# Understanding Information Disclosure from Secure Computation Output: A Study of Average Salary Computation

Alessandro Baccarini
University at Buffalo
Buffalo, New York, USA
anbaccar@buffalo.edu

Marina Blanton
University at Buffalo
Buffalo, New York, USA
mblanton@buffalo.edu

Shaofeng Zou
University at Buffalo
Buffalo, New York, USA
szou3@buffalo.edu

## ABSTRACT

Secure multi-party computation has seen substantial performance improvements in recent years and is being increasingly used in commercial products. While a significant amount of work was dedicated to improving its efficiency under standard security models, the threat models do not account for information leakage from the output of secure function evaluation. Quantifying information disclosure about private inputs from observing the function outcome is the subject of this work. Motivated by the City of Boston gender pay gap studies, in this work we focus on the computation of the average of salaries and quantify information disclosure about private inputs of one or more participants (the target) to an adversary via information-theoretic techniques. We study a number of distributions including log-normal, which is typically used for modeling salaries. We consequently evaluate information disclosure after repeated evaluation of the average function on overlapping inputs, as was done in the Boston gender pay study that ran multiple times, and provide recommendations for using the sum and average functions in secure computation applications. Our goal is to develop mechanisms that lower information disclosure about participants' inputs to a desired level and provide guidelines for setting up real-world secure evaluation of this function.

## CCS CONCEPTS

• **Security and privacy → Information-theoretic techniques**; *Information flow control*; • **Mathematics of computing → Information theory**.

## KEYWORDS

secure function evaluation, information disclosure, entropy, average salary computation

**ACM Reference Format:**
Alessandro Baccarini, Marina Blanton, and Shaofeng Zou. 2024. Understanding Information Disclosure from Secure Computation Output: A Study of Average Salary Computation. In *Proceedings of the Fourteenth ACM Conference on Data and Application Security and Privacy (CODASPY '24), June 19–21, 2024, Porto, Portugal.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3626232.3653280

## 1 INTRODUCTION

Secure multi-party computation and other forms of computing on cryptographically protected data (such as homomorphic encryption) open up possibilities for great utilization and analysis of private data distributed across different domains, which otherwise might not be feasible due to the sensitive nature of the data. For example, analysis of health-related records and medical images distributed across different medical facilities and extracting cues from them lead to medical advances without the need to see the records themselves. Today, data analysis practices by researchers are hindered by laws regulating access to health data in different countries and analyzing medical data at scale presents challenges. Similarly, analyzing sensitive data such as salaries to understand disparities by gender, race, or other types of marginalization can supply decision makers with important information and empower them to address the disparities. This was the case with the Boston area gender pay gap surveys [13, 14, 37–39] that initiated in 2015 and ran through 2017 with more participants and data analysis by additional categories including race. More broadly, wider adoption of privacy-preserving technologies, and secure computation in particular, can lead to higher security standards and practices for a broad range of different aspects of our society.

Secure computation techniques have seen significant advances in recent decades in terms of their speed, as well as availability of implementations and tools to facilitate their use for a variety of applications. Tech giants such as Google and Apple started using secure computation techniques in their products [12, 32, 36, 55] and the number of start-up companies offering related products is growing (see, e.g., [31, 40, 45, 46]). However, a number of fundamental questions still need to be addressed by the research community in order to make secure computing practices common place.

One of the fundamental questions is how much information about a participant's private input(s) might be available as a result of evaluating a desired function on private inputs. Standard security definitions adopted in the cryptographic community require that no information about private inputs is disclosed during function evaluation. That is, given a function $f$ that we evaluate on private inputs $x_1, x_2, \ldots$ coming from different sources, security is achieved if a participant does not learn more information than the function output and any information that can be deduced from the output and its private input. However, there are no constraints on types of functions that can be evaluated in this framework, and thus the information a participant can deduce from the output and its private input about another participant's private input is potentially large. This problem is typically handled by assuming that the function being evaluated is agreed upon by and acceptable to the data owners as not to reveal too much information about private

*CODASPY '24, June 19–21, 2024, Porto, Portugal.*

ACM ISBN 979-8-4007-0421-5/24/06
https://doi.org/10.1145/3626232.3653280

inputs. However, our ability to evaluate functions in this aspect and determine what functions might be acceptable is currently limited. This question is the subject of this work.

Intuitively, what we want is to guarantee that the function being evaluated on private data is non-invertible, i.e., observing the output does not reveal its private input. Cryptography uses the notion of one-way functions – and assumes this property for hash functions – to model non-invertibility. However, what is needed in this case is to ensure that the possible space for a target private input is still large after the adversary observes the result of function evaluation. This notion of non-invertibility was first used in the context of secure multi-party computation in solutions for business applications such as supply chain management and component procurement [24–26] and was formulated as the inability to narrow down the (private) input of another party to a single value or a small set of possible values. Consequently, a series of publications by Ah-Fat and Huth [1–4] put forward formal definitions that use entropy to measure the amount of uncertainty about one or more participants' private inputs after using them in secure multi-party computation. The definitions are framed from a) an attacker's perspective who aims to maximize information disclosure of a target's private input and b) from a target's perspective who determines the maximum information disclosure about their inputs when deciding whether to contribute their inputs to secure evaluation of a particular function. The above formulations are general and applicable to any function, while application-specific formulations of what constitutes sufficient input protection and function non-invertibility also emerged. One example is building machine learning models resilient to membership inference attacks [50, 53] that guarantee that it is infeasible to determine whether someone's data was used for training the model.

**Our contributions.** In this work, we use the entropy-based definitions from [1] as our starting point and analyze a specific function of significant practical relevance. In particular, we focus on the case of average salary computation as used in the Boston gender pay gap study [38]. When the total number of inputs is known (which is typically the case), the average computation is equivalent to computing the sum. We intuitively understand that the larger the number of inputs used in the computation of the average is, the better protection each individual contributing its input obtains. In the extreme case of two participants[1] no protection can be achieved. This was understood by the designers of the Boston gender pay gap study who recommended running the computation with at least 5 contributors [39]. However, the information disclosure was not quantified, which we remedy in this work.

We start by analyzing the function itself and formally show that the amount of information an attacker learns is independent of his/her own inputs. This is consistent with our intuition that, given a sum, one can always remove their contribution to the sum and analyze the resulting value. Thus, the protection depends on the number of spectators, i.e., input parties distinct from those controlled by the adversary and the party or parties being targeted.

We analyze the target's input entropy remaining after participating in the computation (and consequently the entropy loss as a result of participation) for a number of discrete and continuous distributions including uniform, Poisson, normal (Gaussian), and log-normal. Log-normal is typically used for modeling salary data, but is the least trivial to analyze. An unexpected finding of our analysis is that for a given distribution, the absolute entropy loss is normally independent of the distribution parameters and the absolute entropy loss remains very close for different distributions as we vary the number of participants/spectators. Quantifying the information loss allows us to devise a mechanism to lower information disclosure to any desired level (e.g., 1% of original entropy, 0.05 bits of entropy, etc.).

We extend our analysis of information loss to the case when the computation is run more than once (as was the case for the Boston gender pay gap study) and examine the case with two evaluations. This corresponds to (i) the target participating in two computations with the same input where the set of participants differs between the executions and (ii) the target participating in one computation, where the other is run without the target's input. We observe that information loss increases as a result of multiple computations, regardless of whether the target participates once or twice. Furthermore, the protection is maximized when one half of the original contributors are replaced, i.e., 50% of the initial participants remain and the other 50% are replaced with new participants. Our multi-execution analysis is based on the normal distribution, but we expect the outcome to be similar for other distributions as well.

We provide additional proofs and generalize our analysis to three and more executions in the full version of the text [8]. An interesting finding is that the best configuration that minimizes information loss is determined by pairwise overlaps of participants between the executions and not by other parameters and sizes. This allows us to determine optimal setup for a single and repeated execution of the average function.

We empirically validate our findings throughout this work and provide recommendations for securely evaluating the average function in real world applications. In particular, in all of our experiments the cost of participating in the average computation, i.e., the difference in the entropy before and after the computation is a fraction of a bit (for both Shannon entropy used with discrete distributions and differential entropy used with continuous distributions). This translates to small relative entropy loss in practice. When modeling salary data using log-normal distribution with the parameters specifically chosen for salaries [17], the entropy loss is below 5% with at least 5 non-adversarial participants or spectators and achieving 1% entropy loss requires 24 spectators. These numbers are also surprisingly similar across different distributions. Furthermore, when the computation is repeated (we use a normal distribution to adequately approximate the log-normal setup), engaging in the computation the second time with an overlapping set of 50% participants whose inputs do not change results in only 30% entropy loss of the first participation. These and other findings lead to a number of recommendations for running this computation in practice, which we provide at the end of this work.

**On the choice of metric.** Our analysis uses Shannon entropy. One might argue that this is not the best metric because it does not distinguish between, e.g, leaking the least significant vs. most significant bit of one's salary, while learning the latter is much more valuable to an adversary than learning the former. However,

[1] We use the term "participants" to denote parties contributing inputs to the computation. The computation itself can be performed by a different set of parties, but our result is independent of the mechanism used to realize secure function evaluation.

as we show throughout this work, information leakage for this application is always small regardless of the setup. In particular, the most favorable for the adversary setup across all distributions discloses only about 0.7 bits of entropy, i.e., the adversary cannot learn even a single bit of target's salary. Furthermore, we derive effective mechanisms for reducing information loss to a controlled low level such that the worst case scenario will not realize. We conduct similar analysis using min-entropy in the full version of the paper [8] and show that Shannon entropy trends are consistent with those for min-entropy. A primary advantage of using Shannon entropy is that we are able to go much further in our analysis and ultimately derive close-form expressions, which cannot be accomplished for other metrics.

## 2 RELATED WORK

In what follows, we review prior literature on information disclosure from function output in the context of computing on private data and related techniques that limit information disclosure.

### 2.1 Quantitative Information Flow

The field of *quantitative information flow* is closely related to our work. Denning [23] is credited as the first to quantify information flow as a measure of the interference between variables at two stages during a program's execution (typically denoted by "high-" and "low-security" variables, which equates to the target's inputs and output in our setting, respectively). Smith [52] formally established the foundations of quantifying the information leakage under the threat model that an attacker can recover a secret in one attempt (denoted by the notion of *vulnerability*). It has been shown by Massey [42] that the Shannon entropy cannot capture this information under the guessing assumption, and Smith recommends min-entropy in its place. Alvim et al. [6] generalized the min-entropy into the $g$-leakage to incorporate gain functions to model the *benefit* an adversary gains from making guesses about the secret. Subsequent works encompassed variations on the $g$-leakage [5]. Other works in differential privacy feature derivations of leakage bounds [19], leakage analysis in the case of an adaptive adversary [34], and knowledge-based approaches for measuring risk [41, 47].

The fundamental advantage of our Shannon-based approach is the ability to derive closed-form expressions for the information leakage of the average salary computation, while other metrics do not share this characteristic. For example, the chain rule of entropy (a simple, yet critical component of our analysis) is not satisfied if min-entropy is used [33, 51] in place of Shannon entropy. Our reductions would no longer hold, and we would be forced to resort to complete enumeration or approximation methods to compute the entropy. However, in the full version we provide supplementary analysis that demonstrates similarities between Shannon entropy and min-entropy based analyses. We also remain open to evaluating other metrics in the future.

An additional distinction between our work and existing literature on (quantitative) information flow is that we do not consider possible leakage from intermediate aspects of a computation's execution. Whereas other works may examine a program's loops [41], side-channel vectors [34], or inter-dependent structures [7], we strictly investigate the relationship between the output and target's input, since function itself is assumed to be evaluated using secure multi-party protocols.

### 2.2 Function Information Disclosure

Existing literature on information leakage from the output of a secure function evaluation is limited, relative to the rest of the field of secure computation. Secure multi-party protocols are designed to guarantee no information is disclosed throughout a computation, but do not ensure input protection after the output is revealed. The work of Deshpande et al. [24–26] was pioneering in that respect and designed secure multi-party protocols for business applications that ensured that the function being evaluated is *non-invertible*, i.e., no participant can infer other participants' inputs from the output. A trivially invertible example is the average salary calculation between two individuals, since either party can recover the other's input exactly. Deshpande et al. [25, 26] first addressed non-invertibility in the context of secure supply chain processes. The proposed protocols offered protection from inference of future inputs to a repeated calculation after a result is disclosed. A later work by Deshpande et al. [24] achieved non-invertibility for a framework designed for secure price masking for outsourcing manufacturing. The authors argued information leakage was minimal by analyzing mutual information between correlated normal random variables, but did not consider other distributions or entropy metrics.

Ah-Fat and Huth [1] provided the first in-depth analysis of information leakage from the outputs of secure multi-party computations. The authors formalized two metrics to measure expected information flow from the attacker's and target's perspectives, namely, the *attacker's weighted average entropy* (awae) and *target's weighted average entropy* (twae), respectively. Participants' inputs are modeled using probability distributions and were specified to be uniform, but this constraint can be relaxed. The inherent difficulty of this entropy-based approach is the requirement to enumerate every possible input combinations from all parties, which scales poorly as the input space and number of participants grow. We utilize their definitions for our analysis and demonstrate their utility to computation designers to determine potential disclosure about participants' inputs

This model was expanded in [2] to encompass the Rényi, min-, and $g$-entropy. The extension is presented in combination with a technique for distorting secure computation outputs to limit information disclosure from the output and achieve balance between accuracy and privacy. This was further developed in [3] with a fuzzing method based on randomized approximations. A closed-form expression for the min-entropy of a two- and three-party auction was derived in [4], alongside a conjecture for the case with an arbitrary number of parties.

Conceptually, the notion of *output privacy* is related to our work. The terminology was introduced in the field of data mining [15, 35, 43, 44, 56], with the goal of designing techniques to protect inputs from inference attacks on the output model. Information about the inputs that can be obtained from the output includes, but is not limited to, properties which can be uniquely attributed to a small number of input participants. Conventional approaches for minimizing disclosure involve applying transformations on the result via monotonic functions [15] or even proactive learning [56]. These techniques have little to no impact on the result of the computation. This direction differs from our work since the type of disclosure they aim to rectify is not quantified.

There is also literature that uses specific formulations to demonstrate that computation does not disclose sensitive information about participants. This includes resilience to *membership inference attacks* [50, 53] in the context of machine learning training and *differential privacy* [27, 28] for statistical databases. In particular, differential privacy ensures the output of a query is negligibly dependent on a single individual's record in the database and resilience to membership inference attacks prevents one from determining whether a specific individual's data was used for model training. These concepts have no direct relationship to our work, aside from designing mechanisms for lower information disclosure as a result of computation on private data. In this work, we do so by varying the number of participants in the computation, while other methods augment the function directly to produce a differentially private output.

## 3 PRELIMINARIES

Following [1]'s notation, let $P$ denote the set of all participants in a computation with $|P| = m$. All participants $P$ are partitioned into three groups: parties controlled by an attacker $A \subset P$, a group of parties being targeted $T \subseteq P \setminus A$, and the remaining participants called spectators $S = P \setminus (A \cup T)$. Let the random variable $X_{P_i}$ correspond to the input of a single participant $P_i$ and $x_{P_i}$ denotes a value that $X_{P_i}$ takes. In addition, the notation $\vec{X}_P = (X_{P_1}, \ldots, X_{P_m})$ denotes a multidimensional random variable and $\vec{x}_P$ is a vector of the individual values of the same size. We also let $X_P = \sum_i X_{P_i}$ define a new random variable representing the sum of the participants' random variables. The same notation applies to the sets $A$, $T$, and $S$. Our present analysis is based upon the assumption that all participants' inputs are independent and identically distributed, which we consequently relax.

For discrete distributions, we use Shannon entropy $H(X)$ to measure the information of a discrete random variable $X$ with mass function $\Pr(X = x)$ defined over domain $D_X$. Specifically,

$$H(X) = -\sum\nolimits_{x \in D_X} \Pr(X = x) \cdot \log \Pr(X = x),$$

where all logarithms are to the base 2. If we are dealing with continuous distributions, we shift to the differential entropy $h(X)$ with density function $f(x)$ over the support set $\mathcal{X}$, defined as

$$h(X) = -\int_{\mathcal{X}} f(x) \log f(x) dx.$$

We study information leakage of the computation of the average:

$$o = f(\vec{x}_A, \vec{x}_T, \vec{x}_S) = \frac{1}{m}\left(\sum\nolimits_i x_{T_i} + \sum\nolimits_j x_{A_j} + \sum\nolimits_k x_{S_k}\right),$$

where $o$ denotes the output of the function. We model the output $o$ by the random variable $O$ defined over the domain $D_O$, namely

$$O = \frac{1}{m}\left(\sum\nolimits_i X_{T_i} + \sum\nolimits_j X_{A_j} + \sum\nolimits_k X_{S_k}\right).$$

The $1/m$ factor can be ignored in the final expression since the number of participants is typically known by all parties and can trivially be removed from the output. We omit it throughout the remainder of the paper.

In this work, we consider distributions where the sum of independent individual random variables is well studied and their mass or density functions have closed-forms expressions or can be reasonably approximated. This includes the following distributions:

- *Discrete uniform* $\mathcal{U}(a, b)$, where $a$ and $b$ are integers corresponding to the minimum and maximum of the range of the support set $\{a, a+1, \ldots, b-1, b\}$.
- *Poisson* $\text{Pois}(\lambda)$, where $\lambda \in \mathbb{R}_{>0}$ is the shape parameter that indicates the expected (average) rate of an event occurring over a given interval.
- *Normal (Gaussian)* $\mathcal{N}(\mu, \sigma^2)$, where $\mu \in \mathbb{R}$ and $\sigma^2 \in \mathbb{R}_{>0}$ correspond to the mean and squared standard deviation, respectively.
- *Log-normal* $\log \mathcal{N}(\mu, \sigma^2)$ with parameters $\mu \in \mathbb{R}$ and $\sigma^2 \in \mathbb{R}_{>0}$, which correspond to the mean and squared standard deviation of the random variable's natural logarithm.

$X \sim \text{Dist}$ indicates that random variable $X$ has distribution Dist.

As stated earlier, Ah-Fat and Huth [1] provided multiple information-theoretic measures to quantify information disclosure after a function evaluation, which we use here:

Definition 1 ([1]). *The joint weighted average entropy (*jwae*) of a target $T$ attacked by parties $A$ is defined over all $\vec{x}_A \in D_A$ and $\vec{x}_T \in D_T$ as*

$$\text{jwae}(\vec{x}_A, \vec{x}_T) = \sum\nolimits_{o \in D_O} \Pr(O = o \mid \vec{X}_A = \vec{x}_A, \vec{X}_T = \vec{x}_T) \cdot H(\vec{X}_T \mid \vec{X}_A = \vec{x}_A, O = o).$$

This metric measures the information an attacker would learn (on average) about the target when the input vectors are $\vec{x}_A$ and $\vec{x}_T$. One can subsequently define the average of the jwae over all possible $\vec{x}_T$ or $\vec{x}_A$ vectors weighted by their respective prior probabilities.

Definition 2 ([1]). *The target's weighted average entropy (*twae*) of a target $T$ attacked by parties $A$ is defined for all $\vec{x}_T \in D_T$ as*

$$\text{twae}(\vec{x}_T) = \sum\nolimits_{\vec{x}_A \in D_A} \Pr(\vec{X}_A = \vec{x}_A) \cdot \text{jwae}(\vec{x}_A, \vec{x}_T).$$

The twae informs a target how much information an attacker can learn about its input when the input is $\vec{x}_A$.

Definition 3 ([1]). *The attacker's weighted average entropy (*awae*) of a target $T$ attacked by parties $A$ is defined for all $\vec{x}_A \in D_A$ as*

$$\text{awae}(\vec{x}_A) = \sum\nolimits_{\vec{x}_T \in D_T} \Pr(\vec{X}_T = \vec{x}_T) \cdot \text{jwae}(\vec{x}_A, \vec{x}_T).$$

The awae informs an attacker about how much information it can learn about the target's input when the attacker's input vector is $\vec{x}_A$. The attacker can consequently compute the awae on all values in $D_A$ to determine which input maximizes the information learned about the target's input (and thus what should be entered into the

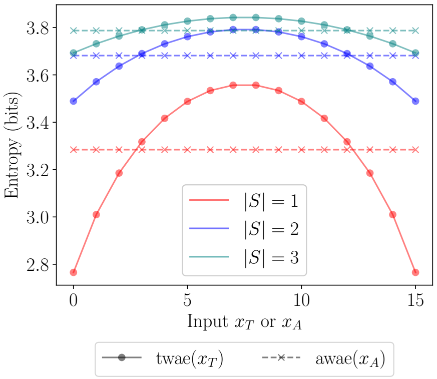


**Figure 1: The twae($\vec{x}_T$) and awae($\vec{x}_A$) using inputs over $\mathcal{U}\,(0, 15)$ with a different number of spectators $|S|$.**

computation). Using the definition of jwae, it follows that:

$$\begin{aligned}\text{awae}(\vec{x}_A) = &\sum_{\vec{x}_T \in D_T} \Pr(\vec{X}_T = \vec{x}_T) \sum_{o \in D_O} \Pr(O = o \mid \vec{X}_A = \vec{x}_A, \vec{X}_T = \vec{x}_T) \\ &\cdot H(\vec{X}_T \mid \vec{X}_A = \vec{x}_A, O = o) \\ = &\sum\nolimits_{\vec{x}_T \in D_T} \sum\nolimits_{o \in D_O} \Pr(O = o, \vec{X}_T = \vec{x}_T \mid \vec{X}_A = \vec{x}_A) \\ &\cdot H(\vec{X}_T \mid \vec{X}_A = \vec{x}_A, O = o).\end{aligned}$$

Since $\vec{X}_T$ is independent of $\vec{X}_A$, we derive that awae equals to conditional entropy:

$$\begin{aligned}\text{awae}(\vec{x}_A) &= \sum_{o \in D_O} \Pr(O = o \mid \vec{X}_A = \vec{x}_A) \cdot H(\vec{X}_T \mid \vec{X}_A = \vec{x}_A, O = o) \\ &= H(\vec{X}_T \mid \vec{X}_A = \vec{x}_A, O)\end{aligned}$$

where the last equality is due to the definition of conditional entropy.

# 4 SINGLE EXECUTION

In this section we analyze a single execution of the average function on private inputs and the associated information disclosure of the target's inputs. Recall that the computation is modeled by

$$O = f(\vec{X}_A, \vec{X}_T, \vec{X}_S) = \sum\nolimits_i X_{T_i} + \sum\nolimits_j X_{A_j} + \sum\nolimits_k X_{S_k}, \qquad (1)$$

and we let $n = |S|$ denote the number of spectators.

As a first step, we plot the values of awae and twae for our function of interest. Figure 1 illustrates these values with a single adversarial participant, a single target and a varying number of spectators (1–3). All inputs follow the uniform distribution $\mathcal{U}\,(0, 15)$. Calculating the twae and awae values using Definitions 2 and 3 requires enumerating all input and output combinations. This quickly becomes computationally inefficient as the input space grows.

Each participant, acting as a target, can utilize the twae prior to the computation to determine how much information an attacker can learn (on average) from the output for a specific input that the participant enters into the computation. As the figure illustrates, the target's remaining average entropy is maximized when the input is in the middle of the range, indicating that those values have better protection than inputs near the extrema. As the number of spectators increases, the curves shift upwards, i.e., the uncertainty about the target's input increases and the gap in the uncertainty between different input values reduces.

The awae, on the other hand, gives an adversary the ability to determine which input to enter into the computation that leads to maximum information disclosure about a target's input (without knowing what input the target used). As displayed in the figure, the adversarial knowledge does not change by varying its inputs into the computation. This is consistent with our intuition that, given the output, the adversary can remove their contribution to the computation and possess information about the sum of the inputs of the remaining parties. We formalize this as the following result:

Claim 1. awae($\vec{x}_A$) *is independent of attacker's input vector* $\vec{x}_A$.

Proof. According to the chain rule of entropy which states that $H(X, Y) = H(X \mid Y) + H(Y)$ [22, Chapter 2.5], we have that:

$$\begin{aligned}H(\vec{X}_T \mid \vec{X}_A{=}\vec{x}_A, O) &= H(\vec{X}_T, O \mid \vec{X}_A = \vec{x}_A) - H(O \mid \vec{X}_A = \vec{x}_A) \\ &= H(\vec{X}_T \mid \vec{X}_A = \vec{x}_A) + H(O \mid \vec{X}_T, \vec{X}_A = \vec{x}_A) \\ &\quad - H(O \mid \vec{X}_A = \vec{x}_A) \\ &= H(\vec{X}_T) + H\left(\sum\nolimits_i X_{S_i}\right) - H\left(\sum\nolimits_i X_{T_i} + \sum\nolimits_j X_{S_j}\right),\end{aligned}$$

which is independent of $\vec{x}_A$. □

Using our notation from Section 3, the above expression for awae($\vec{x}_A$) simplifies to

$$H(\vec{X}_T) + H\left(X_S\right) - H\left(X_T + X_S\right) = H(\vec{X}_T \mid X_T + X_S).$$

The next step is to determine which measure (awae or twae) we should use in our analysis of the average salary computation. Ah-Fat and Huth [1] argued that the awae served as a more precise metric for measuring information leakage of a secure function evaluation than twae for their choice of function and used awae in their subsequent work [2]. Our perspective also aligns with that conclusion. In particular, while the twae informs the target of the amount of information leakage for the input they possess, the target may not be technically savvy enough to be able to apply the metric and make an informed decision regarding computation participation (plus, the choice to participate or not participate can leak information about their input). Perhaps more importantly, a function needs to be analyzed by the computation designers in advance and without access to the inputs of future computation participants to determine a safe setup for the participants. Thus, the available mechanism for this purpose is the attacker's perspective or awae, and we focus on this metric in the rest of this work.

Based on the above, in what follows we use $H(\vec{X}_T \mid X_T + X_S)$ to measure the leakage, and the simplified function is

$$f(\vec{X}_T, \vec{X}_S) = \sum\nolimits_i X_{T_i} + \sum\nolimits_j X_{S_j} = X_T + X_S.$$

This refines the parameters we can vary in our analysis to (1) the number of participants in the target and spectators groups and (2) the types of distributions and statistical parameters of the inputs. Furthermore, the computational difficulty associated with directly computing the awae is absent when using $H(\vec{X}_T \mid X_T{+}X_S)$. Instead, the computation simplifies to calculating the entropy of sums of random variables. We examine the behavior of the conditional entropy for several characteristic probability distributions next.

## 4.1 Discrete Distributions

We start with discrete input modeled using the uniform and Poisson distributions. The sum of $n$ identical independent Poisson random variables $X_i \sim \text{Pois}(\lambda)$ is equivalent to a single Poisson random variables $X = \sum_i X_i \sim \text{Pois}(n\lambda)$ with the mass function $\Pr(X = x) = (n\lambda)^k \, e^{-n\lambda}/(k!)$ . Note that the Poisson distribution is defined over all non-negative integers, hence the distribution has infinite support. We choose to halt the calculation of $H(X)$ when $\Pr(X = x) < 10^{-7}$ as the contribution of events beyond this point to the entropy is infinitesimal.

Conversely, the sum of $n$ identical independent uniform random variables $X_i \sim \mathcal{U}\,(0, N-1)$ is not immediately obvious. Caiado and Rathie [16] derived several equivalent expressions for the mass function of the sum of $n$ uniform random variables, one of which we use in our analysis and is defined as:

$$\Pr(X = x) = \frac{n}{N^n}\left(\sum\nolimits_{p=0}^{\lfloor x/N \rfloor} \frac{\Gamma\,(n + x + pN)\,(-1)^p}{\Gamma(p+1)\Gamma(n-p+1)\Gamma(x-pN+1)}\right),$$

where $\Gamma(n) = (n-1)!$ is the Gamma function. The domain of $X$ is $\{0, \ldots, n(N-1)\}$.

Our analysis of awae for these two distribution is given in Figures 2 and 3, respectively. We compute and display

- the original entropy of target's inputs prior to the computation $H(\vec{X}_T)$ (subfigure a)
- the awae or target's remaining entropy after the computation $H(\vec{X}_T \mid X_T + X_S)$ (subfigure a)
- their difference of the two that represents the absolute entropy loss $H(\vec{X}_T) - H(\vec{X}_T \mid X_T + X_S)$ (subfigure b) and
- the entropy loss relative to the original entropy prior to the execution $(H(\vec{X}_T) - H(\vec{X}_T \mid X_T + X_S))/H(\vec{X}_T)$ (subfigure c)

with a single target ($|T| = 1$), a varying number of spectators, and varying distribution parameters. Relative entropy loss is included to demonstrate to potential input contributors, who are likely non-experts, that information disclosure is small. That is, disclosure of, e.g., 5% of input's information is easier to explain to non-experts than 0.1 bits of entropy. The absolute loss is equivalent to the mutual information between the target input and the output: $I(\vec{X}_T; O) = H(\vec{X}_T) - H(\vec{X}_T \mid X_T + X_S)$.

Figure 2 presents this information for the Poisson distribution with $\lambda \in \{4, 8, \ldots, 128\}$. In Figure 2a, entropy after the execution converges toward the corresponding entropy prior to the execution for all values of $\lambda$ as the number of spectators increases. Increasing $\lambda$ by a factor of two repeatedly yields an upward shift of these two curves by a constant amount while preserving their respective shapes. The increase is expected as a result of the inputs having more entropy as $\lambda$ increases, but the shape of the remaining entropy is notable, as $\lambda$ does not appear to impact the entropy loss. This is further confirmed when displaying the absolute entropy loss in Figure 2b: The resultant curves overlap each other, regardless of $\lambda$.

The relative entropy loss in Figure 2c, calculated as a percentage of the target's initial entropy, demonstrates how many spectators the computation needs to include to lower the entropy loss to the desired level. The larger the original entropy is (larger $\lambda$), the fewer spectators will be needed to stay within the desired percentage. For example, 5 spectators are needed with $\lambda = 4$ to limit relative loss to 5% (marked by ■) and 24 spectators are needed to cap the loss at 1% (marked by ×). When $\lambda = 128$, the number of spectators reduces to 3 and 13 to maintain loss tolerances of 5% and 1%, respectively.

The same trends hold for the uniform distribution in Figure 3, where we use $N \in \{8, 16, \ldots, 256\}$, but the values themselves slightly differ. For example, the absolute entropy loss in Figure 3b is slightly larger than the loss in Figure 2b when the number of spectators is small. When $N = 8$ with 3 bits of original entropy, 5 and 24 spectators are needed to achieve at most 5% and 1% relative loss, respectively. This is the same as what was observed for Poisson distribution with 3-bit inputs ($\lambda = 4$).

## 4.2 Continuous Distributions

For continuous distributions, we shift to differential entropy and analyze normal and log-normal distributions, the latter of which is typically used to model salaries. While there is no direct relationship between differential and Shannon entropy (see [22, Chapter 8.3]), we demonstrate that they exhibit very similar behavior for the average computation.

The differential entropy of a normal random variable $X_i \sim \mathcal{N}(\mu, \sigma^2)$ is $h(X_i) = \frac{1}{2}\log\left(2\pi e\sigma^2\right)$ [22, Chapter 8.1]. The sum of $n$ identical normal random variables is also normal, namely $X \sim \mathcal{N}(n\mu, n\sigma^2)$. This enables us to directly apply the differential entropy definition to the sum.

The log-normal distribution is a well-established means of modeling salary data for 99% of the population [20], with the top 1% modeled by the Pareto distribution [54]. The differential entropy of a log-normal random variable $X_i \sim \log\mathcal{N}(\mu, \sigma^2)$ is $h(X_i) = \log\left(e^{\mu+\frac{1}{2}}\sqrt{2\pi\sigma^2}\right)$. However, the sum of $n$ log-normal random variables has no closed form and is an active area of research [9–11, 21, 30, 48, 49, 57]. We adopt the Fenton-Wilkinson (FW) approximation [2] [21, 30] that specifies a sum of $n$ identical independent log-normal random variables $X_i \sim \log\mathcal{N}(\mu, \sigma^2)$ as another log-normal random variable $X \sim \log\mathcal{N}(\hat{\mu}, \hat{\sigma}^2)$ with parameters

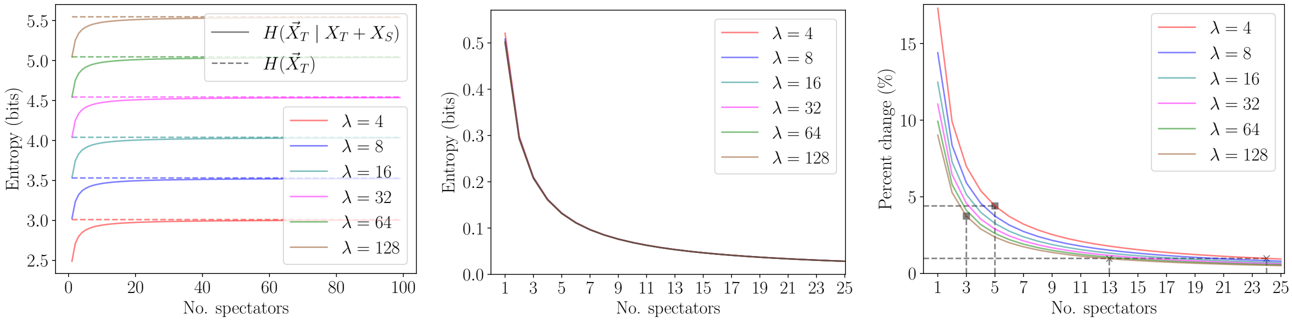
$$\hat{\sigma}^2 = \ln\left(\frac{\exp(\sigma^2) - 1}{n} + 1\right),\ \hat{\mu} = \ln(n \cdot \exp(\mu)) + \frac{1}{2}\left(\sigma^2 - \hat{\sigma}^2\right).$$

This enables us to compute differential entropy using a closed-form expression. Unlike prior distributions, we use a single set of $\mu$ and $\sigma^2$ parameters calculated from real salary data in [17]; namely, $\mu = 1.6702$ and $\sigma^2 = 0.145542$.

Figures 4 and 5 present experimental evaluation of entropy loss with a single target and a varying number of spectators for normal and log-normal distributions, respectively. As before, we report target's entropy before and after the execution, the difference of the two as the absolute entropy loss, and the entropy loss relative to the entropy before the execution.

In Figure 4 (normal), we set the mean $\mu = 0$ for all experiments (since differential entropy does not depend on $\mu$) and vary $\sigma^2$ from 4 to 128. The results are consistent with the discrete counterparts in terms of the trends, curve shapes, and specific values. The absolute loss in Figure 4b is once again constant for any $\sigma^2$ and the relative loss is dictated by the amount of input's entropy in Figure 4c. When

[2] Other approximations for the sum of log-normal random variables are difficult to translate into an expression for the differential entropy and hence we choose the FW approximation. Its disadvantage is that that the FW approximation deteriorates for $\sigma^2 > 4$ and small values of $x$ in the density function [10, 57]. Fortunately, our $\sigma^2$ is sufficiently small, allowing us to use the FW approximation free of consequence.

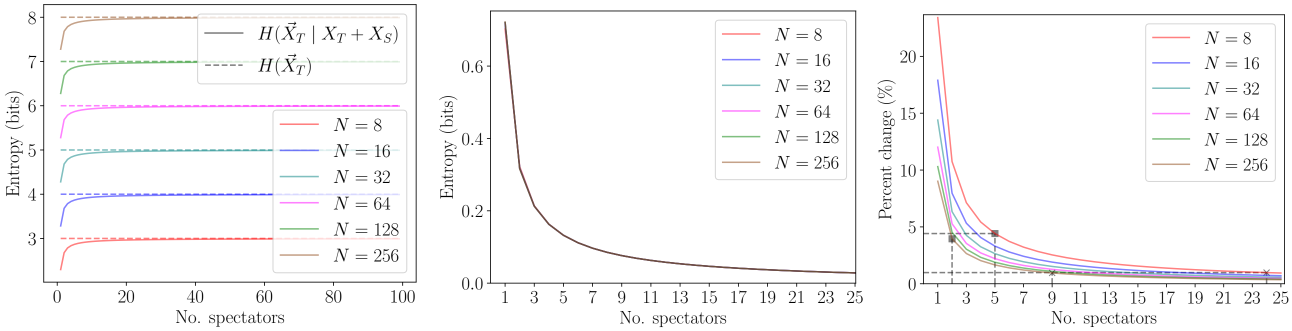


**(a) Target's entropy before $H(\vec{X}_T)$ and after $H(\vec{X}_T \mid X_T + X_S)$ the execution.**

**(b) Target's absolute entropy loss $H(\vec{X}_T) - H(\vec{X}_T \mid X_T + X_S)$.**

**(c) Target's relative entropy loss $\frac{H(\vec{X}_T) - H(\vec{X}_T \mid X_T + X_S)}{H(\vec{X}_T)}$.**

**Figure 2: Analysis of target's entropy loss using the Poisson distribution with Pois($\lambda$), and varying $\lambda$ with $|T| = 1$.**



**(a) Target's entropy before $H(\vec{X}_T)$ and after $H(\vec{X}_T \mid X_T + X_S)$ the execution.**

**(b) Target's absolute entropy loss $H(\vec{X}_T) - H(\vec{X}_T \mid X_T + X_S)$.**

**(c) Target's relative entropy loss $\frac{H(\vec{X}_T) - H(\vec{X}_T \mid X_T + X_S)}{H(\vec{X}_T)}$.**

**Figure 3: Analysis of target's entropy loss using the uniform distribution with $\mathcal{U}(0, N-1)$, and varying $N$ with $|T| = 1$.**

$\sigma^2 = 4$ and inputs have 3 bits of entropy, the number of spectators required to maintain at most 5% and 1% entropy loss (5 and 24 spectators, respectively) is the same as for Poisson and uniform distributions with 3-bit inputs ($\lambda = 4$ and $N = 8$, respectively). With 5.5-bit inputs ($\sigma^2 = 128$), 3 and 13 spectators are needed to achieve at most 5% and 1% loss, respectively, which the same as for Poisson distribution with 5.5-bit inputs ($\lambda = 128$).

The results in Figure 5 (log-normal with real salary parameters) are consistent with both the discrete and continuous distributions. Surprisingly, we observe the same 5 and 24 spectators achieve at most 5% and 1% relative loss, as observed with all other distributions (with input original entropy being slightly over 3 bits).

Before concluding our discussion of continuous distributions, we are able to show one more result. We experimentally demonstrated that the amount of absolute entropy loss is parameter-independent for several distributions, but we can formally prove this for normally distributed inputs:

Claim 2. *If the inputs are modeled by independent identically distributed normal random variables, the absolute entropy loss $h(\vec{X}_T) - h(\vec{X}_T \mid X_T + X_S)$ depends only on the number of target $|T| = t$ and spectator $|S| = n$ inputs and is $\frac{1}{2}\log\left(\frac{t}{n} + 1\right)$.*

Proof. Let $|T| = t$ and $|S| = n$, such that $X_T \sim \mathcal{N}(0, t\sigma^2)$ and $X_S \sim \mathcal{N}(0, n\sigma^2)$. The absolute entropy loss is therefore

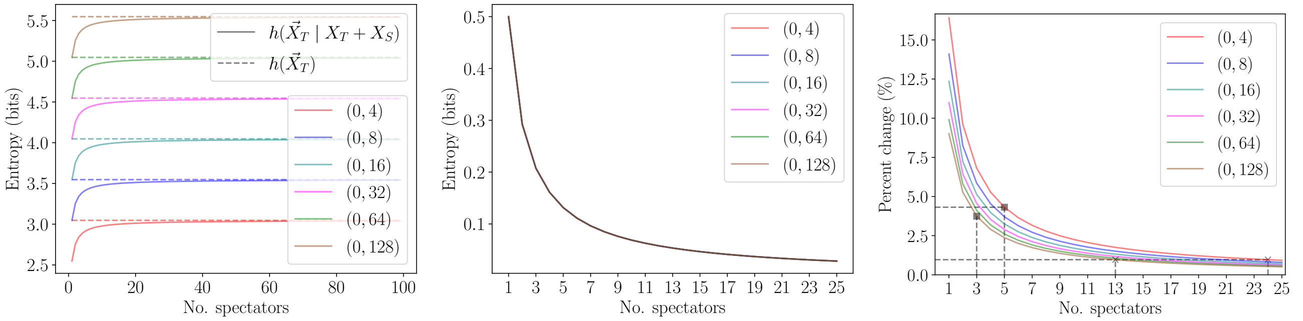
$$\begin{aligned} h(\vec{X}_T) - h(\vec{X}_T \mid X_T + X_S) &= h(\vec{X}_T) - \Big(h(\vec{X}_T) + h\left(X_S\right) - h\left(X_T + X_S\right)\Big) \\ &= h\left(X_T + X_S\right) - h\left(X_S\right) \\ &= \frac{1}{2}\log 2\pi e(t+n)\sigma^2 - \frac{1}{2}\log 2\pi e n\sigma^2 \\ &= \frac{1}{2}\log\left(\frac{t}{n} + 1\right) = \Theta\left(\log\left(\frac{t}{n} + 1\right)\right), \end{aligned}$$

which depends only on $n$ and $t$. □

## 4.3 Discrete vs. Continuous Distributions

We next compare the information loss across all four (discrete and continuous) distributions. We choose parameters to maintain the initial entropy of an input, $H(X_i)$ or $h(X_i)$, to be approximately 3 bits, as to reasonably correspond to the log-normal distribution. This leads to Pois(4), $\mathcal{U}(0, 7)$, and $\mathcal{N}(0, 4)$. We plot this information for a single target and a varying number of spectators in Figure 6.

In the figure, all distributions converge with ≥ 4 spectators and are very close even with 3 spectators. This convergence on large values is expected as a consequence of the central limit theorem. From

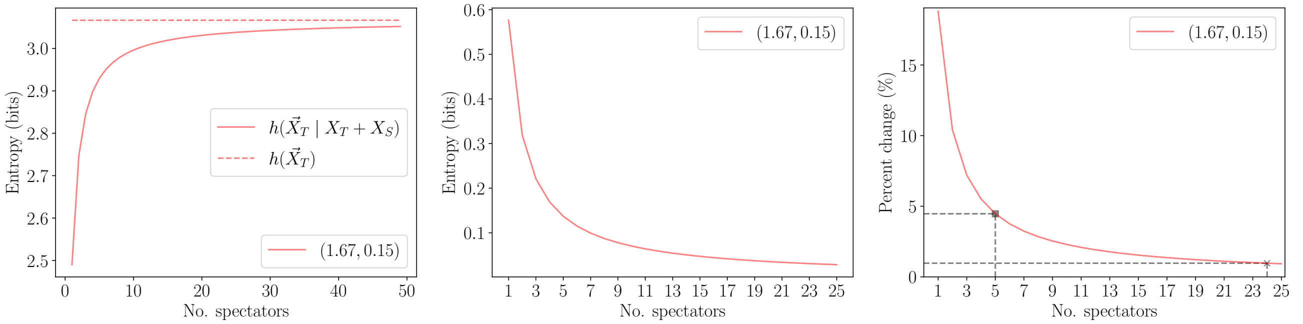


**(a) Target's entropy before $h(\vec{X}_T)$ and after $h(\vec{X}_T \mid X_T + X_S)$ the execution.**

**(b) Target's absolute entropy loss $h(\vec{X}_T) - h(\vec{X}_T \mid X_T + X_S)$.**

**(c) Target's relative entropy loss $\frac{h(\vec{X}_T) - h(\vec{X}_T \mid X_T + X_S)}{h(\vec{X}_T)}$.**

**Figure 4: Analysis of target's entropy loss using the normal distribution with $\mathcal{N}(0, \sigma^2)$, and varying $\sigma^2$ with $|T| = 1$.**



**(a) Target's entropy before $h(\vec{X}_T)$ and after $h(\vec{X}_T \mid X_T + X_S)$ the execution.**

**(b) Target's absolute entropy loss $h(\vec{X}_T) - h(\vec{X}_T \mid X_T + X_S)$.**

**(c) Target's relative entropy loss $\frac{h(\vec{X}_T) - h(\vec{X}_T \mid X_T + X_S)}{h(\vec{X}_T)}$.**

**Figure 5: Analysis of target's entropy loss using the log-normal distribution with $\log \mathcal{N}(1.6702, 0.145542)$ and $|T| = 1$.**

the four distributions, the closest are the Poisson results with $\lambda = 4$ (discrete) and the normal distribution $\mathcal{N}(0, 4)$ (continuous). Unlike normal, log-normal, and the single-variate uniform distributions, an exact expression of the entropy of a Poisson distribution has not been derived. Instead, when computing the necessary values in Section 4.1, we directly applied the definition of Shannon entropy. To draw a parallel between discrete and continuous distributions, and specifically show a similarity between Poisson and normal distributions, we turn to an approximation of Poisson distribution's entropy computation.

It was conjectured that for sufficiently large $\lambda$ (e.g., $\lambda > 10$), the Poisson distribution's Shannon entropy can be approximated by $H(X_i) = \frac{1}{2}\log(2\pi e\lambda)$, which resembles $h(X_i) = \frac{1}{2}\log(2\pi e\sigma^2)$ used for normal distributions. Evans and Boersma [29] proposed a tighter bound (further formalized by Cheraghchi in [18]), to be

$$H(X_i) = \frac{1}{2}\log(2\pi e\lambda) - \frac{1}{12\lambda} - \frac{1}{24\lambda^2} - \frac{19}{360\lambda^3} + O(\lambda^4)$$

and remains close to that of normal distribution with $\sigma^2 = \lambda$.

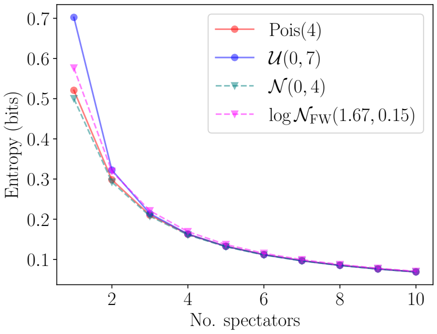


**Figure 6: Comparing target's absolute entropy loss for discrete $H(\vec{X}_T) - H(\vec{X}_T \mid X_T + X_S)$ and continuous $h(\vec{X}_T) - h(\vec{X}_T \mid X_T + X_S)$ distributions.**

One implication of this result for us is that Claim 2, which we demonstrated for normal distributions, would apply to the approximation of Poisson distributions as well. As a result, we obtain independence of the (absolute) entropy loss of distribution parameters for both discrete and continuous distributions and almost identical behavior across the distributions as a function of the number of spectators.

Furthermore, our analysis using Shannon/differential entropy is partially echoed for the min-entropy (demonstrated in the full version of the text [8]). In particular, we conjecture independence of the attacker's input in the min-entropy-based awae and show that plots for the absolute min-entropy loss closely resemble those for Shannon entropy for the Poisson distribution.

Up to this point, we have assumed that all participants' inputs are sampled from identically distributed random variables. However, we can relax this assumption and investigate if/how the information disclosure changes if parties' inputs are non-identically distributed. For example, employee salaries may differ slightly from company to company, while still following the same distribution. We can model this by adjusting the statistical parameters of individual participants. This poses an interesting problem where there are multiple groups with different distribution parameters. As such, we are interested in determining which prior claims are still valid or need to be modified. Claim 1 (attacker input independence) will hold regardless of how participants' inputs are distributed. Conversely, Claim 2 (dependence on the number of targets and participants in the absolute entropy loss) must be reworked since the claim is formulated under the assumption that inputs are identically distributed. We investigate this relaxation, as well as conduct additional experiments, in the full version of the paper [8].

## 5 TWO EXECUTIONS

A natural generalization of the results of the prior section is to consider executing the average salary computation more than once. For example, after running the Boston gender pay gap study once, the same computation was executed the following year with an extended set of participants. In this case, if the time interval between the executions is small enough such that the inputs do not change between the executions or change minimally, one would expect that repeated participations would lead to additional information disclosure compared to a single execution. Thus, in this section we analyze the case of two executions and demonstrate their impact on the participants. We consider both the cases when a target contributes its input to both executions and when the target participates only in one of the executions and other takes place without the target, but on related inputs. Both cases result in additional information disclosure compared to a single execution, which we quantify in this section.

We partition the set of spectators $S$ into the following subsets:

- spectators present only in the first execution $S_1 \subset S$,
- spectators present only in the second execution $S_2 \subseteq S \setminus S_1$,
- and spectators present in both executions $S_{12} = S \setminus (S_1 \cup S_2)$.

A person participating more than once (target or spectator) enters the same input into both execution.

When the target participates in both executions, we have:

$$O_1 = \sum_i X_{T_i} + \sum_{i \in S_{12}} X_i + \sum_{i \in S_1} X_i = X_T + X_{S_{12}} + X_{S_1}$$
$$O_2 = \sum_i X_{T_i} + \sum_{i \in S_{12}} X_i + \sum_{i \in S_2} X_i = X_T + X_{S_{12}} + X_{S_2}.$$

The random variables $O_1$ and $O_2$ are *not* independent, as they both are comprised of $X_T$ and $X_{S_{12}}$. We therefore want to compute the conditional entropy (using differential entropy notation):

$$h(\vec{X}_T \mid O_1, O_2) = h(\vec{X}_T, O_1, O_2) - h(O_1, O_2). \tag{2}$$

Claim 3. *The above conditional entropy can be expressed as*

$$h(\vec{X}_T | O_1, O_2) = h(\vec{X}_T) + h(X_{S_{12}}{+}X_{S_1}, X_{S_{12}}{+}X_{S_2}) - h(O_1, O_2). \tag{3}$$

The derivation can be found in the full version of the text.

In the special case when no spectators participate in both executions (i.e., $S_{12} = \emptyset$), the middle term simplifies to $h(X_{S_1}) + h(X_{S_2})$.

When the target participates only in one of the experiments, we define executions $O_1'$ and $O_2'$, which are the same as $O_1$ and $O_2$, respectively, except that the target's inputs are not included. For instance, $O_1' = X_{S_{12}} + X_{S_1}$. The relevant entropies in that case are $h(\vec{X}_T | O_1', O_2)$ and $h(\vec{X}_T | O_1, O_2')$.

The above requires us to introduce the definition of joint entropy of correlated random variables. Now, the normal distribution stands out among those considered in Section 4 as a suitable candidate for our analysis. The generalized multivariate normal distribution is well-studied and has a closed-form differential entropy, which we discuss next.

### 5.1 Bivariate Normal Distributions

Evaluating Equation 3 requires defining the differential entropy of a multivariate normal random variable. We then derive the necessary core parameters for our distributions and use them to compute the conditional entropy.

Let $X_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ be a single normal random variable as defined in Section 3. We define $\vec{X} = (X_1, \ldots, X_k)^{\mathrm{T}}$ to be a general multivariate normal distribution of a $k$-dimensional random vector, with $\vec{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$. Here, $\boldsymbol{\mu} \in \mathbb{R}^k$ is the mean vector specified as $\boldsymbol{\mu} = \mathrm{E}[\vec{X}] = (\mathrm{E}[X_1], \mathrm{E}[X_2], \ldots, \mathrm{E}[X_k])^{\mathrm{T}} = (\mu_1, \mu_2, \ldots, \mu_k)^{\mathrm{T}}$, and $\Sigma \in \mathbb{R}^{k \times k}$ is the $k \times k$ covariance matrix with each element defined as $\Sigma_{i,j} = \mathrm{E}\left[(X_i - \mu_i)(X_j - \mu_j)\right] = \mathrm{Cov}\left[X_i, X_j\right]$. The differential entropy of the multivariate normal distribution $\vec{X}$ is given by $h(\vec{X}) = \frac{1}{2} \log\left((2\pi e)^k \det \Sigma\right)$, [22, Chapter 8.4] where $\det \Sigma$ is the determinant of the covariance matrix. The next step is to characterize our multivariate distributions and determine their covariance matrices. We also derive their mean vectors which are used for intermediate results.

To compute the second and third terms of Equation 3, we formalize the bivariate distributions $\vec{S} = (X_{S_{12}} + X_{S_1}, X_{S_{12}} + X_{S_2})^{\mathrm{T}}$ and $\vec{O} = (O_1, O_2)^{\mathrm{T}}$. We denote $\mu_P = \sum_i \mu_{P_i}$ and $\sigma_P^2 = \sum_i \sigma_{P_i}^2$ as the sum of the means and standard deviations, respectively, of all participants within a group $P$. Note that the mean is absent from the formula for the differential entropy, and therefore we can safely assume all $\mu_i = 0$. Starting with $\vec{O}$, we invoke the linearity of the expectation for the mean vector:

$$\boldsymbol{\mu}_{\vec{O}} = \begin{pmatrix} \mathrm{E}[O_1] \\ \mathrm{E}[O_2] \end{pmatrix} = \begin{pmatrix} \mathrm{E}\left[X_T{+}X_{S_{12}}{+}X_{S_1}\right] \\ \mathrm{E}\left[X_T{+}X_{S_{12}}{+}X_{S_2}\right] \end{pmatrix} = \begin{pmatrix} \mu_T{+}\mu_{S_{12}}{+}\mu_{S_1} \\ \mu_T{+}\mu_{S_{12}}{+}\mu_{S_2} \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}.$$

For the covariance matrix, using the properties Cov $[X, X]$ = Var $[X]$ = $\sigma_X^2$ and Cov $[X, Y]$ = Cov $[Y, X]$ yields

$$\Sigma_{\vec{O}} = \begin{pmatrix} \mathrm{Cov}\,[O_1, O_1] & \mathrm{Cov}\,[O_1, O_2] \\ \mathrm{Cov}\,[O_2, O_1] & \mathrm{Cov}\,[O_2, O_2] \end{pmatrix} = \begin{pmatrix} \mathrm{Var}\,[O_1] & \mathrm{Cov}\,[O_1, O_2] \\ \mathrm{Cov}\,[O_1, O_2] & \mathrm{Var}\,[O_2] \end{pmatrix}$$
$$= \begin{pmatrix} \sigma_T^2 + \sigma_{S_{12}}^2 + \sigma_{S_1}^2 & \mathrm{Cov}\,[O_1, O_2] \\ \mathrm{Cov}\,[O_1, O_2] & \sigma_T^2 + \sigma_{S_{12}}^2 + \sigma_{S_2}^2 \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & \mathrm{Cov}[O_1, O_2] \\ \mathrm{Cov}\,[O_1, O_2] & \sigma_2^2 \end{pmatrix}.$$

The expression for Cov $[O_1, O_2]$ can be stated as follows:

Claim 4. Cov $[O_1, O_2] = \sigma_T^2 + \sigma_{S_{12}}^2$ if $S_{12}$ is non-empty, and Cov $[O_1, O_2] = \sigma_T^2$ otherwise.

The proof is available in the full version of the text [8]. The final parameters of the bivariate distribution $\vec{O}$ are

$$\boldsymbol{\mu}_{\vec{O}} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \Sigma_{\vec{O}} = \begin{pmatrix} \sigma_1^2 & \sigma_T^2 + \sigma_{S_{12}}^2 \\ \sigma_T^2 + \sigma_{S_{12}}^2 & \sigma_2^2 \end{pmatrix}.$$

Repeating this procedure for the spectator joint distribution $\vec{S}$ yields a similar set of parameters:

$$\boldsymbol{\mu}_{\vec{S}} = \begin{pmatrix} \mu_{S_{12}} + \mu_{S_1} \\ \mu_{S_{12}} + \mu_{S_2} \end{pmatrix}, \Sigma_{\vec{S}} = \begin{pmatrix} \sigma_{S_{12}}^2 + \sigma_{S_1}^2 & \sigma_{S_{12}}^2 \\ \sigma_{S_{12}}^2 & \sigma_{S_{12}}^2 + \sigma_{S_2}^2 \end{pmatrix}.$$

Equipped with expressions for $\Sigma_{\vec{O}}$ and $\Sigma_{\vec{S}}$, we are prepared to begin our experimental analysis of $h(\vec{X}_T \mid O_1, O_2)$.

## 5.2 Experimental Evaluation

The above allows us to experimentally evaluate the target's entropy loss for when inputs are normally distributed. We use normal distribution $\mathcal{N}(0, 4)$ to reasonably approximate the log-normal distribution with real data. Once again, $|T| = 1$ for concreteness and we let $|S_1| = |S_2|$ in all experiments, i.e., the number of spectators is the same in both executions.

It is informative to analyze information loss as the fraction of shared spectators changes and we do so for three different computation sizes. To be as close to the setup that guarantee 1%–5% entropy loss for the log-normal distribution (5–24 spectators), we choose to execute our experiments with 6, 10, and 24 spectators (where having an even number is beneficial for illustration purposes). This corresponds to the number of non-adversarial participants when the target is absent and the number of non-adversarial participants is one higher when the target is participating.

We display the following information in Figure 7:

- the target's initial entropy $h(\vec{X}_T)$,
- the target's entropy after a single execution $h(\vec{X}_T \mid O_1)$,
- the target's entropy after participating twice $h(\vec{X}_T|O_1, O_2)$,
- the target's entropy after participating in one of the two executions, i.e., $h(\vec{X}_T \mid O_1, O_2')$ and $h(\vec{X}_T \mid O_1', O_2)$

and plot the values as a function of the fractional overlap between two executions for a given number of spectators.

Naturally, the value of $h(\vec{X}_T \mid O_1)$ remains constant when the number of participants is fixed. We observe that when participating twice, $h(\vec{X}_T \mid O_1, O_2)$ converges to $h(\vec{X}_T \mid O_1)$ as the fraction of shared spectators increases. This is expected because at 100% overlap, we are functionally calculating $h(\vec{X}_T \mid O_1, O_1) = h(\vec{X}_T \mid O_1)$. Conversely, increasing the fraction of the overlap has the inverse effect for $h(\vec{X}_T \mid O_1, O_2')$, causing it to trend downward. At 100% overlap, $h(\vec{X}_T \mid O_1, O_2') = 0$ (point omitted from the plots). This is a consequence of effectively computing $h(\vec{X}_T \mid O_1, X_{S_{12}})$:

$$\begin{aligned} h(\vec{X}_T|O_1, X_{S_{12}}) &= h(\vec{X}_T, O_1, X_{S_{12}}) - h(O_1, X_{S_{12}}) \\ &= h(\vec{X}_T) + h(X_{S_{12}}) - \big(h(X_T + X_{S_{12}} \mid X_{S_{12}}) + h(X_{S_{12}})\big) \\ &= h(\vec{X}_T) + h(X_{S_{12}}) - \big(h(X_T) + h(X_{S_{12}})\big) = h(\vec{X}_T) - h(X_T). \end{aligned}$$

When $|T| = 1$, then $h(\vec{X}_T) = h(X_T)$, thus reducing the above equation to zero. This informs us that the output of the second computation $O_2'$ without any unique spectators reveals the target's information entirely. We analytically prove these observations in the full version of the text by deriving exact expressions for the absolute entropy loss.

Our next observation pertains to the point of intersection where $h(\vec{X}_T \mid O_1, O_2) = h(\vec{X}_T \mid O_1, O_2')$, which occurs when 50% of the spectators are shared across the computation. This appears for the special case when the total number of spectators in a single evaluation is even. Concretely, we compare

$$\begin{aligned} h(\vec{X}_T|O_1, O_2) &= h(\vec{X}_T, O_1, O_2) - h(O_1, O_2), \\ h(\vec{X}_T|O_1, O_2') &= h(\vec{X}_T, O_1, O_2') - h(O_1, O_2'). \end{aligned} \tag{4}$$

It can be shown using the procedure outlined in Section 5 that $h(\vec{X}_T, O_1, O_2) = h(\vec{X}_T, O_1, O_2')$. Therefore, we prove the following:

Claim 5. *With normally distributed inputs, the terms $h(O_1, O_2)$ and $h(O_1, O_2')$ are equal when $|S_{12}| = |S_1|$.*

Proof. Following the steps used to derive the covariance matrix of $\vec{O} = (O_1, O_2)$, the covariance matrix of $\vec{O'} = (O_1, O_2')$ is

$$\Sigma_{\vec{O'}} = \begin{pmatrix} \sigma_T^2 + \sigma_{S_{12}}^2 + \sigma_{S_1}^2 & \sigma_{S_{12}}^2 \\ \sigma_{S_{12}}^2 & \sigma_{S_{12}}^2 + \sigma_{S_2}^2 \end{pmatrix}.$$

Recall that the differential entropy of the multivariate normal is $h(\vec{X}) = \frac{1}{2} \log \left((2\pi e)^k \det \Sigma\right)$. The sole object of interest is the $\det \Sigma$ term, as the remainder contribute a constant factor. We calculate

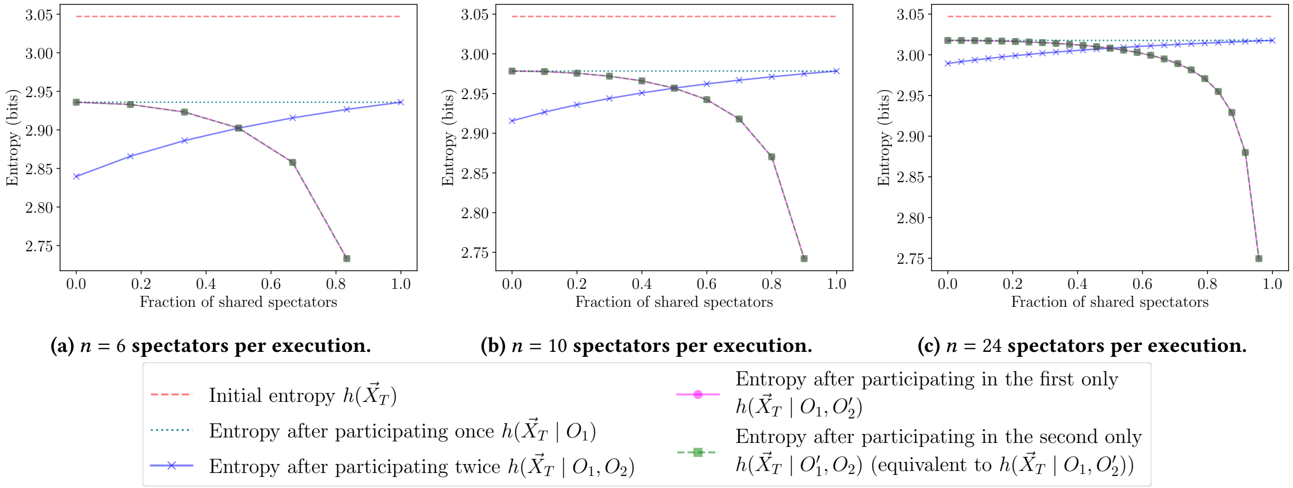
$$\begin{aligned} \det \Sigma_{\vec{O}} &= (\sigma_T^2 + \sigma_{S_{12}}^2 + \sigma_{S_1}^2)(\sigma_T^2 + \sigma_{S_{12}}^2 + \sigma_{S_2}^2) - (\sigma_T^2 + \sigma_{S_{12}}^2)^2 \\ &= \sigma_T^2(\sigma_{S_1}^2 + \sigma_{S_2}^2) + \sigma_{S_{12}}^2(\sigma_{S_1}^2 + \sigma_{S_2}^2) + \sigma_{S_1}^2\sigma_{S_2}^2. \end{aligned}$$

Similarly,

$$\begin{aligned} \det \Sigma_{\vec{O'}} &= (\sigma_T^2 + \sigma_{S_{12}}^2 + \sigma_{S_1}^2)(\sigma_{S_{12}}^2 + \sigma_{S_2}^2) - \sigma_{S_{12}}^4 \\ &= \sigma_T^2(\sigma_{S_{12}}^2 + \sigma_{S_2}^2) + \sigma_{S_{12}}^2(\sigma_{S_1}^2 + \sigma_{S_2}^2) + \sigma_{S_1}^2\sigma_{S_2}^2. \end{aligned}$$

Therefore, the equality $h(\vec{X}_T \mid O_1, O_2) = h(\vec{X}_T \mid O_1, O_2')$ is satisfiable if and only if $\sigma_{S_{12}}^2 = \sigma_{S_1}^2$, which occurs when $|S_{12}| = |S_1|$. □

As computation designers, we can minimize information disclosure for all participants by targeting 50% participants' overlap between the first and second executions. For the configurations in Figure 7, at 50% overlap, the percentages of information loss from the second evaluation relative to the first evaluation are comparable for the selected number of spectators $n$ (30.18% for $n = 6$, 31.3% for $n = 10$, and 32.45% for $n = 24$). This corresponds to the intersection points in Figure 7.



**(a)** $n = 6$ **spectators per execution.** **(b)** $n = 10$ **spectators per execution.** **(c)** $n = 24$ **spectators per execution.**

**Figure 7: Target information loss after participating in one or two computations. Omitted: if the target participates in one experiment and all the shared spectators are reused, then** $h(X_T \mid O_1, O_2') = 0$.

| Number of evaluations the target participates in | Spectator overlap | | |
|---|---|---|---|
| | 40% | 50% | 60% |
| One | 18.0% | 31.3% | 52.3% |
| Two | 40.1% | 31.3% | 23.5% |

**Table 1: Percentage of information loss after two executions relative to a single execution for** $n = 10$.

As we may be unable to guarantee that exactly 50% of participants overlap between two executions, we can increase our tolerance for entropy loss by inviting more participants and building a buffer to accommodate overlaps in a range, e.g., 40–60%. Using data in Figure 7, this information is convenient to gather for $n = 10$. That is, if we increase the fraction of overlapping spectators, single-participation targets are most at risk. The converse is true if the overlap decreases – the target suffers less exposure from participating one evaluation. Table 1 summarizes the results. This means that performing two executions in the worst case costs a participant entropy loss 1.5 times higher than if only a single computation is executed. As a result, with the target entropy loss of 5% and 1%, we need to increase the number of spectators from 5 and 24 to 7 and 33, respectively.

We note that our analysis of repeated executions applies only when the inputs of the participants in the overlapping set of participants do not change. And if the executions are distant enough in time that the participants' inputs significantly change, they would no longer be treated as repeated dependent executions.

In the full version of the text [8] we conduct additional two-evaluation experiments, such as adjusting the number of shared spectators while maintaining a fixed number of unique spectators. Furthermore, we extend our analysis to three or more executions.

## 6 CONCLUSIONS AND RECOMMENDATIONS

In this work we study information disclosure associated with revealing the output of average salary computation on private inputs. Using the framework of [1], we analyze the function and derive several information-theoretic properties associated with the computation. Inputs are modeled using several discrete and continuous distributions, leading to multiple interesting conclusions about their entropy loss. We expand the scope to multiple executions on related inputs and determine the best configurations that minimize information disclosure. This leads to the following recommendations for computation designers:

- The amount of information disclosure about a target is independent of adversarial inputs. It was also experimentally shown to be independent of distribution parameters for three different distributions and analytically shown for normal distribution. All examined distributions produce nearly identical entropy loss curves.
- One can reduce the amount of entropy loss to a desired level by increasing the number of participants. For example, the computation designer can advertise at most 5% or 1% maximum entropy loss for the average salary application, which will require recruiting 6 or 25, respectively, non-adversarial participants when running only a single evaluation.
- In the presence of repeated computations, information disclosure about inputs continues for both participants who stay and participants who leave. With two executions, protection is the largest with 50% overlap in the participants, while both a small overlap and an overwhelming overlap result in undesirable information disclosure about different types of participants (i.e., those who stay vs. those who leave).
- With more executions, pairwise overlaps sizes determine information disclosure. For 3 and 4 executions, optimal configurations have overlap sizes near 1/3 of the number of participants.
- Information disclosure about participants' inputs can still be kept at a desirable level by enrolling enough participants and restricting percentage of reused inputs to be in a desired range. For example, with two executions and following the

guidelines of the keeping the overlap near 50%, the number of non-adversarial input contributors needs to be at least 8 to meet the target of 5% information loss.

## ACKNOWLEDGMENTS

This work was supported in part by a Google Faculty Research Award and NSF grants 2112693 and 2213057. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the funding sources.

## REFERENCES

[1] P. Ah-Fat and M. Huth. 2017. Secure Multi-party Computation: Information Flow of Outputs and Game Theory. In *POST*. 71–92.

[2] P. Ah-Fat and M. Huth. 2019. Optimal Accuracy-privacy Trade-off for Secure Computations. *IEEE Transactions on Information Theory* 65, 5 (2019), 3165–3182.

[3] P. Ah-Fat and M. Huth. 2020. Protecting Private Inputs: Bounded Distortion Guarantees With Randomised Approximations. *PoPETS* 2020, 3 (2020), 284–303.

[4] P. Ah-Fat and M. Huth. 2020. Two and Three-Party Digital Goods Auctions: Scalable Privacy Analysis. arXiv preprint arXiv:2009.09524.

[5] M. Alvim, K. Chatzikokolakis, A. McIver, C. Morgan, C. Palamidessi, and G. Smith. 2014. Additive and multiplicative notions of leakage, and their capacities. In *IEEE Computer Security Foundations Symposium*. 308–322.

[6] M. Alvim, K. Chatzikokolakis, C. Palamidessi, and G. Smith. 2012. Measuring information leakage using generalized gain functions. In *IEEE Computer Security Foundations Symposium*. 265–279.

[7] M. Alvim, A. Scedrov, and F. Schneider. 2014. When Not All Bits Are Equal: Worth-Based Information Flow.. In *POST*. 120–139.

[8] A. Baccarini, M. Blanton, and S. Zou. 2022. Understanding Information Disclosure from Secure Computation Output: A Study of Average Salary Computation. arXiv preprint arXiv:2209.10457.

[9] R. Barakat. 1976. Sums of Independent Lognormally Distributed Random Variables. *Journal of the Optical Society of America* 66, 3 (1976), 211–216.

[10] N. Beaulieu, A. Abu-Dayya, and P. McLane. 1995. Estimating the Distribution of a Sum of Independent Lognormal Random Variables. *IEEE Transactions on Communications* 43, 12 (1995), 2869–2873.

[11] N. Beaulieu and Q. Xie. 2004. An Optimal Lognormal Approximation to Lognormal Sum Distributions. *IEEE Transactions on Vehicular Technology* 53, 2 (2004), 479–489.

[12] A. Bhowmick, D. Boneh, S. Myers, and K. Tarbe. 2021. The Apple PSI system. https://www.apple.com/child-safety/pdf/Apple_PSI_System_Security_Protocol_and_Analysis.pdf.

[13] Boston Women's Workforce Council (BWWC) 2017. 2016 Report. https://htv-prod-media.s3.amazonaws.com/files/bwwc-report-final-january-4-2017-1483635889.pdf.

[14] Boston Women's Workforce Council (BWWC) 2018. 2017 Report. https://www.boston.gov/sites/default/files/document-file-01-2018/bwwc_2017_report.pdf.

[15] S. Bu, L. Lakshmanan, R. Ng, and G. Ramesh. 2006. Preservation of patterns and input-output privacy. In *IEEE International Conference on Data Engineering*. 696–705.

[16] C. Caiado and P. Rathie. 2007. Polynomial Coefficients and Distribution of the Sum of Discrete Uniform Variables. In *SSFA*.

[17] L. Cao, T. Tong, D. Trafimow, T. Wang, and X. Chen. 2022. The A Priori Procedure for estimating the mean in both log-normal and gamma populations and robustness for assumption violations. *Methodology* 18, 1 (2022), 24–43.

[18] M. Cheraghchi. 2019. Expressions for the Entropy of Basic Discrete Distributions. *IEEE Transactions on Information Theory* 65, 7 (2019), 3999–4009.

[19] D. Clark, S. Hunt, and P. Malacaria. 2002. Quantitative analysis of the leakage of confidential data. *Electronic Notes in Theoretical Computer Science* 59, 3 (2002), 238–251.

[20] F. Clementi and M. Gallegati. 2005. Pareto's law of income distribution: Evidence for Germany, the United Kingdom, and the United States. In *Econophysics of Wealth Distributions*. Springer, 3–14. https://doi.org/10.1007/88-470-0389-X_1

[21] B. Cobb, R. Rumí, and A. Salmerón. 2012. Approximating the Distribution of a Sum of Log-normal Random Variables. *Statistics and Computing* 16, 3 (2012), 293–308.

[22] T. Cover and J. Thomas. 2006. *Elements of Information Theory*. Wiley-Interscience.

[23] D. Denning. 1982. *Cryptography and data security*. Addison-Wesley Reading.

[24] V. Deshpande, L. Schwarz, M. Atallah, M. Blanton, and K. Frikken. 2011. Outsourcing manufacturing: Secure price-masking mechanisms for purchasing component parts. *Production and Operations Management* 20, 2 (2011), 165–180.

[25] V. Deshpande, L. Schwarz, M. Atallah, M. Blanton, K. Frikken, and J. Li. 2005. Secure Collaborative Planning, Forecasting and Replenishment (SCPFR). CERIAS Tech Report 2006-65.

[26] V. Deshpande, L. Schwarz, M. Atallah, M. Blanton, K. Frikken, and J. Li. 2006. Secure Collaborative Planning, Forecasting and Replenishment (SCPFR). In *Multi-Echelon/Public Applications of Supply Chain Management Conference*. 165–180.

[27] C. Dwork. 2008. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*. 1–19.

[28] C. Dwork and A. Roth. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science* 9, 3–4 (2014), 211–407.

[29] R. Evans and J. Boersma. 1988. The Entropy of a Poisson Distribution (C. Robert Appledorn). *SIAM Rev.* 30, 2 (1988), 314–317. https://doi.org/10.1137/1030059

[30] L. Fenton. 1960. The sum of log-normal probability distributions in scatter transmission systems. *IRE Transactions on Communications Systems* 8, 1 (1960), 57–67.

[31] Inpher 2024. Inpher. https://inpher.io/.

[32] M. Ion, B. Kreuter, A. Nergiz, S. Patel, S. Saxena, K. Seth, M. Raykova, D. Shanahan, and M. Yung. 2020. On deploying secure computing: Private intersection-sum-with-cardinality. In *IEEE EuroS&P*. 370–389.

[33] M. Iwamoto and J. Shikata. 2013. Information theoretic security for encryption based on conditional Rényi entropies. In *International Conference on Information Theoretic Security*. 103–121.

[34] B. Köpf and D. Basin. 2011. Automatically deriving information-theoretic bounds for adaptive side-channel attacks. *Journal of Computer Security* 19, 1 (2011), 1–31.

[35] R. Kotecha and S. Garg. 2017. Preserving output-privacy in data stream classification. *Progress in Artificial Intelligence* 6 (2017), 87–104.

[36] B. Kreuter. 2017. Secure Multiparty Computation at Google. Real World Crypto. Available from https://www.youtube.com/watch?v=ee7oRsDnNNc.

[37] A. Lapets, F. Jansen, K. Albab, R. Issa, L. Qin, M. Varia, and A. Bestavros. 2018. Accessible Privacy-Preserving Web-Based Data Analysis for Assessing and Addressing Economic Inequalities. In *ACM COMPASS*. 48:1–48:5.

[38] A. Lapets, N. Volgushev, A. Bestavros, F. Jansen, and M. Varia. 2016. Secure MPC for Analytics as a Web Application. In *SecDev*. 73–74.

[39] A. Lapets, N. Volgushev, A. Bestavros, F. Jansen, and M. Varia. 2016. *Secure Multi-Party Computation for Analytics Deployed as a Lightweight Web Application*. Technical Report BUCS-TR-2016-008. Boston University.

[40] Ligero Inc. 2022. Secure and Private Collaboration for Blockchains and Beyond. https://ligero-inc.com/. Last accessed: 2022-08-16.

[41] P. Mardziel, M. Hicks, J. Katz, and M. Srivatsa. 2012. Knowledge-oriented secure multiparty computation. In *Workshop on Programming Languages and Analysis for Security*. 1–12.

[42] J. Massey. 1994. Guessing and entropy. In *IEEE International Symposium on Information Theory*. 204.

[43] R. Mendes and J. Vilela. 2017. Privacy-preserving data mining: methods, metrics, and applications. *IEEE Access* 5 (2017), 10562–10582.

[44] A. Monreale and W. Wang. 2016. Privacy-preserving outsourcing of data mining. In *IEEE COMPSAC*, Vol. 2. 583–588.

[45] Nth party 2024. Nth party. https://www.nthparty.com/.

[46] Partisia 2024. Partisia. https://partisia.com/.

[47] A. Rastogi, P. Mardziel, M. Hicks, and M. Hammer. 2013. Knowledge inference for optimizing secure multi-party computation. In *SIGPLAN Workshop on Programming Languages and Analysis for Security*. 3–14.

[48] S. Schwartz and Y. Yeh. 1982. On the distribution function and moments of power sums with log-normal components. *Bell System Technical Journal* 61, 7 (1982), 1441–1462.

[49] D. Senaratne and C. Tellambura. 2009. Numerical Computation of the Lognormal Sum Distribution. In *IEEE GLOBECOM*. 1–6.

[50] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. 2017. Membership Inference Attacks Against Machine Learning Models. In *IEEE S&P*. 3–18.

[51] M. Skórski. 2019. Strong chain rules for min-entropy under few bits spoiled. In *IEEE International Symposium on Information Theory*. 1122–1126.

[52] G. Smith. 2009. On the foundations of quantitative information flow. In *FoSSaCS*. 288–302.

[53] L. Song and P. Mittal. 2021. Systematic Evaluation of Privacy Risks of Machine Learning Models. In *USENIX Security Symposium*. 2615–2632.

[54] W. Souma. 2002. Physics of personal income. In *Empirical Science of Financial Fluctuations*. 343–352.

[55] A. Walker, S. Patel, and M. Yung. 2019. Helping organizations do more without collecting more data. *Google Security Blog* (jun 2019). https://security.googleblog.com/2019/06/helping-organizations-do-more-without-collecting-more-data.html Last accessed: 2022-08-16.

[56] T. Wang and L. Liu. 2011. Output privacy in data mining. *ACM Transactions on Database Systems (TODS)* 36, 1 (2011), 1–34.

[57] J. Wu, N. Mehta, and J. Zhang. 2005. Flexible Lognormal Sum Approximation Method. In *IEEE GLOBECOM*. 3413–3417.