that $H(f(\underline{y})) > k$? Along a similar vein, if these tests single out the anonymity set member which is least likely to be sampled by f, why not wait until $f(y^*)$ is the greatest of all the anonymity set members? In both cases, given a wallet algorithm for constructing anonymity sets in a different way, a LRT like the one described herein exists and can be deployed to test for that behavior. If the behavior is parameterizable under the Neyman-Pearson lemma, then the resulting LRT is a UMP, although such extensions are usually not UMP. Even without a closed-form description of the wallet algorithm distribution, we may empirically estimate test statistics rather than relying on the convenient closed-form description as a harmonic mean, so even if the resulting LRT is complicated, it is far from useless.

That is to say, by trying to "game" our UMP LRT, users open themselves up to similar LRTs, and it is not possible to defend against them all in a protocol with sub-cryptographic levels of security.

Moreover, an entity with access to a lot of data (such as a KYC/AML exchange which has ground-truth linking behavior across a multitude of blockchains, some transparent and some spender-ambiguous) can use the same sorts of techniques to take into account various cases of human behavior, and deploy simultaneous testing procedures to amplify the power of their tests. The degree to which these approaches can be exploited at a large scale given access to additional data is not known precisely, but the threat is known to be highly non-trivial.

6.5. **Too Many People Have to Churn.** Our test efficacy varies with the prevalence of *ad hoc* transactions. In an environment with high prevalence (of *ad hoc* transactions), low power tests are nevertheless useful; positive predictive value can remain high until prevalence drops below a critical threshold. Reducing prevalence, in this context, occurs when a larger proportion of users churn their transactions. As prevalence rises, *ad hoc* transactions are more likely. As this occurs, the positive predictive value of our test rises, despite the low power. The *prevalence threshold* is the point at which the positive predictive value of our test, as a function of

prevalence, has an inflection point, and is changing most rapidly.

$$\frac{\sqrt{(1-\beta)\alpha}-\alpha}{1-\beta-\alpha}$$

The prevalence threshold roughly is the point at which we see reduced rates of return in terms of positive predictive value.

We can loosely think of the prevalence threshold as the maximum allowable proportion of *ad hoc* transactions for the purposes of this discussion. In a previous example, we considered the case that $1-\beta = 0.15$ and $\alpha = 0.05$. For that example, the prevalence threshold is 0.36. The test therefore gains power most quickly as the percentage of *ad hoc* transactions passes 36%. In this scenario, for every *ad hoc* transaction, users might consider making at least $2.\overline{77}$ churn transactions to mitigate our basic test. Of course, if users followed this advice, the majority of the transactions in the TP must be simulated in order for spender-ambiguity to hold, calling into question the utility of the protocol for permissionless e-cash.

7. Conclusion

Our UMP LRT for detecting *ad hoc* transactions in a TPSASA presents a serious threat. That these tests even exist with non-trivial power is a matter of concern, despite that they are expensive to parameterize. The tests are low-cost, can be carried out by small-scale threats, and can be used as a "wedge" by parties with access to lots of additional information to reduce effective anonymity set sizes. Unlike previous attacks, assessing the power of our test is equivalent to assessing the significance of a related test, so we are able to say much about otherwise unknown false negative rates. Moreover, UMP tests are famously low-power if uniformity in test power across the whole parameter space is not necessary. Thus, we can directly modify our approaches to handle more complicated scenarios, often obtaining even more powerful tests (e.g. the case that the default wallet distribution PMF f is dependent on y^*, or the case that we have access to prior models of user behavior). All this emphasizes the urgency with which migrations to full-chain membership proofs should take place.

7.0.1. *Future Work.* Future work may include any of the following.

- Theoretically modeling the PMF f.
- Numerically parameterizing our tests using simulated data from a simulated blockchain.
- Numerically parameterizing our tests using empirical data from a snapshot of an example blockchain.
- Numerically estimating how our test's performance varies as statistical distance between user behavior and default wallet distributions expands.
- Our approaches may fit into (or be falsified by) a more broad context in mathematical statistics, which may be more appropriate for our use-cases, suggesting a deeper investigation into statistical models of anonymity sets.
- Extending the extended attacks to formally take Bayesian updating into account given new ground-truth user information.
- Similarly, a deeper discussion on the threat represented by low power tests in high-prevalence, high-sample-size environments by attackers with lots of additional information about users.

- Our work herein is, in some senses, a generalization of previous work. More detailed descriptions drawing equivalencies between this work and previous work would be appropriate.

References

1. Ayesha Ali, *Scaling privacy perserving payments*, Ph.D. thesis, Massachusetts Institute of Technology, 2024.
2. Vitalik Buterin et al., *Ethereum white paper*, GitHub repository **1** (2013), 22–23.
3. George Casella and Roger Berger, *Statistical inference*, CRC Press, 2024.
4. Alishah Chator, *Practice-oriented privacy in cryptography*, Ph.D. thesis, Johns Hopkins University, 2023.
5. Liam Eagen, *Zero knowledge proofs of elliptic curve inner products from principal divisors and weil reciprocity*, Cryptology ePrint Archive (2022).
6. Daira Hopwood, Sean Bowe, Taylor Hornby, Nathan Wilcox, et al., *Zcash protocol specification*, GitHub: San Francisco, CA, USA **4** (2016), no. 220, 32.
7. 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*, Proceedings of the 2013 conference on Internet measurement conference, 2013, pp. 127–140.
8. Andrew Miller, Malte Möser, Kevin Lee, and Arvind Narayanan, *An empirical analysis of linkability in the Monero blockchain*, arXiv preprint arXiv:1704.04299 (2017).
9. Monero Outreach, *Breaking monero. poisoned outputs (eae attack)*, `https://www.monerooutreach.org/breaking-monero/poisoned-outputs.html`, 2020, Accessed: 2024-10-14.
10. Malte Möser, Kyle Soska, Ethan Heilman, Kevin Lee, Henry Heffan, Shashvat Srivastava, Kyle Hogan, Jason Hennessey, Andrew Miller, Arvind Narayanan, et al., *An empirical analysis of traceability in the Monero blockchain*, arXiv preprint arXiv:1704.04299 (2017).
11. Satoshi Nakamoto, *Bitcoin whitepaper*, URL: https://bitcoin. org/bitcoin. pdf-(: 17.07. 2019) **9** (2008), 15.
12. Jerzy Neyman and Egon Sharpe Pearson, *Ix. on the problem of the most efficient tests of statistical hypotheses*, Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character **231** (1933), no. 694-706, 289–337.
13. Ingolf Gunnar Anton Pernice, Georg Gentzen, and Hermann Elendner, *Cryptocurrencies and the Velocity of Money*, Cryptoeconomic Systems **0** (2021), no. 1, https://cryptoeconomicsystems.pubpub.org/pub/pernice-cryptocurrencies-velocity.
14. 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*, Financial Cryptography and Data Security: FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, April 7, 2017, Revised Selected Papers 21, Springer, 2017, pp. 248–263.
15. Nicolas Van Saberhagen, *Cryptonote v 2.0*, (2013).
16. Hanlin Yang, *Behavioral anomalies in cryptocurrency markets*, Available at SSRN 3174421 (2019).

Cypherstack
Email address: `brandon@cypherstack.com`

Cypherstack
Email address: `rigo@cypherstack.com`

Cypherstack
Current address: Department of Mathematical and Statistical Sciences, Clemson University
Email address: `freeman@cypherstack.com`