Atlas-X Equity Financing: Unlocking New Methods to Securely Obfuscate Axe Inventory Data Based on Differential Privacy

Antigoni Polychroniadou
J.P. Morgan AI Research, AlgoCRYPT CoE
New York, USA
antigoni.polychroniadou@jpmorgan.com

Gabriele Cipriani
J.P. Morgan Quantitative Research
London, United Kingdom
gabriele.cipriani@jpmorgan.com

Richard Hua
J.P. Morgan Quantitative Research
New York, USA
richard.hua@jpmorgan.com

Tucker Balch
J.P. Morgan AI Research
New York, USA
tucker.balch@jpmchase.com

ABSTRACT

Banks publish daily a list of available securities/assets (axe list) to selected clients to help them effectively locate Long (buy) or Short (sell) trades at reduced financing rates. This reduces costs for the bank, as the list aggregates the bank's internal firm inventory per asset for all clients of long as well as short trades. However, this is somewhat problematic: (1) the bank's inventory is revealed; (2) trades of clients who contribute to the aggregated list, particularly those deemed large, are revealed to other clients. Clients conducting sizable trades with the bank and possessing a portion of the aggregated asset exceeding 50% are considered to be concentrated clients. This could potentially reveal a trading concentrated client's activity to their competitors, thus providing an unfair advantage over the market.

Atlas-X Axe Obfuscation, powered by new differential private methods, enables a bank to obfuscate its published axe list on a daily basis while under continual observation, thus maintaining an acceptable inventory Profit and Loss (P&L) cost pertaining to the noisy obfuscated axe list while reducing the clients' trading activity leakage. Our main differential private innovation is a differential private aggregator for streams (time series data) of both positive and negative integers under continual observation.

For the last two years, the Atlas-X system has been live in production across three major regions—USA, Europe, and Asia—at J.P. Morgan, a major financial institution. To our knowledge, it is the first differential privacy solution to be deployed in the financial sector. We also report benchmarks of our algorithm based on (anonymous) real and synthetic data to showcase the quality of our obfuscation and its success in production.

KEYWORDS

Differential privacy; Time-series data; Markets; privacy under continuous observation

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

ACM Reference Format:
Antigoni Polychroniadou, Gabriele Cipriani, Richard Hua, and Tucker Balch. 2024. Atlas-X Equity Financing: Unlocking New Methods to Securely Obfuscate Axe Inventory Data Based on Differential Privacy. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 8 pages.

1 INTRODUCTION

An axe is an interest in a particular security that a firm is looking to buy or sell[1]. In general, a firm providing an axe to external counterparties has a strong interest in keeping such information private as it provides an indication of the direction (buy or sell) they want to trade a particular security. If other market participants are informed of how a particular firm is axed in a given security, they can extract precious information on the firm's trading strategy and, perhaps, could even drive the price of the security in a more disadvantageous direction before the firm can transact.

Large broker-dealer banks, including J.P. Morgan, distribute aggregated axe lists to clients (called hedge funds) with the aim of reducing the costs of running their activities. The axe list is shared electronically (via email or other means) and, most importantly, is common to all clients receiving it. It consists of a list of tuples (symb, direction, quantity) where symb is the symbol of the security/asset to buy or sell based on the direction and quantity is the number of the shares (positions) of the security to trade. The universe of assets covered by the axe list is rather large, encompassing thousands of securities listed in major markets. For a given asset, the bank's axe is given by the aggregation of the positions held by the bank. Importantly, when the traded positions of a large ("concentrated") client contribute the most to the axe quantity, the published axe reveals the trading activity of the client. This is particularly problematic because hedge funds' trading strategies are confidential and their disclosure can undermine the funds' performance. Sensitive trading moves reconstruction is feasible due to various factors, including side information. Clients report positions exceeding specific thresholds to regulatory bodies like the Fed, providing a trail for piecing together trading patterns. Informal conversations in financial circles can also divulge valuable insights. This scenario poses the risk of someone replicating trading moves by observing aggregations, impacting the original strategy's efficiency and market dynamics.

[1]The term comes from the jargon: "having an axe to grind".

The problem we address in this work is how to minimize the adverse effects of the information leakage caused by sharing the axe list with clients. Such undesirable consequences are important both from a reputational point of view, with the bank losing clients which don't want their trading decision made public, as well as a risk management / financial one when the information contained in the axe list is used to trade against the bank itself. Such problems are also exacerbated by the fact that the axe list is published daily, with clients having access to the full history of the published axe lists.

Historically, banks have been using some ad-hoc methods to mitigate the leakage. For instance, they might aggregate several stocks together into buckets (e.g., reveal only range of available stocks to trade in some sector), or trim the positions of other stocks. This does not guarantee privacy and does not provide a useful axe list with good utility (in the case, the profit for the bank). In some cases the clients' positions are removed from the inventory to eliminate the leakage at the expense of poor utility for the bank and the inability of the bank to offer reduced rates to clients.

Our approach is instead more robust, based on a new differential private aggregator for data streams under continual observation. We also introduce precise measures to quantify the utility of the published axe (in terms of profits for the bank) as well as the quality of the obfuscation.

Figure 1: The daily direction (Buy or Sell) of two axe lists that differ only in the positions of a single concentrated client should be statistically indistinguishable.

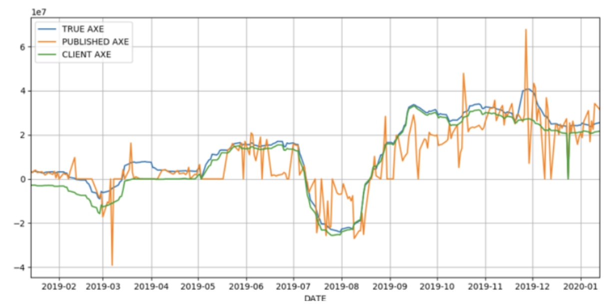

Figure 2: Example of an obfuscated published axe (in orange color) for a given asset, together with the historical data for the bank's true axe (in blue color) and the positions of a highly concentrated client (in green color). The Y-axis refers to the axe quantity while the X-axis the observation date.

2 PROBLEM STATEMENT

In this paper, we investigate an intriguing question related to the secure sharing of aggregated time-series data (e.g. axe inventory trades) on a daily basis of all clients, while preserving the privacy of any changes in the direction (buy or sell) of the data/trades of contributing concentrated clients' trading activity.

How can a bank maintain the continuous release of updated aggregate time-series market data while preserving each individual client's privacy?

As illustrated in Figure 1, by comparing two different axe lists - one that includes the concentrated clients' positions with the bank and one that does not - we would like to obscure whether a single concentrated client is buying or selling an asset on any given day. If the direction is revealed, then the client's activity is exposed, as they are guiding the direction of the asset in question. As depicted in Figure 2, a concentrated client (represented by the green line) holds the majority of positions on the true axe list (represented by the blue line), dictating daily movements. Our objective is to obscure the true axe (represented by the orange line) by concealing the directional activities of the concentrated client, whether it involves buying or selling (increasing or decreasing quantity/ positions). If all the clients follow the same direction as the concentrated client, then this is normal behavior that we should not attempt to hide since the client's behavior is not particularly distinctive and follows the crowd. We would like to maintain privacy with respect to a utility constraint. Jumping ahead, the utility is determined by the profits or losses the bank can incur by obfuscating the direction.

This problem calls for differential privacy. The notion of differential privacy was proposed by Dwork et al. [6]. Since then, there is an extensive work in the literature studying the different tradeoffs between utility and privacy. However, the differentially private setting we consider is different from the traditional setting which assumes a static input database, and a third party that needs to publish some obfuscated/sanitized aggregate statistics of the database once. In our use case, the database is dynamic and changes every day. The differential private mechanism needs to update the published aggregate statistics on a daily basis. Traditional differential private mechanisms can lead to a significant loss in terms of utility or privacy.

In the context of differential privacy we address the following generic question:

How can data aggregators continuously release updated aggregate statistics while preserving each individual user's privacy and without degrading the utility of the data?

The utility measure that we have identified and consider in this work is the cost of keeping positions on the bank's inventory. We describe the P&L as the cost of financing a position on a given asset, due to the need of raising the cash to buy it or acquiring the asset to sell it in the market.

The work of [8] proposes algorithms for differential privacy under continual observation but only for the minimum functionality of counting binary (0 or 1) values at each time-step. The counter statistic is a basic primitive in numerous data streaming algorithms. In this work, we are interested in more complex statistics released under continual observation and explore the utility of the use case at hand.

In this section, for readability reasons we have abstracted and generalized the problem statement based on the use case of axe inventory. For an in-depth connection of the problem statement to the axe inventory use case, please refer to the full version, wherein we have also defined all the financial terms related to the problem.

2.1 Our Contributions

Our contributions can be succinctly summarized as follows:

- **Real-World Use-Case Identification**: We pinpoint a real-world scenario that underscores the potential benefits of DP. Atlas-X is the first DP solution under CO running live in the financial arena.
- **New algorithm for DP under continual observation (CO)**: We introduce a new algorithm on the harder setting of DP with CO; DP under CO presents greater challenges than traditional DP due to the persistent data access. While traditional DP focuses on static datasets. Key Challenges:
  (1) Accumulative Privacy Risk: With ongoing data addition and observations, the cumulative privacy risk grows. New data points incrementally reveal more about individuals, raising the risk of reidentification and sensitive information exposure.
  (2) Cumulative Knowledge: In CO, adversaries might exploit the accumulated knowledge gained from earlier queries to deduce more sensitive information.

  These challenges require more advanced techniques and strategies to ensure long-term privacy preservation while enabling valuable insights from the continuously-observed data.
- **Analysis & Implementation:** Identified the metrics/risks studied for this use case together with the business and provided an analysis based on real production data. Furthermore, we report benchmarks of the Atlas-X system, which runs in production. The system offers increased opportunities for clients to locate trades at advantageous prices as well as achieving better profits for the bank. Atlas-X has also proven to be useful in retaining existing clients of the bank, as we are able to prove that information about their trading activity is effectively safeguarded.
- **Integration with Trading Platform:** Our system seamlessly integrates into an existing trading platform of J.P. Morgan, further validating its practical utility.

Next we provide more details on our new algorithm and the novel use case.

New Algorithm: We show how to address the above questions using differential privacy techniques. We propose a differentially private continual aggregator that outputs at every time step the approximate updated aggregator. We can achieve a construction that has error that is only poly-log in the number of time steps. Assume that the input stream $\sigma \in \mathbb{Z}$ is a sequence of positive and negative integers. The integer $\sigma(t)$ at time $t \in \mathbb{N}$ may denote whether positions/shares in a stock increased or decreased at time t, e.g., whether a client bought or sold σ shares of a stock at time t. The mechanism must output an approximate aggregator of the sum of all positive and negative integers seen so far until timestep t. We propose an ϵ-differentially private continual aggregator with small error. For each $t \in \mathbb{N}$ we guarantee $O(\frac{T^{1/4}\sqrt{\Delta}}{\epsilon})$ error with global sensitivity Δ. See Theorem 6.4 for our formal statement and Theorem 6.5. Prior works [5, 8] have considered simpler statistics under continual observation as well as simpler utility considerations.

Use Case: We have identified a real-world problem for differential privacy under continual observation on a large dataset in which the privacy of the previous axe inventory publication can be significantly enhanced. We propose a new privacy preserving algorithm that generates a noisy axe list while protecting clients' privacy and maintaining the desired profit for the bank (P&L). Differential privacy is a statistical learning tool that enables us to add carefully computed mathematical noise to the axe list. The noise term is large enough to obfuscate individual client positions and small enough to achieve the desired P&L. Our new algorithm is robust with provable guarantees of privacy. To estimate the effectiveness of our method, we have also defined the utility associated with the axe publication as well as measures of the quality of the obfuscation. The model parameters have then been derived by employing these findings, as displayed in Section 7 for further detail.

2.2 Related work

The works of [5, 8] introduced the concept of differential privacy under continual observation and constructed differentially private continual counters of streams of 0's and 1's. Their binary mechanisms are used in the context of the orthogonal problem of privacy-preserving federated learning [4] with the most recent ones being [10] and it represents a separate context from our primary use case where differential privacy under continuous observation is used. Privacy-preserving federated learning is a distributed machine learning approach that allows multiple parties to collaboratively train a shared model while keeping their data private. It is an emerging technology that is gaining popularity due to its ability to protect data privacy and reduce data movement while allowing multiple parties to train a model with their own data. To ensure differential privacy, federated learning employs various clipping mechanisms [1, 2, 13] too. The latest advancements in privacy-preserving federated learning [3, 9, 11, 12] based on secure multiparty computation (MPC) provide enhanced security measures by employing masked or encrypted training gradients.

Prime Match [14] from J.P. Morgan, based on MPC, significantly enhances security and privacy. In Prime Match, buy and sell orders of clients and the bank are encrypted for matching, with orders only being revealed if a match occurs. Unlike Atlas-X based on differential privacy, which hides specific axe dataset/order properties without requiring client participation in MPC, Prime Match ensures no information leakage unless a match is found.

2.3 Technical Overview

Problem Statement. Our goal is to achieve a differential private mechanism for aggregation under continual observation. A mechanism is differentially private if it cannot be used to distinguish two streams that are almost the same. In other words, an attacker cannot tell whether an event of interest took place or not by looking the output of the mechanism over time. For example, the adversary is unable to determine whether a concentrated clients's positions are included in the inventory axe list at some time t.

We abstract the problem as follows: we consider streams of positive and negative numbers. Let $\sigma(t)$ be an item in the stream at time $t \in \mathbb{N}$ which is either a positive or negative integer. At every time t, we wish to output the sum of numbers $\alpha(t) = \sum_i^t \sigma(t)$, the aggregator, that have arrived up to time t from $i = 1$.

Naive mechanism In this mechanism at every time step t, the mechanism answers with a new sum, and randomizes the answer with fresh independent noise i.e. $\alpha(t) + noise$ where $\alpha(t)$ is the true aggregator at timestep t. The drawback is that the privacy loss grows linearly with respect to the number of queries, which is T in our setting. T is an upper bound on time.

Simple mechanism Another approach is to add independent noise to each item of the stream, i.e. $\sigma(t) + noise_t$ and the mechanism outputs $\sum_{i \leq t}(\sigma(t) + noise_t)$ at time t. In this case the privacy loss depends on $\sqrt{T}$.

Window mechanism In this mechanism we want to publish noisy versions of some partial sums as new items arrive. Given the partial sums, an observer computes an estimate for the aggregator at each time step by summing up an appropriate selection of partial sums. For instance, in the naive mechanism, $\alpha(t)$ can be seen as a sum of noisy partial sums where each item $\sigma(t)$ appears in $O(T)$ of these partial sums and this is why the privacy loss is linear in T. In particular, when an item is flipped in the incoming stream, O(T) of the partial sums will be affected. In the simple mechanism, the published partial sums are noisy versions of each item. That said, each item appears in only one partial sum but each aggregator is the sum of $O(T)$ partial sums.

To guarantee small privacy loss, we would like to have each item appear in a small number of partial sums. Moreover, to achieve smaller error we want each aggregator to be a sum of a small number of partial sums since the noises add up as we sum up several noisy partial sums. Inspired by the work of [8] who consider a counter mechanism for counting an incoming stream of only 0s and 1s, we group consecutive items contiguous windows of size B. Then the idea is that within a window, we apply the simple mechanism from above. Then, treating each window as a single item we apply again the simple mechanism. More details of our algorithm are given in Algorithm 1 of our full version. Jumping ahead, each $\alpha(t)$ and $\beta(d(t))$ is a noisy partial sum and one can reconstruct the approximate aggregator at any time step from these noisy partial sums. This Algorithm achieves ϵ-differential privacy and $O(\frac{T^{1/4}\sqrt{\Delta}}{\epsilon})$ error where Δ is our dynamic global sensitivity – which we calculated based on the windows.

Binary mechanism In Algorithm 2 in the full version we show that the error can be reduced to logarithmic in the number of time steps using the so called binary mechanism. The idea is that the grouping of the items depends on the binary representation of the number t. Consider a binary interval tree, a partial sum corresponding to each node in the tree is published. To reconstruct the current aggregator it suffices to find a set of nodes in the tree to uniquely cover the time range from 1 to t. In this case, every time step appears in $O(\log T)$ partial sums and every aggregator can be represented with a set of $O(\log T)$ nodes.

Challenges for the Axe Inventory Obfuscation In Section 5 we carefully and formally define our obfuscation metrics for our use case. As it is described in Section 5.1, our Axe obfuscation algorithms should aim at publishing noisy obfuscated axes that are not too different from the true one, as failure to do so can cause a P&L loss for the bank. However, at the same time we define the leakage probability in Section 5.2 which is a metric to indicate whether increased or decreased positions to an asset are not observable in the published noisy axe.

3 PRELIMINARIES

3.1 Differential Privacy

Differential privacy [6] states that if there are two databases that differ by only one element, they are statistically indistinguishable from each other. In particular, if an observer cannot tell whether the element is in the dataset or not, she will not be able to determine anything else about the element either.

Definition 3.1. *(ϵ-differential privacy [7]) For any two neighboring datasets $\mathcal{D}_1 \sim \mathcal{D}_2$ that differ by one element, a randomized mechanism $\mathcal{A}\colon \mathcal{D} \to O$ preserves ϵ-differential privacy (ϵ-DP) when there exists $\epsilon > 0$ such that,*

$$Pr\left[\mathcal{A}(D_1) \in \mathcal{T}\right] \leq e^{\epsilon}\; Pr\left[\mathcal{A}(D_2) \in \mathcal{T}\right] \tag{1}$$

holds for every subset $\mathcal{T} \subseteq O$, where $\mathcal{D}$ is a dataset, $\mathcal{T}$ is the response set, and O depicts the set of all outcomes.

The value ϵ is used to determine how strict the privacy is. A smaller ϵ gives better privacy but worse accuracy. Depending on the application ϵ should be chosen to strike a balance between accuracy and privacy.

Definition 3.2. *(Global Sensitivity [7]) For a real-valued query function $q : \mathcal{D} \to \mathbb{R}$, where $\mathcal{D}$ denotes the set of all possible datasets, the global sensitivity of q, denoted by Δ, is defined as*

$$\Delta = \max_{\mathcal{D}_1 \sim \mathcal{D}_2} |q(\mathcal{D}_1) - q(\mathcal{D}_2)|, \tag{2}$$

for all $\mathcal{D}_1 \in \mathcal{D}$ and $\mathcal{D}_2 \in \mathcal{D}$.

In our algorithms, the noise may not come from a single Laplace distribution, but rather is the sum of multiple independent Laplace distributions. The sum of independent Laplace distributions maintains differential privacy [5, 6].

Corollary 3.3. *Suppose θ_i's are independent random variables, where each θ_i has Laplace distribution $Lap(b_i)$ and suppose $Y = \sum_i \theta_i$ for $i \in [t]$. The quantity $|Y|$ is at most $O(\sqrt{\sum_i b_i^2}\log\frac{1}{\delta})$. We use the following property of the sum of independent Laplace distributions.*

4 FINANCIAL CONCEPTS

A concise summary of the axe inventory use case is that the publishing bank aggregates its internal firm inventory of long (buy) and short (sell) trades and then provides these offerings to its customers in order to equalize their long and short aggregated positions with regard to a given asset. In the full version we introduce the financial concepts and jargon used for our use case for the interested readers. In particular, we define the profit and losses incurred by "long" and "short" positions (which refer to buying and selling respectively, and are defined in detail below), highlighting the importance of hedging costs for the the bank ("funding" and "borrow" rates). We then demonstrate how banks reduce their hedging costs via a process known as "internalization" and how we can reduce such costs by enticing clients to trade via axe lists. Lastly, we describe the implications of sharing axe lists among clients and how axe lists can leak information about the trading activity of clients with large ("concentrated") positions.

5 OBFUSCATION METRICS

In this section, we present the obfuscation metrics used to measure the quality of our differentially private method.

5.1 P&L

Here we describe an approximation for the daily inventory P&L realized when an axe trade is executed with a client. It should be noted, as before, that while our approximation discards some aspects of the netting process and other costs, it is anyway a faithful representation of the true P&L impact.

The borrow/funding P&L accrued over one day when keeping a net quantity $x(t)$ of a given asset on balance sheet can be summarized as follows:

$$PL^{INV}(t) = \begin{cases} -r_F(t)x(t)P(t) & \text{if } x(t) \geq 0 \\ r_B(t)x(t)P(t) & \text{if } x(t) < 0 \end{cases} \quad (3)$$

where $P(t)$ is the asset's price, $x(t)$ the net inventory positions and r_F / r_B the funding / borrow rates, respectively. From now on we'll restrict our analysis to such P&L contributions, indicating them as "inventory P&L" or, simply, P&L. Equation 3 is obtained assuming that all risky P&L contributions are perfectly hedged and setting $t_E - t_S = 1$ (see full version for details). It should be noted, nevertheless, that the total P&L accrued by the bank is affected by other factors like, for instance, the P&L deriving from fees charged to clients or other considerations.

When a client accepts an axe trade for a quantity $a_{HIT}(t)$, the net inventory changes from $x(t)$ to $x(t) + a_{HIT}(t)$ and the trade's marginal inventory P&L, which is the P&L with the axe trade minus the P&L without, for the bank reads:

$$\begin{aligned} \Delta PL^{AXE}(t) &= PL^{INV}\big(a_{HIT}(t) + x(t)\big) \\ &- PL^{INV}\big(x(t)\big) \\ &= PL^{INV}\big(a_{HIT}(t) - a_{TRUE}(t)\big) \\ &- PL^{INV}\big(-a_{TRUE}(t)\big) \end{aligned} \quad (4)$$

where $a_{TRUE}(t) = -x(t)$ is the "true" axe of the positions in inventory. The marginal P&L measures the effect of a trade on the bank's inventory P&L: a large positive marginal P&L is associated with a good trade from a P&L perspective. By the same token, trades with negative marginal P&L would cause a P&L loss for the bank if executed.

The behaviour of the marginal P&L as a function of the traded axe quantity $a_{HIT}(t)$ is rather intuitive, see Fig. 3:

- The maximum is achieved when the traded axe quantity is equal to the true one, $a_{HIT}(t) = a_{TRUE}(t)$. That corresponds to the perfect case in which the axe trade fully consumes the balance sheet, flattening it to zero, hence removing all inventory costs.
- When the traded axe $a_{HIT}(t)$ is larger then the true axe $a_{TRUE}(t)$, the marginal P&L decreases at the funding rate $r_F(t)$. That is because any additional axe quantity will make the net inventory longer, hence increasing the funding costs.
- Otherwise, the marginal P&L increases at the borrow rate $r_B(t)$. Any smaller traded axe quantity makes the net inventory shorter and increases the borrowing costs.

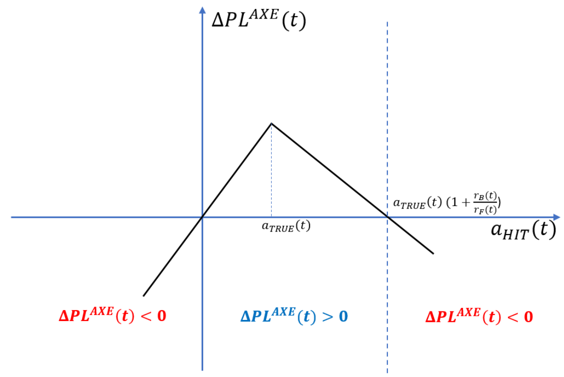

Figure 3: Marginal P&L profile (Y-axis) of a long axe trade as a function of the axe quantity traded by a client (X-axis).

It should be noted that an axe trade generates a profit only when its quantity stays close enough to the true axe quantity:

$$a_{HIT}(t) \in \begin{cases} \left[0, a_{TRUE}(t)\left(1 + \frac{r_B(t)}{r_F(t)}\right)\right] & \text{if } a_{TRUE}(t) \geq 0 \\ \left[a_{TRUE}(t)\left(1 + \frac{r_F(t)}{r_B(t)}\right), 0\right] & \text{if } a_{TRUE}(t) < 0 \end{cases} \quad (5)$$

The intervals above identify the values of $a_{HIT}(t)$ making the marginal P&L $\Delta PL^{AXE}(t)$ positive. As a consequence, axe obfuscation algorithms should aim at publishing obfuscated axes that are not too different from the true one, as failure to do so can cause P&L losses.

5.2 Leakage Probability

We define the Leakage Probability as the probability of correctly guessing a client trading direction (i.e. whether a given fund is increasing or decreasing its positions on given assets) using the direction of change of the published axe, i.e.:

$$\begin{aligned} LP(t) = Prob\Big[&\operatorname{sgn}\Big(\big(p(t) - p(t-\tau)\big) \\ &\big(a_{PUB}(t) - a_{PUB}(t-\tau)\big)\Big) < 0\Big] \end{aligned} \quad (6)$$

where $a_{PUB}(t)$ is the published obfuscated axe list, $p(t)$ is the client's position in the asset and τ a time lag (a few days, typically). Please notice that when the client increases their positions ($p(t) - p(t-\tau) > 0$) the effect on the axe is the opposite ($a_{PUB}(t) - a_{PUB}(t-\tau) < 0$), and vice versa hence the definition above.

From a practical point of view, both the direction and quantity of the change in the published axe are important. Our definition of Leakage Probability keeps into account only the direction because it is meant to be a simple and "robust" estimator of the information leaked by the published axe. Any attacker able to detect both the sign and the quantity of change in the true axe will also be able to infer the sign only. A high Leakage Probability denotes a situation in which an attacker could understand whether the bank (or a concentrated client) are taking new positions on. A low Leakage Probability means, instead, that the bank trading decisions (whether they have been increasing or decreasing their exposure to an asset) are not observable in the published axe.

We also define the over-axe frequency and worst case cost in the full version. The over-axe measures how often the published axe, if fully accepted by clients, would cause a negative inventory P&L / loss for the bank

6 AXE OBFUSCATION VIA CONTINUAL AGGREGATOR DP MECHANISM

We consider streams of positive and negative numbers. $\sigma(t)$ at time $t \in \mathbb{N}$ denotes a positive or a negative integer. At every t, we wish to output the sum of numbers that have arrived up to time t.

Definition 6.1. *(Continual Aggregator) Given a stream σ of positive and negative integers, let $\sigma(t) \in \mathbb{Z}$ be an integer at time step $t \in \mathbb{N}$ and let:*

$$\begin{aligned}\sigma^+(t) &= \sigma(t) \text{ if } \sigma(t) \geq 0 \text{ else } 0,\\ \sigma^-(t) &= \sigma(t) \text{ if } \sigma(t) \leq 0 \text{ else } 0,\end{aligned}$$

the aggregator for the stream is a mapping $\alpha : \mathbb{Z} \to \mathbb{Z}$ such that for each $t \in \mathbb{N}$, $\alpha(t) := \sum_{i=1}^{t} \sigma^+(i) + \sum_{i=1}^{t} \sigma^-(i)$.

Next, we define the notion of a continual aggregator mechanism which continually outputs the sum of integers seen thus far.

Definition 6.2. *(Continual Aggregator Mechanism) A counting mechanism M takes a stream σ of integers in $\mathbb{Z}$ and produces a (randomized) mapping $M(\sigma) : \mathbb{Z} \to \mathbb{Z}$. Moreover, for all $t \in \mathbb{N}$, $M(\sigma)(t)$ at timestep t is independent of all $\sigma(i)$'s for $i > t$.*

Definition 6.3. *(Utility) An aggregator mechanism M is (λ, δ)-useful at time t, if for any stream σ, with probability (over the randomness of M) at least $1 - \delta$, we have $|\alpha(t) - M(\sigma)(t)| \leq \lambda$. Note that λ may be a function of δ and t.*

The above definition covers the usefulness of the mechanism for a single time step. A standard union bound argument can be used for multiple time steps.

Sensitivity: By using clipping to bound the sensitivity of our summation queries, we are able to enforce upper max and lower min bounds on the positions. This ensures that all positions will be below the upper bound, and the resulting sensitivity of a summation query is equal to the difference between the upper and lower bounds used in clipping, max − min over a period of time T, denoted as either $\max_T - \min_T$ or, for brevity, Δ in the rest of the paper. We do not choose our clipping bounds by looking at the data; instead, we calculate them by exploiting a property of the dataset that can be determined without viewing the data, thereby providing us with prior knowledge about the scale of the data for clipping. The property refers to the Average Daily Trading Volume (ADTV) of each stock which helps us determine the bounds. ADTV is the average number of shares traded within a day for a given stock, calculated by taking the total number of shares traded over a period of time and dividing it by the number of days in that period. As a rule of thumb in our use case the daily added positions of a client is never above the ADTV (our max) and this is public knowledge, i.e. the bank forbids trades exceeding the Average Daily Trading Volume (ADTV). It is forbidden by the bank since the bank does not want to take any risk executing such large daily trades with (concentrated) clients. We would like to note that ADTV can help confirm trends and patterns to market participants, which is public information that we do not seek to hide.

Our Window Algorithm: We concentrate on obfuscating the already clipped True Axe's changes over 1-day periods. Given a time-grid $\{t = 0 \dots T-1\}$, define the True Axe differences as:

$$\sigma(t) = a_{TRUE}(t) - a_{TRUE}(t-1)$$

Then split them into positive and negative parts:

$$\begin{aligned}\sigma^+(t) &= \sigma(t) \text{ if } \sigma(t) \geq 0 \text{ else } 0\\ \sigma^-(t) &= \sigma(t) \text{ if } \sigma(t) \leq 0 \text{ else } 0\end{aligned}$$

We can reconstruct the True Axe as:

$$a_{TRUE}(t) = a_{TRUE}(0) + \sum_{i=1}^{t} \sigma^+(i) + \sum_{i=1}^{t} \sigma^-(i)$$

Perturbations are applied to the Axe differences ($\theta^+(t)$ and $\theta^-(t)$ are random shocks we describe in a moment):

$$\begin{aligned}\alpha^+(t) &= \sigma^+(t) + \theta^+(t)\\ \alpha^-(t) &= \sigma^-(t) + \theta^-(t)\end{aligned}$$

The Published Axe is eventually given by:

$$\begin{aligned}a_{PUB}(t) &= a_{TRUE}(0) + \sum_{i=1}^{t} \alpha^+(i) + \sum_{i=1}^{t} \alpha^-(i)\\ &= a_{TRUE}(t) + \Theta^+(t) + \Theta^-(t)\end{aligned}$$

where $\Theta^+(t)$,$\Theta^+(t)$ are the cumulative shocks.

To get better efficiency we also split the time-grid into buckets, each long B days:

$$\begin{aligned}t &= d(t)B + c(t)\\ c(t) &= t \pmod B\\ d(t) &= \frac{t - c(t)}{B}\end{aligned}$$

and define the cumulative shocks $\Theta^+(t)$,$\Theta^+(t)$ as follows:

$$\begin{aligned}\Theta^+(t) &= \Theta^+\big(p(t)\big) + \theta^+(t)\\ \Theta^-(t) &= \Theta^-\big(p(t)\big) + \theta^-(t)\end{aligned}$$

where $T > B$ is a reset period and $p(t)$ is the start of the current B-bucket:

$$p(t) = d(t)B$$

The random shocks inside a T-period are given by sensitivity which is the difference between maximum and minimum change of the True Axe:

$$\theta^{+}(t) \sim \text{Lap}\left(\frac{\left|\max_{i\in[p(t),t]} - \min_{i\in[p(t),t]}\right|}{\epsilon}\right)$$

$$\theta^{-}(t) \sim \text{Lap}\left(\frac{\left|\max_{i\in[p(t),t]} - \min_{i\in[p(t),t]}\right|}{\epsilon}\right)$$

See more details in Algorithm 1 of the full version.

Theorem 6.4. *Let $0 < \delta < 1$ and $\epsilon > 0$. The continual aggregator mechanism is 2ϵ-differentially private. Furthermore, for each $t \in \mathbb{N}$, the mechanism with block size B is $(O(\frac{1}{\epsilon} \cdot \sqrt{\Delta \cdot (T/B + B)} \cdot \log \frac{1}{\delta}), \delta)$-useful at time t out of the T time steps.*

Proof. We will use the term item to refer to an integer in the stream σ. We let $\sum_{k=I}^{j} \sigma(k)$ to denote a partial sum involving items i through item j. We start by observing that each item $\sigma(t)$ in the published stream $\alpha_{PUB}(t)$ appears in at most two noisy partial sums: at most one where $c(t) \neq 0$ and at most one in the case where $c(t) = 0$. In particular, let $d(t) = \frac{t}{B}$, then $\sigma(t)$ appears in only $\beta(d(t))$ and $\alpha(t)$. That said, the aggregator mechanism preserves 2ϵ-differential privacy. This means that if we change $\sigma(t)$, at most 2 noisy partial sums will be affected.

Next we focus on the utility: Observe that at any time $t = d(t)B + c(t)$ where $d(t), c(t) \in \mathbb{Z}$ and $0 \leq c(t) < B$, the error of $\alpha_{PUB}(t)$ includes the sum of $K = d(t) + c(t)$ independent Laplacian distributions $Lap(\Delta/\epsilon)$. The estimated aggregator at any time t is the sum of at most $\lfloor t/B \rfloor + B$ partial sums. Followed from Corollary 3.3, since $t/B \leq K \leq (t/B + B)$, at time T, the Aggregator Mechanism is $(O(\frac{1}{\epsilon} \cdot \sqrt{\Delta \cdot (T/B + B)} \cdot \log \frac{1}{\delta}), \delta)$-useful at time t . □

Suppose a mechanism M adds $Lap(\Delta/\epsilon)$ noise to every partial sum before releasing it. In M, each item in the stream appears in at most x partial sums, and each estimated aggregator is the sum of at most y partial sums. Then, the mechanism M achieves $x\epsilon$-differential privacy. Moreover from Corollary 3.3 the error is $O(\frac{\sqrt{\Delta \cdot y}}{\epsilon})$ with high probability. Alternatively, to achieve ϵ-differential privacy, one can scale appropriately by having $\epsilon' = \epsilon/x$. Now if the mechanism instead adds $Lap(\Delta/\epsilon')$ noise to each partial sum we achieve ϵ-differential privacy, and $O(\frac{x\sqrt{\Delta \cdot y}}{\epsilon})$ error with high probability.

If we let $B = \sqrt{T}$, then the estimated aggregator at any time is the sum of at most $2B$ noisy partial sums. The error is roughly $O(\frac{T^{1/4}\sqrt{\Delta}}{\epsilon})$ with high probability.

6.1 Binary Mechanism with Less Error

In the full version we present a version of our mechanism which incurs a smaller error. The estimated aggregator of the algorithm is the sum of at most $\log T$ noisy partial sums and each partial sum has an independent Laplace noise of $Lap(\frac{\log T \cdot \Delta}{\epsilon})$. That said, from Corollary 3.3 we get the following:

Theorem 6.5 (Utility). *The continual aggregator mechanism is $(O(\frac{1}{\epsilon} \cdot (\log T) \cdot \sqrt{\Delta \cdot \log t} \cdot \log \frac{1}{\delta}), \delta)$-useful at time t out of the T time steps.*

7 ATLAS-X PERFORMANCE

We have implemented our differentially private continual Aggregator mechanism in the production (in Python code) of the internal brokerage platform. The production system operates on a Linux machine based on python 3.7 (256GB memory). The algorithm runs daily to obfuscate the bank's axe list for three different regions (USA, Europe and Asia). The obfuscated axe is then sent to roughly 60 selected clients (hedge funds), which all receive the same axe list. The rest of this section describes the results of several experiments performed with the monte carlo simulation engine described in the full version based on real inventory data.

We first turn to the important question of how to determine the obfuscation model parameters discussed in Section 6. To do that, we have, instead, run several monte carlo simulations (using the methodology described in the full version) for a grid of obfuscation parameters $\{(\epsilon, T, B)\}$. We have then measured the obfuscation statistics described in Section 5, namely the Expected P&L / Leakage Probability and discussed the results with the trading desk who were took the final decision (in the supplementary material we also measure the Over Axe Frequency and Worst Case Cost). Moreover, given that our application revolves around high-frequency trading, signals spanning a few days pose a significant threat if revealed. We have judiciously chosen a time parameter of T=30 days.

Our analysis was based on the selection of the most concentrated client, whose positions were effectively driving our axe on many assets. In particular, Fig. 4 shows the expected inventory P&L difference between the case in which the bank publishes the DP obfuscated axe including the most concentrated client versus those calculated with the true (un-obfuscated) axe. As expected the results, calculated for different privacy budgets as well as obfuscation parameters, show that publishing the true axe is always advantageous from a P&L perspective. The expected loss with the DP budget chosen in production ($\epsilon = 0.3$) is roughly 1$ (daily, for each asset we publish an axe for). The P&L difference increases with ϵ and flattens for larger values of ϵ.

Fig. 5, instead, compares the expected P&L difference between publishing the obfuscated axe with and without the concentrated client, respectively, using the same model parameters. The results, show that there is an average 4.4$ P&L increase when including the concentrated client (per asset, per day) using the DP budget chosen in production. Again, the P&L difference increases with ϵ and flattens for larger values.

The *P&L* estimates above were then compared with the expected Leakage Probability using the same model parameters. Fig. 6 shows the difference in the estimated Leakage Probability between publishing the obfuscated axe with and without the most concentrated client for time lag of 1-week. The results show that, for the DP budget chosen in production, there is an increase of $\sim 3\%$ in Leak Probability when including the most concentrated client for 1-week lag. That means that only in 3% of the cases our published axe leaks information about the trading activity (direction) of the concentrated client over a 1-week or 2-weeks lags as we show in the

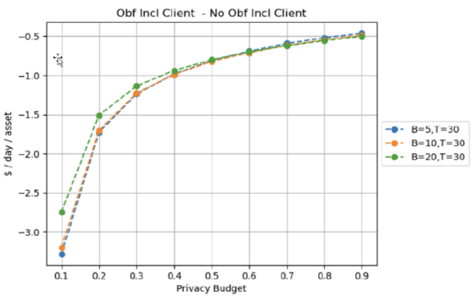

Figure 4: Expected inventory P&L difference (Y-axis) between the case in which the bank publishes the DP obfuscated axe including the most concentrated client versus those calculated with the true (un-obfuscated) axe, measured in dollar per day per asset and calculated for different privacy budgets ϵ (X-axis) as well as obfuscation parameters.

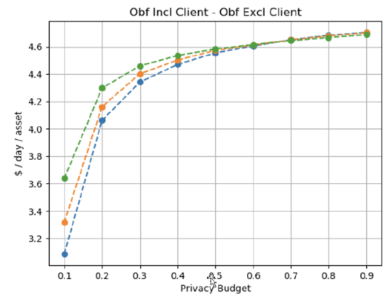

Figure 5: Expected inventory P&L difference (Y-axis) between publishing the obfuscated axe with and without the concentrated client, respectively, measured in dollar per day per asset and calculated for different privacy budgets ϵ (X-axis) as well as obfuscation parameters.

Figure 6: Expected Leak Probability difference (Y-axis) between publishing the DP-obfuscated axe including the most concentrated client versus those excluding it, calculated for different privacy budgets ϵ (X-axis) and with a lag of 1 week.

full version. This was considered as acceptable by the trading desk and the final model parameters used in production, corresponding to ($\epsilon = 0.3, T = 30, B = 20$). Due to space constraints, we direct the reader to the full version for more experiments and diverse scenarios.

ACKNOWLEDGMENTS

This paper was prepared in part for information purposes by the Artificial Intelligence Research group and AlgoCRYPT CoE of JPMorgan Chase & Co and its affiliates (“JP Morgan”), and is not a product of the Research Department of JP Morgan. JP Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

REFERENCES

[1] Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi (Eds.). ACM, 308–318.

[2] Galen Andrew, Om Thakkar, Brendan McMahan, and Swaroop Ramaswamy. 2021. Differentially Private Learning with Adaptive Clipping. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (Eds.). 17455–17466.

[3] James Bell, Adrià Gascón, Tancrède Lepoint, Baiyu Li, Sarah Meiklejohn, Mariana Raykova, and Cathie Yun. 2023. ACORN: Input Validation for Secure Aggregation. In *32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023*, Joseph A. Calandrino and Carmela Troncoso (Eds.). USENIX Association, 4805–4822. https://www.usenix.org/conference/usenixsecurity23/presentation/bell

[4] Kallista A. Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, Bhavani Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu (Eds.). ACM, 1175–1191.

[5] T.-H. Hubert Chan, Elaine Shi, and Dawn Song. 2011. Private and Continual Release of Statistics. *ACM Trans. Inf. Syst. Secur.* 14, 3 (2011), 26:1–26:24.

[6] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. 2006. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 486–503.

[7] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*. 265–284.

[8] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. 2010. Differential privacy under continual observation. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, Leonard J. Schulman (Ed.). ACM, 715–724.

[9] Yue Guo, Antigoni Polychroniadou, Elaine Shi, David Byrd, and Tucker Balch. 2022. MicroFedML: Privacy Preserving Federated Learning for Small Weights. *IACR Cryptol. ePrint Arch.* (2022), 714. https://eprint.iacr.org/2022/714

[10] Peter Kairouz, Brendan McMahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. 2021. Practical and Private (Deep) Learning Without Sampling or Shuffling. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021 (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 5213–5225.

[11] Hanjun Li, Huijia Lin, Antigoni Polychroniadou, and Stefano Tessaro. 2023. LERNA: Secure Single-Server Aggregation via Key-Homomorphic Masking. In *Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4-8, 2023, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 14438)*, Jian Guo and Ron Steinfeld (Eds.). Springer, 302–334.

[12] Yiping Ma, Jess Woods, Sebastian Angel, Antigoni Polychroniadou, and Tal Rabin. 2023. Flamingo: Multi-Round Single-Server Secure Aggregation with Applications to Private Federated Learning. In *44th IEEE Symposium on Security and Privacy, SP 2023, San Francisco, CA, USA, May 21-25, 2023*. IEEE, 477–496.

[13] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2018. Learning Differentially Private Recurrent Language Models. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

[14] Antigoni Polychroniadou, Gilad Asharov, Benjamin E. Diamond, Tucker Balch, Hans Buehler, Richard Hua, Suwen Gu, Greg Gimler, and Manuela Veloso. 2023. Prime Match: A Privacy-Preserving Inventory Matching System. In *32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023*, Joseph A. Calandrino and Carmela Troncoso (Eds.). USENIX Association, 6417–6434. https://www.usenix.org/conference/usenixsecurity23/presentation/polychroniadou