Variations and Extensions of Information Leakage Metrics with Applications to Privacy Problems with Imperfect Statistical Information

Shahnewaz Karim Sakib
Iowa State University, USA
Email: ssakib@iastate.edu

George T Amariucai
Kansas State University, USA
Email: amariucai@ksu.edu

Yong Guan
Iowa State University, USA
Email: guan@iastate.edu

2023 IEEE 36th Computer Security Foundations Symposium (CSF) | 979-8-3503-2192-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/CSF57540.2023.00007

***Abstract*—The conventional information leakage metrics assume that an adversary has complete knowledge of the distribution of the mechanism used to disclose information correlated with the sensitive attributes of a system. The only uncertainty arises from the specific realizations that are drawn from this distribution. This assumption does not hold in various practical scenarios where an adversary usually lacks complete information about the joint statistics of the private, utility, and the disclosed data. As a result, the typical information leakage metrics fail to measure the leakage appropriately. In this paper, we introduce multiple new versions of the traditional information-theoretic leakage metrics, that aptly represent information leakage for an adversary who lacks complete knowledge of the joint data statistics, and we provide insights into the potential uses of each. We experiment on a real-world dataset to further demonstrate how the introduced leakage metrics compare with the conventional notions of leakage. Finally, we show how privacy-utility optimization problems can be formulated in this context, such that their solutions result in the optimal information disclosure mechanisms, for various applications.**

***Index Terms*—Information Leakage, Subjective Leakage, Objective Leakage, Confidence Boost, Local Differential Privacy Leakage, Subjective Local Differential Privacy Leakage, Imperfect Statistical Information, Quantifying Privacy**

I. Introduction

Suppose a user, who wishes to remain anonymous, discloses “pop music” to be one’s preferred music genre. Based on this information alone, it may be deduced that the anonymous user belongs to an age demographic of 16-19 as a 2018 statistical survey shows the preference of such music genre among that specific age group [1]. Disclosing an apparently harmless piece of information can hence be used to infer, either correctly or incorrectly, a potentially sensitive attribute of a user.

In general, the observation of a disclosed variable correlated with a secret is expected to leak information about the secret. The disclosure can be intentional (e.g., over social media platforms) or can be the consequence of system design flaw (e.g., improperly secured communications or databases). Consider an eavesdropper monitoring the channel that a user uses to log into their private account. Even though the password is usually encrypted while transferring over the network, it is nevertheless possible for the eavesdropper to reduce the search space of the password by analyzing the timing of the packets as the packets are correlated with the keystrokes. Zhang and Wang [2] have shown a method to reduce the password search space by a factor of at least 250 using the keystroke timing.

Therefore, it is possible for the users’ information to extend beyond their expected privacy bound, essentially as a consequence of the platform design, even in the presence of various privacy safeguards. Such an extension leaks information regarding the sensitive attributes of a user. One of the fundamental topics of interest in computer security is how to quantify this privacy leakage. Various privacy measures have been proposed for quantifying the leakage previously, encompassing a broad range from information theory to data science. When using such metrics for providing security guarantees, it is essential to correctly specify their operational significance.

Various information leakage metrics have been proposed based on Shannon’s entropy and mutual information [3], [4], [5], [6]. Authors in [7] defined different one-shot measures of information leakage, namely maximal leakage, maximal realizable leakage, maximal correlation, and local differential privacy leakage. Another notion in information theory, known as min-entropy, has been studied extensively to define the information leakage [8], [9], [10]. Each of these metrics will only provide operational meaning when it is assumed that the probabilistic mechanism used to disclose information associated with the private data is completely known. Even the recently proposed measures of information leakage based on both f-information [11] and χ^2-information [12] also have the same assumption. For example, suppose the system utilizes Gaussian noise as a privacy measure. In that case, the metrics mentioned above assume that both the mean and variance of the noise are known, whereas only the samples drawn according to this distribution are not.

This assumption of complete knowledge of the joint distribution between the private and disclosed data (the end-to-end joint distribution) does not hold in practice. In general, even if the attacker tries to solve the same optimization problem as the data owner solved when deriving their optimal disclosure mechanism, the attacker no longer has access to the same context that the data owner used to learn their statistics. Since this results in mismatches between the real and the attacker’s computed statistics, the previous notions of information leakage do not provide an operational meaning. As the attacker only has an approximation of the joint distribution,

DOI 10.1109/CSF57540.2023.00007

we need metrics to accurately determine the probability of correct guessing by the adversary or the adversary's belief about how correct their guess is. It is plausible that the adversary may act if they believe they have enough information to infer a conclusion. Observe that here the correctness of the inference is sometimes insignificant. As long as the adversary is confident in their inferred conclusion, they will carry out the action. It is important to note that there can be two different ways to measure the confidence of the adversary: one is a *posterior evaluation*, following the acquisition of the disclosed data, and the other is an even more subjective *prior evaluation*.

The most pertinent work to our framework was performed by Chatzikokolakis et al. [13]. In that paper, the authors also considered the scenario where an adversary approximates the joint distribution based on their collection of samples. The authors subsequently analyzed the distribution of the *estimated* mutual information between the private and disclosed information. Eventually, they provided an estimation of the channel capacity based on this estimated mutual information.

The rest of the paper is organized as follows. In Section II, we discuss different state-of-the-art information leakage metrics. The system setup is delineated in Section III. Section IV discusses how to evaluate the correct probability that the adversary has a correct guess after observing the disclosed information. The proper evaluation of the attacker's belief of success is explained in Section V. Section VI analyzes the metric to capture the subjective evaluation of the belief of the attacker's success. Several optimization problems have been formulated in Section VII. We solve the proposed optimization problem and compare the optimized worst-case leakage values with the conventional notions of information leakage in Section VIII. We review several prior works in Section IX. Finally, in Section X, we summarize our paper and present the concluding remarks.

II. Established Measures of Information Leakage

Numerous leakage metrics have been proposed to represent information leakage in various scenarios. However, while defining each metric, identifying the correct output is essential to provide a contextual meaning. In this section, we shall group different state-of-the-art leakage metrics by the properties of the output these metrics capture.

Measurements of Uncertainty

The most straightforward way to define a privacy metric is to measure the uncertainty of an adversary's guess, and for a secure system, such uncertainty will be high. Shannon entropy [14] is the information-theoretic notion of measuring uncertainty, and most information-theoretic metrics are developed on this notion of entropy. Rényi entropy [15] is a generalization of Shannon entropy, with an additional parameter α. Depending on the value of α, Rényi entropy can represent different measures. Shannon entropy is a special case of Rényi entropy with $\alpha \to 1$. When $\alpha = 0$, we shall have Hartley (or max) entropy, and taking $\alpha \to \infty$ results in min-entropy.

Conditional entropy [16] is prevalent in communication networks, where data is transmitted over a noisy channel, and the receiver has to infer the transmitted data from the received data. The sender aims to keep the conditional entropy as small as possible, usually by using error-correction coding, to ensure that the receiver can have a better inference.

Cross entropy measures the average number of bits required to encode data originating from one distribution compared to encoding the same data with a different distribution [17]. For example, let us assume an event Z has been generated using the underlying probability distribution P, and an approximation of this distribution P is Q. The cross-entropy between P and Q, referred to as $H(P, Q)$, thus represents the number of bits required to represent this event Z when the encoding is done using the probability distribution Q instead of P.

Quantification of Information Gain

In various privacy setups, an adversary eavesdrops on the communication channel between the legitimate users to collect information to compromise the users' privacy. Thus, it is important to quantify how much information the observation has leaked about the private variable. Relative entropy (also known as Kullback–Leibler divergence, D_{KL}) is one such metric [18]. Some applications rely on obfuscating data, for example in smart metering. For such cases, relative entropy indicates how far the distribution of distorted data is from the true distribution.

Similarly, mutual information computes how much information is shared between the random variable observed by the adversary and the random variable representing the private information. If mutual information between these two random variables is high, the system will leak a considerable amount of information. In a sense, the mutual information metric and Kullback–Leibler divergence provide the same measure. However, mutual information is symmetric, while Kullback–Leibler divergence does not maintain the symmetry.

Additionally, we can extend the notion of mutual information to the scenarios where an adversary possesses prior information regarding the private variable and the observed variable. Such an extension will result in conditional mutual information, and this metric computes the amount of information about the private variable gained by the attacker upon observing the disclosed information, conditioned on the prior information [19]. Minor modification of mutual information will result in maximal information leakage [20]. This metric indicates the maximum amount of information an adversary can gain upon obtaining only a single observation. Finally, Fisher information [21] is a method of measuring the amount of information that an observed variable contains regarding the parameter of interest that models the distribution of the observed variable.

Data Indistinguishability

Data Indistinguishability indicates if an adversary can distinguish between two separate objects of interest. Differential privacy [22], formulated around two databases that differ by a single entry, has emerged as the consensus definition of publishing data in a privacy-preserving manner. This metric guarantees that the probability distributions of the result of a database query are approximately the same (within a small multiplicative factor of e^{ϵ}) for two neighboring databases. Even though differential privacy provides formal privacy guarantees, a no-free-lunch theorem shows that such guarantees degrade when data are correlated [23].

Relaxing the original notion of differential privacy, by using a small additive noise δ, results in an approximate differential privacy metric that allows a wider range of query types [24] than the original differential privacy metric, albeit in exchange for privacy. The usage of δ allows the analysis not to be overly-restrictive when evaluating two probability distributions on sets on which both distributions result in very small probabilities. For example, if one distribution's integral over a small set results in 10^{-10} while the other distribution's integration over the same set results in 10^{-15}, then their ratio is 10^{5}. This ratio is a lot larger than e^{ϵ} but still irrelevant as the integrals over both the distributions result in minimal values.

We can extend this approximate differential privacy to a framework where users consider the data aggregator untrusted and apply randomness to their own data before sending them to the central server [25]. Another possible extension of differential privacy can be done to a framework where the parameter controlling the generation of these datasets is protected instead of protecting the datasets themselves. Such an extension results in distributional privacy [26]. Finally, characterizing the distance between two datasets with distinguishability metrics $d_{\mathcal{X}}$, instead of Hamming distance, results in $d-\chi$-privacy [27].

III. System Setup

In this paper, we shall consider a setup where each user shares personal information in exchange for utility, such as gratifications that can be achieved by social interactions. In such a setup, each user will be comprised of several features. A user may wish to keep some of these features private while disclosing other feature values to get some form of utility. For example, a user might be reluctant to reveal their political affiliation. In contrast, they might be willing to let others know their food preferences so that they can get better restaurant recommendations. We shall refer to their political affiliation as a *private feature*, and to their food preferences as a *utility feature*. Additionally, we are also considering the existence of some *other* features that are neither utility nor private features.

Throughout the paper, we shall use random variable X_p to represent private features, and utility features will be represented by the random variable X_u. *Note that no restrictions are imposed on the correlation between X_u and X_p.* Additionally, we denote the rest of the features that are neither utility nor private by X and assume that X is correlated with both X_p and X_u. Finally, we denote the support of random variable X_p as $\mathcal{X}_p$, and support of X_u as $\mathcal{X}_u$.

Let us discuss an example to understand the correlation between X_p, X_u, and X. Consider the Netflix recommender system [28]. The utility of the platform is achieved by issuing a recommendation (X_u) for specific show to a user. However, as can be seen from [28], in addition to the show's features, the recommender considers a variety of user features, like the user interactions with the platform, the time of the day when the show is being watched, along with the device on which the user is watching the show, etc. These user features X are clearly related to the recommendation X_u, but they may also be related to the user's political affiliation X_p, which the user may expect to keep private. In fact, Narayanan et al. [29] showed that it is possible to infer users' political reference from their movie ratings. Therefore, instead of releasing either X_u (which in this case the user does not even know) or X, the user's best option may be to release Y, a perturbed version of X.

In essence, $(X_p, X_u) \to X \to Y$ form a Markov chain. Here, privacy is *inversely* proportional to the leaked information about X_p from Y, whereas utility is *directly* proportional to the gained information about X_u from Y.

Due to the Markov property, we get the following conditional distribution of Y given X, X_p, and X_u:

$$P_{Y|X,X_u,X_p} = P_{Y|X} P_{X|(X_u,X_p)}. \tag{1}$$

Particular instantiations of the Markov chain include the situations in which $X = X_u$ [30], when

$$P_{Y|X,X_u,X_p} = P_{Y|X_u} P_{X_u|X_p}, \tag{2}$$

or $X_u \subset X$ [31] in which case

$$P_{Y|X,X_u,X_p} = P_{Y|X} P_{X|X_p}. \tag{3}$$

When we have $X = X_u$, $X_p \to X_u \to Y$ forms the Markov chain whereas $X_u \subset X$ results in the $X_p \to (X_u, X) \to Y$ Markov chain.

In this paper, we are considering an adversary who has *bounded resources* and *lacks* complete statistical information about the joint distribution of the private, utility, and disclosed variables. It is possible for the adversary to gain information regarding the joint distribution through some side-channels. For example, the adversary can collect several (X_p, X_u, X, Y) tuples, possibly from some of their friends, and use these tuples to approximate the joint distribution.

Usually, the adversary approximates the true joint distribution between X_p and Y, $P(X_p, Y)$, as $Q(X_p, Y)$ based on their collection of (X_p, X_u, X, Y) tuples. We are assuming that the adversary knows the correct initial distribution of X_p, P_{X_p}. Thus, the uncertainty will arise due to the lack of the knowledge of $P_{Y|X_p}$ and consequently the adversary will approximate $P_{Y|X_p}$ as $Q_{Y|X_p}$.

The adversary can learn $Q_{Y|X_p}$ in several ways. As a matter of fact, the uncertainty about $Q_{Y|X_p}$ arises from two sources. One of them is the privacy mechanism, $P_{Y|X}$, while the other one is the likelihood of X given X_p, $P_{X|X_p}$. It is possible that the adversary may possess the complete knowledge of either one of them. However, in most cases, the adversary lacks the perfect knowledge of the statistics of both of these distributions. Therefore, depending on the application domain and the knowledge of the adversary, they can either learn the privacy mechanism or the likelihood of X given X_p or both of these distributions directly from the collected tuples. Note that the adversary learns each of the distributions with possibly different resolution approximations. Accordingly, the adversary can learn $P_{Y|X}$ with certain accuracy and $P_{X|X_p}$ with an accuracy that is most probably different from the accuracy of the learned $P_{Y|X}$. Throughout the paper, we have assumed that the adversary only lacks the true knowledge of the privacy mechanism.

Once the adversary has $Q_{Y|X}$, they can compute $Q_{X_p|Y}$ as follows:

$$Q_{X_p|Y} = \frac{Q_{X_p,Y}}{Q_Y} = \frac{\Sigma_X \Sigma_{X_u} Q_{Y|X} P_{X|(X_p,X_u)} P_{X_p,X_u}}{\Sigma_X \Sigma_{X_u} \Sigma_{X_p} Q_{Y|X} P_{X|(X_p,X_u)} P_{X_p,X_u}}.$$

Note that, $P_{X_p|Y}$ can also be computed using the true privacy mechanism as follows: $P_{X_p|Y} = \frac{P_{X_p,Y}}{P_Y} = \frac{\Sigma_X \Sigma_{X_u} P_{Y|X} P_{X|(X_p,X_u)} P_{X_p,X_u}}{\Sigma_X \Sigma_{X_u} \Sigma_{X_p} P_{Y|X} P_{X|(X_p,X_u)} P_{X_p,X_u}}$.

Similar to an adversary, a utility provider also lacks the perfect knowledge of the privacy mechanism. Consequently, the utility provider approximates the privacy mechanism as $Q'_{Y|X}$. However, the utility provider is interested in inferring the correct value of X_u from Y. Therefore, they utilize collected (X_p, X_u, X, Y) tuples to approximate $P_{X_u|Y}$ as $Q'_{X_u|Y} = \frac{Q'_{X_u,Y}}{Q_Y} = \frac{\Sigma_X \Sigma_{X_p} Q'_{Y|X} P_{X|(X_p,X_u)} P_{X_p,X_u}}{\Sigma_X \Sigma_{X_u} \Sigma_{X_p} Q'_{Y|X} P_{X|(X_p,X_u)} P_{X_p,X_u}}$, and employ $Q'_{X_u|Y}$ to infer X_u.

To summarize, in our proposed setup, each user has both private (X_p) and utility (X_u) features. Additionally, we have also considered *other* features that are neither private nor provide any utility (X) and disclosed a perturbed version of these *other* features (Y). Both the adversary and the utility provider lack complete knowledge about the privacy mechanism. Thus, they get an approximation of the privacy mechanism based on their collected (X_p, X_u, X, Y) tuples. Subsequently, the adversary utilizes $Q_{X_p|Y}$ to infer X_p, whereas the utility provider employs $Q'_{X_u|Y}$ to guess X_u. Table I presents the summary of the notations used throughout the paper.

IV. True Evaluation of Attacker's Success

We shall have several *categories* of the privacy measures in our setup as the information leakage, in the proposed setup, depends on the approximated mechanism $Q_{X_p|Y}$. We shall begin by providing a measure to evaluate the *true* probability that the attacker made a correct guess regarding the value of X_p after observing Y.

Let us analyze the definition of *min-entropy leakage* first. This metric provides a one-shot measure for guessing X_p. For a blind guess, that is, without collecting any Y, the adversary will always choose such $x \in \mathcal{X}_p$ that will maximize the prior probability of X_p (i.e., P_{X_p}). This measure is known as min-entropy and defined by (4):

$$H_\infty(X_p) = -\log_2 \max_{x \in \mathcal{X}_p} P_{X_p}(x). \tag{4}$$

After observing Y, the adversary will believe that they can have a better guess than the blind guess, and the uncertainty in guessing the correct value of X_p is reduced. Therefore, the uncertainty in guessing X_p now is represented by the conditional min-entropy $H_\infty(X_p|Y)$. Finally, the min-entropy leakage, referred to as L($P_{X_p|Y}$), is defined as the difference between these two measures of entropy. The mathematical representation of the metric is shown in (5) [10]. We denote the support of Y by $\mathcal{Y}$, and we let $x_1^*(y) = \arg\max\limits_{x \in \mathcal{X}_p} P_{X_p|Y}(x|y)$, leading to

$$\begin{aligned} \mathrm{L}(P_{X_p|Y}) &= I_\infty(X_p; Y) = H_\infty(X_p) - H_\infty(X_p|Y) \\ &= H_\infty(X_p) + \log_2 \sum_{y \in \mathcal{Y}} P_Y(y) \max_{x \in \mathcal{X}_p} P_{X_p|Y}(x|y) \\ &= H_\infty(X_p) + \log_2 \sum_{y \in \mathcal{Y}} P_Y(y) P_{X_p|Y}(x_1^*(y)|y). \end{aligned} \tag{5}$$

Symbol	Meaning
X_p	Random variable to represent private features
X_u	Random variable to represent utility features
X	Random variable to represent features that are neither utility nor private
Y	Random variable to represent disclosed information
$\mathcal{X}_p$	Support of X_p
$\mathcal{X}_u$	Support of X_u
$\mathcal{Y}$	Support of Y
P_{X_p,X_u}	Original joint distribution between X_p and X_u
P_Y	Original distribution of Y
Q_Y	Approximated distribution of Y by adversary
$P_{Y\|X}$	Original privacy mechanism
$Q_{Y\|X}$	Approximated privacy mechanism by adversary
$Q'_{Y\|X}$	Approximated privacy mechanism by utility provider
$x_1^*(y)$	$\arg\max\limits_{x \in \mathcal{X}_p} P_{X_p\|Y}(x\|y)$
$x_2^*(y)$	$\arg\max\limits_{x \in \mathcal{X}_p} Q_{X_p\|Y}(x\|y)$
$x_3^*(y)$	$\arg\max\limits_{x \in \mathcal{X}_u} Q'_{X_u\|Y}(x\|y)$
u_{min}	Minimum utility of the system
δ_L	Minimum distance between original and approximated privacy mechanism
δ_U	Maximum distance between original and approximated privacy mechanism

TABLE I: Summary of notations

Observe that the measure of min-entropy leakage does not consider the disclosure mechanism, approximated by the adversary (i.e., $Q_{X_p|Y}$), in any capacity. Consequently, this measure is not applicable for the adversary who lacks the perfect knowledge of the privacy mechanism. Therefore, we need to provide a measure to accurately compute the *actual* information leaked by any system when an adversary lacks the perfect knowledge of the privacy mechanism. We shall refer to this measure of actual information leakage as *objective leakage*. Depending on the application scenarios and the characteristics of the adversary, we can have several classes of objective leakage.

Average Objective Leakage

Let us begin by discussing how to compute, on *average*, how much information has been leaked by Y. Adopting the same approach of computing min-entropy leakage, we identify $x \in \mathcal{X}_p$ that maximizes $Q_{X_p|Y}$. We denote this index as $x_2^*(y) =$

P_{X_p}	value
X_p = 1	0.3
X_p = 2	0.15
X_p = 3	0.35
X_p = 4	0.2

TABLE II: Example of initial distribution of X_p, P_{X_p}

$\arg\max_{x \in \mathcal{X}_p} Q_{X_p|Y}(x|y)$. Finally, we use this $x_2^*(y)$, instead of $x_1^*(y)$ in (5), to compute the objective leakage.

This measure gives the *actual* leakage of the system, averaged over all possible values of observations. Thus, we refer to this metric as average objective leakage (AOL). The mathematical formula for AOL is given by (6):

$$\begin{aligned} \text{AOL}(P_{X_p|Y}, Q_{X_p|Y}) &= H_\infty(X_p) \\ &+ \log_2 \sum_{y \in \mathcal{Y}} P_Y(y) P_{X_p|Y}(x_2^*(y)|y). \end{aligned} \tag{6}$$

Let us break down the definition to understand the meaning behind such a formulation. For each $y \in \mathcal{Y}$, $P_{X_p|Y}(x_2^*(y)|y)$ represents the *true* probability that the adversary made a correct guess about the value of X_p after observing Y. Multiplication with $P_Y(y)$ provides the properly scaled measurement of the true probability. Eventually, summing over all possible values of y gives us the average scaled measurement of the probability that the adversary's guess is correct.

Issa et al. [7] introduced a framework to allow an adversary to have multiple guesses instead of a single guess. The measure of average objective leakage can easily be extended to a multiple guessing framework. After observing Y, we shall let adversary have k *independent* guesses to predict X_p, instead of a single guess. To measure the average value of objective leakage in this multiple guessing framework, we extend the notion of average objective leakage to k-average objective leakage (k-AOL).

Observe that $H_\infty(X_p)$ provides a measure of initial uncertainty in guessing X_p when the adversary is allowed to have a single-blind guess. For measuring the initial uncertainty in the multiple guessing framework, we need to extend the measure of min-entropy to k guesses. The adversary can exploit the knowledge of P_{X_p} to construct the k blind guesses to maximize the probability of having the correct value of X_p. The adversary can sort P_{X_p} according to the probability of each value of X_p and subsequently use the first k indices of X_p as k independent guesses. For example, if P_{X_p} is represented by Table II and the adversary makes $k = 2$ guesses, then they will guess $X_p = 3$ and $X_p = 1$ as these two values of X_p have the highest two probabilities. The mathematical formulation of min-entropy in multiple guessing framework is shown in (7).

$$H_\infty^k(X_p) = -\log_2(\sum_{i=1}^{k} P_{X_p}(x_0^*(i))) \tag{7}$$

Here, $x_0^*(i)$ is the value of X_p corresponding to the i-th largest $P_{X_p}(x)$, such that $x_0^*(1) = \arg\max_x P_{X_p}(x)$. Note that, for the rest of the paper, we shall use $H_\infty(X_p)$ to indicate the initial uncertainty in guessing X_p when we allow the adversary to have a single guess and $H_\infty^k(X_p)$ will indicate the initial uncertainty when we let the adversary make k independent guesses.

Now we show how to measure AOL for k independent guesses. Initially, for a specific y, consider the probability of having a correct guess for the value of X_p for each independent guess. Afterward, sum the probabilities for all the k guesses to get the un-scaled measurement of true probability that the adversary made a correct guess. Then, scale the value by multiplying with $P_Y(y)$. Finally, summing over all possible y's and adding the log of the summation with $H_\infty^k(X_p)$ gives the average objective leakage for k independent guesses. The formula for this measure is given by (8):

$$\begin{aligned} \text{k-AOL}(P_{X_p|Y}, Q_{X_p|Y}) &= H_\infty^k(X_p) \\ &+ \log_2 \sum_{y \in \mathcal{Y}} P_Y(y) \sum_{i=1}^{k} P_{X_p|Y}(x_2^*(y,i)|y). \end{aligned} \tag{8}$$

Here, $P_{X_p|Y}(x_2^*(y,i)|y)$ indicates the true probability that the adversary has made a correct guess of X_p for that specific y during their i^{th} guessing attempt.

Maximum Objective Leakage

Average objective leakage provides an average guessing performance of the adversary. It is possible to have some realization of Y for which the probability of correct guessing is high, but averaging over all realizations reduces the weight of this leakage. However, if our X_p is sensitive data (e.g., medical records of an individual), we must consider the *maximum* information that can be leaked by the system and accordingly, we get maximum objective leakage (MaxOL).

We know that for each $y \in \mathcal{Y}$, $P_{X_p|Y}(x_2^*(y)|y)$ indicates the true probability of the attacker having a right guess regarding the value of X_p. We are only interested in measuring the maximum leakage the adversary can realize for their guess. Thus, we only need to consider maximizing such true probabilities over all possible values of y. Summing the $\log_2$ of such maximization with the initial uncertainty will result in the maximum objective leakage. The formula of maximum objective leakage for the one-shot measure is shown in (9), and (10) extends the one-shot measure to the multiple guessing framework:

$$\begin{aligned} \text{MaxOL}(P_{X_p|Y}, Q_{X_p|Y}) &= H_\infty(X_p) \\ &+ \log_2 \max_{y \in \mathcal{Y}} P_{X_p|Y}(x_2^*(y)|y), \end{aligned} \tag{9}$$

$$\begin{aligned} \text{k-MaxOL}(P_{X_p|Y}, Q_{X_p|Y}) &= H_\infty^k(X_p) \\ &+ \log_2 \max_{y \in \mathcal{Y}} \sum_{i=1}^{k} P_{X_p|Y}(x_2^*(y,i)|y). \end{aligned} \tag{10}$$

Minimum Objective Leakage

Minimum objective leakage (MinOL) indicates the lowest possible leakage the adversary can attain for their guess. Thus, this metric represents the best-case information leakage for the system designer. Formulas for one-shot measure and k-shots measure of minimum objective leakage are given by (11) and (12), respectively:

$$\begin{aligned}\text{MinOL}(P_{X_p|Y}, Q_{X_p|Y}) &= H_\infty(X_p) \\ &+ \log_2 \min_{y\in\mathcal{Y}} P_{X_p|Y}(x_2^*(y)|y),\end{aligned} \tag{11}$$

$$\begin{aligned}\text{k-MinOL}(P_{X_p|Y}, Q_{X_p|Y}) &= H_\infty^k(X_p) \\ &+ \log_2 \min_{y\in\mathcal{Y}} \sum_{i=1}^{k} P_{X_p|Y}(x_2^*(y,i)|y).\end{aligned} \tag{12}$$

Now we provide the operational meaning of the minimum objective leakage. While maximum objective leakage implies the worst-case information leakage for the designer, the minimum objective leakage indicates the best-case private information leakage scenario. However, if we substitute X_u for X_p, this becomes a measure of the worst-case utility gain for the utility provider. While designing the system, the designer does not know beforehand how much gain the utility provider will realize. Thus, the designer may consider the worst-case scenario, and accordingly, ensure that the minimum objective leakage of the system meets the utility requirement in this worst case.

V. True Evaluation of Attacker's Belief of Success

Heretofore, we have introduced measures to compute the true probability that the attacker made a correct guess about X_p after observing Y. However, those measures do not reflect the attacker's *belief* of being successful. Depending on the metric definition, it is possible to capture both *true* and *subjective* assessments of the attacker's belief of success. In this section, we shall discuss the measures to calculate the *true* estimation. We have termed the metrics that calculate this true evaluation of attacker's belief of success as *confidence boost*.

We already know that, based on the approximated mechanism $Q_{X_p|Y}$, the adversary makes the guess $x_2^*(y)$ for a particular y. Putting this $x_2^*(y)$ in $Q_{X_p|Y}$ gives the subjective evaluation of the attacker's belief. This belief indicates the probability with which the adversary *thinks* they have made a correct guess regarding the value of X_p for that specific value of y. Therefore, this metric is related to the *confidence gain* that the attacker believes they have achieved by observing Y.

Now we shall present the operational meaning of the confidence boost metric. This metric will be important if the adversary decides to perform an action based on their confidence. Suppose an adversary plans to perform a harmful action on an entity if such an individual has performed a specific action. Consequently, the adversary observes the behavior of said entity for a limited amount of time. It is plausible that the behavior of that particular individual during that limited time may not represent the usual behavior. However, if the attacker gets a high confidence boost, they will most likely perform the harmful action. Note that the correctness of the inference is not of utmost importance in this case. The adversary acts as long as their confidence boost is significant.

Let us provide an example to explain the application of the confidence boost metric. Consider a scenario where the police collect some public information that leaks several sensitive attributes of a specific user. This collected information supports the conclusion that this person is a criminal. Note here that the police are collecting public information through a mechanism that they do not know perfectly. Yet if they have high confidence in their decision, they will arrest that specific person irrespective of the correctness of their decision. In fact, having higher confidence in a wrong decision, in this scenario, can lead to potentially devastating consequences. Such an incorrect inference will not only cause a significant personal loss for that user but also cause considerable damage to the administration, probably in terms of several lawsuits.

Now that we have shown the application of the confidence boost metric, we shall provide the mathematical formulation of the metric. Similarly to *objective leakage*, we can have several classes of *confidence boost* as well.

Average Confidence Boost

Suppose we are interested in measuring the *average* true confidence boost the adversary gets after observing Y. To perform such a measurement, at first, for each value of $y \in \mathcal{Y}$, take the probability with which the adversary believes they have made a correct guess, and this belief is represented by $Q_{X_p|Y}(x_2^*(y)|y)$. Next, multiply the numeric value of the belief with the *true* marginal distribution of Y, $P_Y(y)$. Summing over all possible values of y and adding the final sum with the initial uncertainty of guessing X_p will provide the measure of the average confidence boost (ACB) of the adversary. The mathematical formulation for this measure is given by (13), and (14) extends this one-shot measure to multiple guessing framework:

$$\begin{aligned}\text{ACB}(P_{X_p|Y}, Q_{X_p|Y}) &= H_\infty(X_p) \\ &+ \log_2 \sum_{y\in\mathcal{Y}} P_Y(y) Q_{X_p|Y}(x_2^*(y)|y),\end{aligned} \tag{13}$$

$$\begin{aligned}\text{k-ACB}(P_{X_p|Y}, Q_{X_p|Y}) &= H_\infty^k(X_p) \\ &+ \log_2 \sum_{y\in\mathcal{Y}} P_Y(y) \sum_{i=1}^{k} Q_{X_p|Y}(x_2^*(y,i)|y).\end{aligned} \tag{14}$$

Maximum Confidence Boost

Recall from the previous section that when X_p corresponds to sensitive information, the system designer will need to consider the maximum information leakage that the adversary can realize. Such consideration will result in maximum confidence boost (MaxCB) for one-shot measure and k-maximum confidence boost (k-MaxCB) for independent k guesses of the adversary. The mathematical formulation of MaxCB and k-MaxCB are given by (15) and (16), respectively. Here, $\max_{y\in\mathcal{Y}} Q_{X_p|Y}(x_2^*(y)|y)$ indicates the maximum possible confidence boost the attacker can realize for any value of y. Observe that we are not multiplying this confidence boost with P_Y, as we did in the average measurement. Thus, MaxCB is a function of only $Q_{X_p|Y}$ for the adversary:

$$\begin{aligned}\text{MaxCB}(Q_{X_p|Y}) &= H_\infty(X_p) \\ &+ \log_2 \max_{y\in\mathcal{Y}} Q_{X_p|Y}(x_2^*(y)|y),\end{aligned} \tag{15}$$

$$\begin{aligned}\text{k-MaxCB}(Q_{X_p|Y}) &= H_\infty^k(X_p) \\ + \log_2 \max_{y\in\mathcal{Y}} &\sum_{i=1}^{k} Q_{X_p|Y}(x_2^*(y,i)|y).\end{aligned} \tag{16}$$

Minimum Confidence Boost

Finally, we extend the notion of confidence boost metric to compute both one-shot measure of minimum confidence boost (MinCB) and minimum confidence boost for multiple guessing framework (k-MinCB). The mathematical formulations are shown in (17) and (18), respectively. Similar to MaxCB, MinCB is also a function of only $Q_{X_p|Y}$:

$$\begin{aligned}\text{MinCB}(Q_{X_p|Y}) &= H_\infty(X_p) \\ &+ \log_2 \min_{y\in\mathcal{Y}} Q_{X_p|Y}(x_2^*(y)|y),\end{aligned} \tag{17}$$

$$\begin{aligned}\text{k-MinCB}(Q_{X_p|Y}) &= H_\infty^k(X_p) \\ + \log_2 \min_{y\in\mathcal{Y}} &\sum_{i=1}^{k} Q_{X_p|Y}(x_2^*(y,i)|y).\end{aligned} \tag{18}$$

Interestingly, the minimum confidence boost, both the one-shot and k-shots measure, capture the characteristics of such an adversary who is not at all confident about their approximated mechanism and always considers the worst-case output. Thus, the minimum confidence boost measure represents the confidence boost that a *pessimistic* adversary will gain upon observing the disclosed variable Y.

VI. Subjective Evaluation of Attacker's Belief of Success

Formerly, we have defined the measures for both the proper evaluation of the attacker's success and the true evaluation of the attacker's belief of success. In this section, we shall extend the measures to reflect the confidence boost that an attacker *expects* to get by collecting additional Y.

Average Subjective Leakage

Recall that for each $y \in \mathcal{Y}$, $Q_{X_p|Y}(x_2^*(y)|y)$ indicates the probability with which the adversary believes they have made a correct guess for the value of X_p for that specific y. Multiplying this value of belief with the attacker's *approximated* marginal distribution Q_Y will result in the attacker's *expected* confidence boost. The formula for one-shot measure of average subjective leakage (ASL) is shown in (19), and (20) extends this one-shot measure to the multiple guessing framework:

$$\begin{aligned}\text{ASL}(Q_{X_p|Y}) &= H_\infty(X_p) \\ + \log_2 &\sum_{y\in\mathcal{Y}} Q_Y(y) Q_{X_p|Y}(x_2^*(y)|y),\end{aligned} \tag{19}$$

$$\begin{aligned}\text{k-ASL}(Q_{X_p|Y}) &= H_\infty^k(X_p) \\ + \log_2 \sum_{y\in\mathcal{Y}} Q_Y(y) &\sum_{i=1}^{k} Q_{X_p|Y}(x_2^*(y,i)|y).\end{aligned} \tag{20}$$

To understand what aspect of the measure this subjective leakage portraits, observe that the definition of the metric relates the probability with which the attacker thinks they have made the correct guess for each $y \in \mathcal{Y}$ to the attacker's approximated distribution of Y, $Q_Y(y)$. Thus, this metric represents an *apriori measurement* of the confidence boost an adversary will expect to get if they decide to collect more Y. This measure will enable the adversary to decide if the cost incurred during the process of gathering the disclosed information is worth the effort.

Let us again consider the example where the police collect public information that leaks private information about a user. The collected public information is consistent with the conclusion that the user is a criminal. However, the police may believe they do not have enough information to infer the conclusion with high confidence. The question now arises how many more resources the police is willing to invest in collecting additional public information that leaks private information about the user.

The average subjective leakage metric will enable the police to answer the question. Let us assume that the police have collected information about the user's behavior for a week and want to analyze if further information collection for another week is worth the effort. Consequently, they compute the average value of the subjective leakage using the gathered information, and if the average subjective leakage is minimal for a further collection of information, the police may conclude that further information collection may not boost the confidence any higher and may decide not to allocate more time and resources for the information collection.

Observe that we do not have notions of maximum subjective leakage or minimum subjective leakage. Subjective leakage allows the adversary to make a decision through the utilization of Q_Y. Recall that for measuring either the maximum or minimum of any proposed metrics, we have dropped the multiplication with the marginal distribution of Y as such multiplication does not have any operational meaning. Therefore, we do not have any mathematical formulation of either maximum subjective leakage or minimum subjective leakage.

Subjective Local Differential Privacy Leakage

The measures we have introduced heretofore deal with both the information gain and reduction in uncertainty of the adversary for guessing X_p after observing Y. Now, we shall introduce measures to capture the data distinguishing ability of the adversary upon observing the disclosed variable Y.

Adhering to the formulation provided by the authors in [7], we get (21) to represent the local differential privacy leakage (LDPL) of the original distribution $P_{X_p|Y}$. This LDPL measure computes the ratio of likelihoods for two values of X_p and a specific Y:

$$\text{LDPL}(P_{X_p|Y}) = \max_{\substack{y\in\mathcal{Y} \\ x,x'\in\mathcal{X}_p}} \log_2 \frac{P_{Y|X_p}(y|x)}{P_{Y|X_p}(y|x')}. \tag{21}$$

The local differential privacy leakage measure maximizes over Y and thus, implies the worst-case leakage for the system designer. It is possible to extend the metric to represent the average leakage of the system. We shall refer to such metric as average local differential privacy leakage (ALDPL):

$$\begin{aligned} &\text{ALDPL}(P_{X_p|Y}) \\ &= \Sigma_{y\in\mathcal{Y}} P_Y(y) \max_{x,x'\in\mathcal{X}_p} \log_2 \frac{P_{Y|X_p}(y|x)}{P_{Y|X_p}(y|x')}. \end{aligned} \tag{22}$$

Here, $\max_{x,x'\in\mathcal{X}_p} \log_2 \dfrac{P_{Y|X_p}(y|x)}{P_{Y|X_p}(y|x')}$ indicates the true log-likelihood ratio of distinguishing two elements of $\mathcal{X}_p$ for each $y \in \mathcal{Y}$. Subsequently, we compute the average of such ratios over the possible realizations of Y, to have the average local differential privacy leakage metric.

We can also extend the notion of local differential privacy leakage to represent the subjective evaluation of the adversary's belief of distinguishing two input values based on a specific set of realizations of the disclosed variable Y. Suppose we replace $P_{Y|X_p}$ with $Q_{Y|X_p}$ in (21). In that case, we get the attacker's *subjective evaluation* of the belief about their capability to differentiate two different values of X_p for a specific Y. Here we are maximizing over all possible values of Y, and accordingly, we shall refer to the measure as maximum subjective local differential privacy leakage (MaxSLDPL). The mathematical formulation is shown in (23).

$$\text{MaxSLDPL}(Q_{X_p|Y}) = \max_{\substack{y\in\mathcal{Y} \\ x,x'\in\mathcal{X}_p}} \log_2 \frac{Q_{Y|X_p}(y|x)}{Q_{Y|X_p}(y|x')} \tag{23}$$

The adversary will be highly confident that they can differentiate between x and x' if (23) results in a high value. Thus, this metric also represents the *confidence* of the attacker.

Let us now analyze how the metric MaxSLDPL can let the police assess their conclusion that the specific user is a criminal. Police can compute $Q_{Y|X_p}$ beforehand from their collected (X_p, X_u, X, Y) tuples. Once they observe a specific behavior of interest, specified by Y, from the particular user, straightaway they can employ MaxSLDPL metric to measure how much difference this particular realization of Y has made to the belief of the police regarding the user being guilty. Here, x can represent the scenario where the user is guilty of a crime, and x' can indicate those situations where the user is innocent. If the value of MaxSLDPL is high and positive, the police will be more confident in their conclusion that the user is indeed a criminal while a negative value of the difference will reduce the confidence of police in their conclusion.

We can also formulate minimum subjective local differential privacy leakage (MinSLDPL) which captures the characteristics of a pessimistic adversary. The definition is shown in (24):

$$\text{MinSLDPL}(Q_{X_p|Y}) = \min_{y\in\mathcal{Y}} \max_{x,x'\in\mathcal{X}_p} \log_2 \frac{Q_{Y|X_p}(y|x)}{Q_{Y|X_p}(y|x')}. \tag{24}$$

Finally, we can also have the average measure of the subjective local differential privacy leakage metric. Depending on the definition of the metric, it is possible to capture both the *true estimation* and *subjective belief* of the confidence boost of the adversary to distinguish between two inputs of $\mathcal{X}_p$.

We have proposed the objective average subjective local differential privacy leakage (OASLDPL) to represent the true confidence boost of the adversary to distinguish between two input values for a fixed observed Y. The mathematical formulation is shown in (25):

$$\begin{aligned} &\text{OASLDPL}(P_{X_p|Y}, Q_{X_p|Y}) \\ &= \Sigma_{y\in\mathcal{Y}} P_Y(y) \max_{x,x'\in\mathcal{X}_p} \log_2 \tfrac{Q_{Y|X_p}(y|x)}{Q_{Y|X_p}(y|x')}. \end{aligned} \tag{25}$$

Note that we are multiplying the attacker's belief about their capability to differentiate two input values (i.e., $\max_{x,x'\in\mathcal{X}_p} \log_2 \dfrac{Q_{Y|X_p}(y|x)}{Q_{Y|X_p}(y|x')}$) with P_Y. Thus, similar to confidence boost metrics, this multiplication indicates the *true* confidence boost that adversary realizes for distinguishing between two inputs upon observing Y.

Similarly to subjective leakage, if we multiply $\max_{x,x'\in\mathcal{X}_p} \log_2 \dfrac{Q_{Y|X_p}(y|x)}{Q_{Y|X_p}(y|x')}$ with Q_Y we shall get a metric representing the attacker's expectation of their ability to differentiate two input values upon further collection of Y. We have termed the metric as subjective average subjective local differential privacy leakage (SASLDPL) and the formulation is given by (26):

$$\begin{aligned} &\text{SASLDPL}(Q_{X_p|Y}) \\ &= \Sigma_{y\in\mathcal{Y}} Q_Y(y) \max_{x,x'\in\mathcal{X}_p} \log_2 \tfrac{Q_{Y|X_p}(y|x)}{Q_{Y|X_p}(y|x')}. \end{aligned} \tag{26}$$

VII. Problem Setup

Heretofore, we have introduced various information leakage metrics and explained the application scenario of each of the introduced metrics. In this section, we begin by summarizing the intuition behind the formulation of each metric. Afterward, we shall present the problem formulation utilizing these different notions of information leakage metrics.

The importance of the true evaluation of the attacker's success (i.e., objective leakage) is apparent. Such a measure will indicate the correctness of the inference made by the adversary upon observing the disclosed information. The subjective leakage measures will enable the adversary to decide if further information collection for a specific individual is worth the effort. For example, let us assume an adversary tracked an individual for a certain period and collected information about the said individual's behavior. Afterward, the adversary wants to determine whether to keep collecting information about the specific user. For this, the adversary may compute the expected gain that can be achieved by a further collection of information and check if the gain is significant or not. The subjective leakage measures will facilitate such decision-making. On the other hand, confidence boost metrics measure the true boost of belief for an adversary. These metrics can be of significant importance if the adversary decides to perform an action based on her confidence. Upon collecting information, an adversary may get a significant confidence boost on an incorrect inference. However, as the confidence boost is high, the adversary will make a decision based on that wrong inference. The adversary may carry out the unpleasant action simply because of the confidence in their inferred result (such an attacker is referred to as *"robber"* in [32]).

Comparison with g-leakage Framework

Let us compare the proposed metrics to the g-leakage framework of [33]. In the g-leakage framework, the authors introduced a gain function g to quantify how close the adversary's guess is to the true secret. This measure, nonetheless, still assumes the perfect knowledge of the privacy mechanism. However, the measure appropriately identifies that an adversary can gain information even if their guess is slightly wrong. Such a framework is different from our setup. The *objective leakage* measure indicates the *true* probability that the adversary has made a correct guess. For computing the objective leakage, we need to analyze that specific guess of the adversary. Such a guess is made by analyzing the approximated privacy mechanism. The same conclusion also holds for the leakage measures that evaluate the attacker's belief of success. For each of the evaluations of the attacker's belief of success, we have quantified either the subjective or true *gain in confidence* for the adversary. Such a measure is performed to understand the behavior of an adversary and whether such an adversary would take actions that can lead to potentially serious consequences. The measure of g-leakage does not quantify the gain in confidence for the adversary but rather the partial gain that the adversary achieves through their incorrect guess.

Utility Measurement

Recall that the utility provider infers X_u from Y based on their collection of (X_p, X_u, X, Y) tuples. They approximate the true distribution $P_{X_u|Y}$ as $Q'_{X_u|Y}$ and afterward, for each $y \in \mathcal{Y}$, the utility provider guesses $x \in \mathcal{X}_u$ that maximizes the probability of having a correct guess of X_u. We have referred to this guess as $x_3^*(y) = \underset{x \in \mathcal{X}_u}{\arg\max}\; Q'_{X_u|Y}(x|y)$ (See Table I). Now, we need to identify the correct metric that properly reflects the gain of the utility provider. If we compute the various minimum one-shot measures that are introduced in the paper, for such a utility provider, we get (27), and (28):

$$\begin{aligned}\text{MinOL}(P_{X_u|Y}, Q'_{X_u|Y}) = H_\infty(X_u)\\ + \log_2 \min_{y \in \mathcal{Y}} P_{X_u|Y}(x_3^*(y)|y),\end{aligned} \tag{27}$$

$$\begin{aligned}\text{MinCB}(Q'_{X_u|Y}) = H_\infty(X_u)\\ + \log_2 \min_{y \in \mathcal{Y}} Q'_{X_u|Y}(x_3^*(y)|y).\end{aligned} \tag{28}$$

Notice that *confidence boost* metric is a function of $Q'_{X_u|Y}$. For each $y \in \mathcal{Y}$, $Q'_{X_u|Y}(x_3^*(y)|y)$ indicates the utility provider's subjective evaluation of the probability with which they think they have inferred the correct value of X_u. Hence, if we measure *confidence boost* metric in this scenario, we shall get the confidence boost the utility provider obtains through a collection of (X_p, X_u, X, Y) tuples. Even though such measurement can have applications in decision making, confidence boost is not a suitable measure for utility.

The *objective leakage* on the other hand puts $x_3^*(y)$ in the correct distribution $P_{X_u|Y}$, and computes the corresponding leakage, as shown in (27). Thus, the objective leakage metric accurately represents the actual leakage of the system for the utility provider, and accordingly, we adopt objective leakage for computing the utility. We have seen in section IV that there are several classes of the objective leakage. Depending on the applications, the designer may employ any of them as the utility measure. However, typically the designer will be concerned about the worst-case leakage the utility provider can realize, and in that scenario, *minimum objective leakage*, as shown in (27), provides the accurate measure of the utility.

Problem Formulation

For our problem formulation, we are considering a system designer whose objective is to design a disclosure mechanism such that Y leaks minimal information about X_p while revealing a significant amount of information about X_u. These two conditions are contradictory to each other. Thus, we shall have a constrained optimization problem, and the solution of the optimization problem will result in a mechanism that ensures the information leakage between X_p and Y is minimized while maintaining the utility constraint.

In the previous subsection, we have explained the measure of utility. To ensure the usability of the system, the designer needs to ensure that the utility of the designed system is higher than a nominal utility u_{min}. Therefore, we have the utility constraint as $\mathcal{U}(X_u, Y) \ge u_{min}$.

Subsequently, we shall develop another set of constraints for the space of $Q_{X_p|Y}$. We have specified previously that the adversary will collect several (X_p, X_u, X, Y) tuples and approximate $P_{X_p|Y}$ as $Q_{X_p|Y}$. However, recall that we have assumed that the adversary only lacks the perfect knowledge of the privacy mechanism, and thus, approximates the privacy mechanism, $P_{Y|X}$, as $Q_{Y|X}$. Let us assume that the adversary approximates the privacy mechanism as $Q_{Y|X}$ upon collecting several (X_p, X_u, X, Y) tuples. For each $X = x$, the adversary collects n samples of Y. Now, using Theorem 11.2.1 in [34], we can write:

$$\Pr(D_{KL}(P_{Y|X=x}, Q_{Y|X=x}) > \epsilon) \le 2^{-n(\epsilon - |\mathcal{X}| \frac{log(n+1)}{n})}. \tag{29}$$

Note that (29) ensures that $Q_{Y|X}$ converges to $P_{Y|X}$, with probability 1, when $n \to \infty$. Now, from Pinsker's inequality [35], we know that:

$$d_{TV}(P_{Y|X=x}, Q_{Y|X=x}) \le \sqrt{\tfrac{1}{2} D_{KL}(P_{Y|X=x}, Q_{Y|X=x})}. \tag{30}$$

Here, d_{TV} indicates the total variation distance between two distributions.

Therefore, from (29), we can write:

$$\Pr(d_{TV}(P_{Y|X=x}, Q_{Y|X=x}) > \sqrt{\tfrac{1}{2}\epsilon}) \le 2^{-n(\epsilon - |\mathcal{X}| \frac{log(n+1)}{n})}. \tag{31}$$

Additionally, if we denote the Hellinger distance between $P_{Y|X=x}$ and $Q_{Y|X=x}$ as $h(P_{Y|X=x}, Q_{Y|X=x})$, then from Lemma 12.2 of [36], we get the following:

$$h^2(P_{Y|X=x}, Q_{Y|X=x}) \le d_{TV}(P_{Y|X=x}, Q_{Y|X=x}). \tag{32}$$

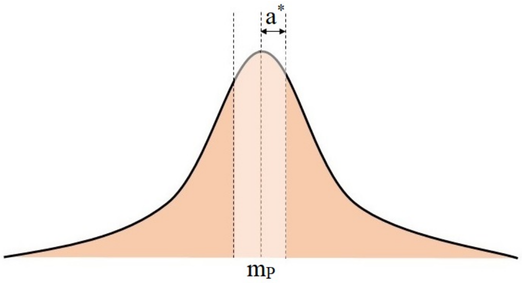


Fig. 1: Distribution of m_Q

Let us assume $Y_1 \sim P_{Y|X=x}$ and $Y_2 \sim Q_{Y|X=x}$. Furthermore, $\mathbb{E}[Y_1] = m_P$, $\mathbb{E}[Y_2] = m_Q$, $\text{Var}(Y_1) = \sigma_P^2$, and $\text{Var}(Y_2) = \sigma_Q^2$. Thus, if $m_P \neq m_Q$, Theorem 1 of [37] provides the following lower bound on $h^2(P_{Y|X=x}, Q_{Y|X=x})$:

$$h^2(P_{Y|X=x}, Q_{Y|X=x}) \geq 1 - \sqrt{1 - \frac{a^2}{a^2 + (\sigma_P + \sigma_Q)^2}}, \quad (33)$$

where $a = m_P - m_Q$. From (32) and (33), we can write:

$$d_{TV}(P_{Y|X=x}, Q_{Y|X=x}) \geq 1 - \sqrt{1 - \frac{a^2}{a^2 + (\sigma_P + \sigma_Q)^2}}. \quad (34)$$

Note that m_P is fixed, and according to the central limit theorem, m_Q will have a normal distribution with mean m_P and variance $\frac{\sigma_P^2}{n}$. Thus, for any a^*, $\Pr(a > a^*)$ will be represented by the darker region of Figure 1. If a^* is small, then such a probability will be high.

The adversary does have an incentive to have an approximated privacy mechanism as close as possible to the original privacy mechanism, such that they can have a maximum probability of having a correct guess. Due to a lack of the perfect knowledge of the true privacy mechanism, the adversary fails to achieve such a feat. However, the adversary still tries to have an approximated mechanism to maximize their probability of having a correct guess. Depending on the value of n (number of samples of Y for each value of $X = x$), (31) dictates that there exists an upper bound (δ_U) for $d_{TV}(P_{Y|X=x}, Q_{Y|X=x})$, whereas (34) shows the existence of lower bound (δ_L) for the same measure. Therefore, we need to optimize over all possible $Q_{Y|X}$ that are within these bounds of $P_{Y|X}$. From the perspective of a system designer, such a constraint is certainly important as lower values of both δ_L and δ_U mean that the approximated mechanism is closer to the true mechanism, and thus, the system can leak significant information regarding X_p.

Depending on the characteristics of the adversary, we can have several optimization problems. Let us consider designing the privacy mechanism for a *pessimistic* adversary where the system designer is interested in minimizing the *confidence boost*. As the adversary is of pessimistic nature, they will be doubtful about their approximated mechanism. Therefore, the designer needs to consider the *lowest* possible confidence boost that can be extracted from the approximated mechanism. Thus, the system designer needs to find the optimized privacy mechanism that minimizes the *minimum confidence boost*. Contrarily, if the designer were interested in devising the information disclosure mechanism for an *optimistic* adversary, the metric of interest would be the maximum confidence boost (MaxCB) as such an adversary will always be highly confident about their approximated mechanism. Finally, for a generic adversary, we shall have $\mathcal{L}(X_p, Y) = \text{ACB}(P_{X_p|Y}, Q_{X_p|Y})$.

The next step of the designer would be to find the optimized privacy mechanism $Q_{Y|X}$. Observe that the designer does not know beforehand which $Q_{Y|X}$ will be chosen by the adversary. The only information that the designer has is that $Q_{Y|X}$, chosen by the adversary, is at least δ_L away from $P_{Y|X}$, and within δ_U of $P_{Y|X}$. Therefore, the designer always needs to consider the worst case and consequently, find $P_{Y|X}$ that minimizes the worst-case value of $\mathcal{L}(X_p, Y)$. Accordingly, we have the following optimization problem:

$$\begin{aligned} \min_{P_{Y|X}} \max_{Q_{Y|X}} & \; \mathcal{L}(X_p, Y), \\ \text{such that } & \mathcal{U}(X_u, Y) \geq u_{min}, \\ \mathrm{d}_{TV}(P_{Y|X=x}, Q_{Y|X=x}) & \leq \delta_U (\forall x), \\ \text{and } d_{TV}(P_{Y|X=x}, Q_{Y|X=x}) & \geq \delta_L (\forall x). \end{aligned} \quad (35)$$

For solving the optimization problem, we have adopted a greedy approach. The details are given below.

- The algorithm *iteratively* finds the optimum $P_{Y|X}$ while a specific threshold condition is maintained. We initialize our step size, μ, to a random value. Next, we utilize the function *OPT_P* to find the optimum $P_{Y|X}$ at distance μ from the initial $P_{Y|X}$, and accordingly, we update our privacy mechanism to the new $P_{Y|X}$. At the same time, we keep track of the optimized worst-case leakage value. Afterward, we reduce the value of μ by half ($\mu = \frac{\mu}{2}$) and check if the reduced value of μ has further optimized the worst-case leakage. Such a check is done by computing the difference between the worst-case leakage values that we achieved for both μ and $\frac{\mu}{2}$. We keep repeating the process while the difference between these two leakages is higher than 0. The details are shown in Algorithm 1.
- Now we shall describe how *OPT_P* results in the optimum $P_{Y|X}$ for a fixed μ. We initialize $P_{Y|X}$ and generate a list of $\hat{P}_{Y|X}$ that are μ away from $P_{Y|X}$. Then, we use the function *OPT_Q* to find the optimum $\hat{P}_{Y|X}$ for the next iteration. We update our $P_{Y|X}$ to this value of $\hat{P}_{Y|X}$ and keep repeating the process while the difference between the previous leakage value and the current leakage value is higher than 0. The details are shown in algorithm 2.
- Finally, we shall discuss how *OPT_Q* finds the $\hat{P}_{Y|X}$ for the next iteration. Recall that we need to consider all $Q_{Y|X}$ that are at least δ_L away from $P_{Y|X}$ and within δ_U of $P_{Y|X}$. Therefore, for each $\hat{P}_{Y|X}$ in $list\_\hat{P}_{Y|X}$, we generate a list of $Q_{Y|X}$ that maintains our distance constraints. Afterward we compute the leakage value only for those $Q_{Y|X}$ that maintain our utility constraint and

Algorithm 1 Algorithm for solving the optimization problem

Input: δ_L, δ_U, u_{min}

Output: Optimum leakage value, optimum privacy mechanism

```
1: Initialize $P_{Y|X}$ to a transition probability matrix
2: new_P ← $P_{Y|X}$
3: $\mu$ ← a positive value
4: new_leak ← a large positive value
5: leak_diff ← a positive value
6: while leak_diff $> 0$ do
7:     current_leak ← new_leak
8:     $P_{Y|X}$ ← new_P
9:     (new_leak, new_P) ← OPT_P($\mu$, $\delta_L$, $\delta_U$, $u_{min}$, $P_{Y|X}$)
10:    leak_diff ← current_leak − new_leak
11:    $\mu \leftarrow \mu/2$
12: end while
13: return current_leak, $P_{Y|X}$
```

Algorithm 2 Algorithm for function OPT_P

Input: μ, δ_L, δ_U, u_{min}, $P_{Y|X}$

Output: Optimum leakage value, optimum $P_{Y|X}$ (for a specific μ)

```
1: function OPT_P($\mu$, $\delta_L$, $\delta_U$, $u_{min}$, $P_{Y|X}$)
2:     new_leak ← a large positive value
3:     leak_diff ← a positive value
4:     new_P ← $P_{Y|X}$
5:     while leak_diff $> 0$ do
6:         current_leak ← new_leak
7:         $P_{Y|X}$ ← new_P
8:         Generate list_$\hat{P}_{Y|X}$ that are $\mu$ away from $P_{Y|X}$
9:         (new_leak, new_P) ← OPT_Q (list_$\hat{P}_{Y|X}$, $\delta_L$, $\delta_U$, $u_{min}$)
10:        leak_diff ← current_leak − new_leak
11:    end while
12:    return current_leak, $P_{Y|X}$
13: end function
```

choose that $Q_{Y|X}$ that maximizes the leakage value. Once we have all the worst-case leakage values for each $\hat{P}_{Y|X}$ in $list\_\hat{P}_{Y|X}$, we choose the one that minimizes such maximization of the leakage value as our optimum $\hat{P}_{Y|X}$. The details of this step are shown in algorithm 3.

Note that step 8 of algorithm 2 calls for the generation of a list_$\hat{P}_{Y|X}$ that are μ away from $P_{Y|X}$. By this we mean that for each $x \in \mathcal{X}$ we generate k probability distributions $\hat{P}_{Y|X=x}$, each one at total variation distance μ from $P_{Y|X=x}$ (for this latter task, it is enough to randomly choose two values $y_i, y_j \in \mathcal{Y}$, and do $\hat{P}_{Y|X=x}(y_i) = P_{Y|X=x}(y_i) - \mu$, and $\hat{P}_{Y|X=x}(y_j) = P_{Y|X=x}(y_j) + \mu$, while ensuring the values are non-negative.). Here k is a compile-time constant, chosen to be lower than or equal to $\binom{|\mathcal{Y}|}{2}$.

A similar approach is applied to generate list_$Q_{Y|X}$ of step 3 of algorithm 3. Initially, we produce a subset of length m of $Q_{Y|X}$'s that are exactly at δ_U distance from a specific $\hat{P}_{Y|X}$ in the same manner. Afterward, we create another subset of $Q_{Y|X}$'s that are exactly $\delta_U - c$ distance from the $\hat{P}_{Y|X}$ (c is a small constant). We keep repeating the process till we reach δ_L and combine all the generated subsets to generate the list_$Q_{Y|X}$. Similar to k, m is also a compile-time constant.

Algorithm 3 Algorithm for function OPT_Q

Input: $list\_\hat{P}_{Y|X}, \delta_L, \delta_U, u_{min}$

Output: Optimum leakage value, optimum $\hat{P}_{Y|X}$

```
1: function OPT_Q($list\_\hat{P}_{Y|X}, \delta_L, \delta_U, u_{min}$)
2:     for each $\hat{P}_{Y|X}$ in the list_$\hat{P}_{Y|X}$ do
3:         Generate list_$Q_{Y|X}$ that are within [$\delta_L$, $\delta_U$] distance of $\hat{P}_{Y|X}$
4:         leak_list_P ← [ ]
5:         for each $Q_{Y|X}$ in list_$Q_{Y|X}$ do
6:             leak_list_Q ← [ ]
7:             if Utility constraint is maintained then
8:                 leak_Q ← $\mathcal{L}(P_{Y|X}, Q_{Y|X})$
9:                 Append leak_Q to leak_list_Q
10:            end if
11:        end for
12:        leak_max ← max(leak_list_Q)
13:        Append leak_max to leak_list_P
14:    end for
15:    leak_min ← min(leak_list_P)
16:    min_index ← leak_list_P.index(leak_min)
17:    min_P ← list_$\hat{P}_{Y|X}$[min_index]
18:    return leak_min, min_P
19: end function
```

Properties of the Proposed Metrics

In this subsection, we shall present several properties of the proposed metrics. We are only analyzing the properties of those metrics that are defined as averages over the range of outputs due to the continuous nature of the functions. Such measures are average subjective leakage, average objective leakage, and average confidence boost. For ease of explanation, we are assuming that the adversary is allowed to make a single guess upon observing Y instead of making k guesses.

Property 1. *$\max_{Q_{Y|X}}$ AOL, where $Q_{Y|X}$ is at distance at least $\delta_L > 0$ from $P_{Y|X}$, is always smaller than the min-entropy leakage (L).*

Proof. We know that $x_1^*(y)$ indicates the value of $x \in \mathcal{X}_p$ that maximizes $P_{X_p|Y}$ and $x_2^*(y)$ represents $x \in \mathcal{X}_p$ that maximizes $Q_{X_p|Y}$. When $\delta_L > 0$, $x_1^*(y)$ and $x_2^*(y)$ will refer to different values. As $x_1^*(y)$ always maximizes $P_{X_p|Y}$, $P_{X_p|Y}(x_1^*(y)|y)$ is always higher than $P_{X_p|Y}(x_2^*(y)|y)$. Therefore, maximum of average objective leakage will be lower than the min-entropy leakage. □

Property 2. *$\max_{Q_{Y|X}}$ ACB, where $Q_{Y|X}$ is at a distance between 0 and δ_U from $P_{Y|X}$ (that is, when $\delta_L = 0$), is always larger than or equal to the min-entropy leakage (L).*

Proof. From the definitions of both average confidence boost (shown in (13)) and min-entropy leakage (shown in (5)), we get the following:

$$\begin{aligned} &ACB(P_{X_p|Y}, Q_{X_p|Y}) - L(P_{X_p|Y}) \\ &= \log_2 \frac{\sum_y P_Y(y) Q_{X_p|Y}(x_2^*(y)|y)}{\sum_y P_Y(y) P_{X_p|Y}(x_1^*(y)|y)} \end{aligned}$$

When $\delta_L = 0$, the search space for $Q_{X_p|Y}$ always includes $P_{X_p|Y}$. Additionally, $x_2^*(y)$ always maximizes $Q_{X_p|Y}$. This leads to $\max_{Q_{X_p|Y}} \sum_y P_Y(y) Q_{X_p|Y}(x_2^*(y)|y) \geq \sum_y P_Y(y) P_{X_p|Y}(x_1^*(y)|y)$, thus the difference between ACB and L is always positive. □

Property 3. $\max_{Q_{Y|X}} ASL$*, where* $Q_{Y|X}$ *is at a distance between* 0 *and* δ_U *from* $P_{Y|X}$ *(that is, when* $\delta_L = 0$*), is always larger than or equal to the min-entropy leakage (L).*

Proof. From the definitions of both average subjective leakage (shown in (19)) and min-entropy leakage (shown in (5)), we get the following:

$$\begin{aligned} ASL(Q_{X_p|Y}) - L(P_{X_p|Y}) &= \log_2 \frac{\sum_y Q_Y(y) Q_{X_p|Y}(x_2^*(y)|y)}{\sum_y P_Y(y) P_{X_p|Y}(x_1^*(y)|y)} \\ &= \log_2 \frac{\sum_y \max_{x \in \mathcal{X}_p} Q_{XY}(x,y)}{\sum_y \max_{x \in \mathcal{X}_p} P_{XY}(x,y)} \end{aligned}$$

Now, since the search space for Q_{XY} always includes P_{XY} (recall that $\delta_L = 0$), it becomes clear that $\max_{Q_{XY}} \max_{x \in \mathcal{X}_p} Q_{XY}(x,y)$ is at least as large as the value of $\max_{x \in \mathcal{X}_p} P_{XY}(x,y)$, leading to a positive difference between ASL and L. □

VIII. Simulation Results

We shall begin the section by analyzing a real-world dataset to check if our proved properties hold. Afterward, we shall compute the worst-case leakage values with the optimized privacy mechanism that results from the optimization problem.

Dataset Description

The Iris Dataset of UCI Machine Learning Repository [38] is used as a real-world example dataset to further extend the analysis of the proposed metrics. The dataset includes 150 instances of three different iris classes: Iris-setosa, Iris-versicolor, and Iris-virginica. For each sample, four features were also measured: the length and width of the sepals and petals (in centimeters). For our analysis, we have selected the "Species" parameter as our private feature (X_p) and "PetalWidthCm" as the utility feature (X_u). The rest of the features (SepalLengthCm, SepalWidthCm, and PetalLengthCm) are treated as X. Observe that, in the dataset each species has 50 samples, and consequently, we have $H_\infty(X_p) = -\log_2(50/150) = 1.585$.

Convergence of Average Metrics

For this subsection, we are analyzing the convergence of our proposed metrics, such as the convergence of average subjective leakage to min-entropy leakage. Consequently, we need to generate Y from X by noise addition. Sharma et al. [39] discussed an optimal noise addition mechanism. This noise addition mechanism minimizes the mutual information between the private variable (X_p) and the disclosed variable (Y). The algorithm has two privacy parameters. The first parameter indicates when to add noise to X to increase privacy (referred to as γ), and the second parameter, β, indicates the utility loss upon the addition of noise. For our experiment, we have used $\gamma = 0.25$ and $\beta = 1.52$.

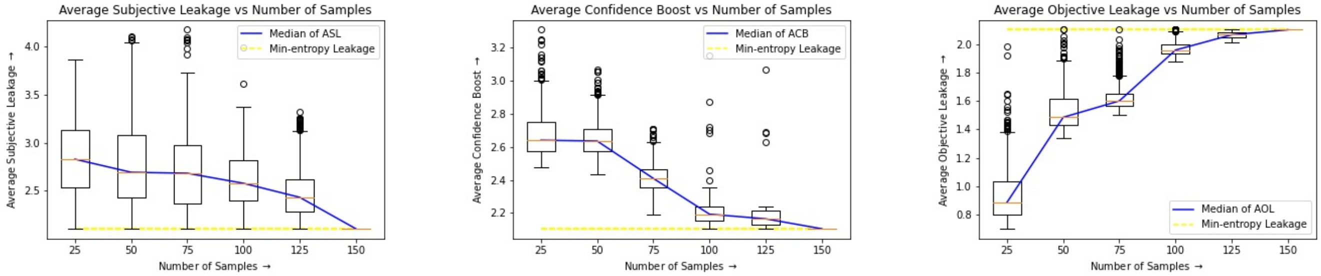
Observe that, throughout the paper; we have employed Bayesian inference to infer X_p from Y. Such inference requires the data to be divided into discrete bins. Thus, we have divided each feature of Y into *three* separate bins. We were interested in discretizing each of the features into equal-sized bins based on the quantile values. Accordingly, we performed quantile cut for each feature of Y. We performed the same operation on X as well. As both X and Y consist of three possible features, and each feature has three possible values, we have $27(3^3)$ possible values of both X and Y. Therefore, we have $P_{Y|X}$ as a 27×27 matrix. Note that we divided each feature of Y into three bins to keep the shape of the matrix $P_{Y|X}$ tractable as the paper is focused on analyzing the performance of the proposed matrix rather than dealing with a large matrix. The analysis is similar when the shape of the matrix of interest is large.

Recall that we are considering an adversary who approximates the privacy mechanism based on their collection of (X_p, X_u, X, Y) tuples. We adopted the method of Chatzikokolakis et al. [13] for such an approximation. Specifically, we have taken X and Y at their face value, and utilized the number of observation for approximating the privacy mechanism. Additionally, we have varied the number of collected tuples to simulate an adversary with different $Q_{Y|X}$. Figure 2 shows the box-plot of the variation of the proposed leakage measures for varying number of samples. The *blue* line indicates the median value for that specific instantiation. The details of the box-plot can be found in Table III.

For explanation, let us consider Figure 2a first. Note that $P_{Y|X}$ is fixed here, and $Q_{Y|X}$ is approximated by the adversary upon collecting a *fixed* number of (X_p, X_u, X, Y) tuples. If the adversary can collect a higher number of (X_p, X_u, X, Y) tuples, then their approximated privacy mechanism ($Q_{Y|X}$) will be closer to the original privacy mechanism ($P_{Y|X}$). Moreover, recall that average subjective leakage is defined as the maximum over $Q_{X_p|Y}$. As the adversary lacks perfect knowledge of the privacy mechanism, maximization over $Q_{X_p|Y}$ depends on the approximated privacy mechanism $Q_{Y|X}$, and the distance between $P_{Y|X}$ and $Q_{Y|X}$. When the adversary has access to a smaller number of samples, the distance between these two privacy mechanisms will be higher. A higher distance will result in a larger search space for $Q_{X_p|Y}$, and consequently, $\max_{Q_{X_p|Y}} Q_{X_p|Y}(x_2^*(y)|y)$ will be higher. Once we increase the number of samples, the distance starts to get lower, and correspondingly, we get a smaller search space for $Q_{X_p|Y}$. As the search space of $Q_{X_p|Y}$ gets smaller, $\max_{Q_{X_p|Y}} Q_{X_p|Y}(x_2^*(y)|y)$ becomes smaller and consequently results in a lower value of average subjective leakage. Finally, when the adversary gets access to all the

Number of Samples	AOL		ACB		ASL	
	Median	(1^{st} quartile, 3^{rd} quartile)	Median	(1^{st} quartile, 3^{rd} quartile)	Median	(1^{st} quartile, 3^{rd} quartile)
25	0.884	(0.769, 1.032)	2.639	(2.575, 2.746)	2.829	(2.531, 3.136)
50	1.486	(1.429, 1.614)	2.634	(2.572, 2.708)	2.693	(2.433, 3.080)
75	1.602	(1.564, 1.652)	2.409	(2.357, 2.465)	2.682	(2.373, 2.980)
100	1.962	(1.938, 2.000)	2.193	(2.153, 2.241)	2.578	(2.396, 2.818)
125	2.070	(2.053, 2.090)	2.165	(2.131, 2.215)	2.432	(2.281, 2.620)
150	2.105	(2.105, 2.105)	2.105	(2.105, 2.105)	2.105	(2.105, 2.105)

TABLE III: Median value and quartile tuples of proposed leakage measures on Iris dataset (Min-entropy leakage = 2.105)



(a) Average subjective leakage

(b) Average confidence boost

(c) Average objective leakage

Fig. 2: Box-plot of variation of leakage metrics with varying number of samples for the Iris dataset. The *blue* line indicates the variation of median values.

samples, $Q_{X_p|Y}$ becomes equal to $P_{X_p|Y}$, and thus the average subjective leakage also converges to min-entropy leakage. Such a variation is represented by the *blue* line of Figure 2a. Observe that such variation is also consistent with our property 3 where we have proved that $\max_{Q_{Y|X}}$ ASL is always larger than the min-entropy leakage. Using the same reasoning, we can explain both Figures 2b and 2c, which are compatible with property 2, and property 1, respectively. Moreover, from Table III, we can see that when the adversary does not have all of the (X_p, X_u, X, Y) tuples, the first quartile of both ASL and ACB is higher than the min-entropy leakage. Certainly, none of the observed values of either ASL and ACB falls below the min-entropy leakage, making the observation consistent with the leakage properties. A similar observation holds for AOL as well, where none of the observed values of AOL goes beyond the min-entropy leakage.

Optimized Worst-case Leakage Values

In this subsection, we shall initially discuss how we can estimate both δ_L and δ_U and how the proposed leakage measures are related to these values. Recall that δ_L indicates the lower bound between $P_{Y|X=x}$ and $Q_{Y|X=x}$ $(\forall x)$, whereas the upper bound is represented by δ_U. The adversary usually approximates the privacy mechanism based on their collection of (X_p, X_u, X, Y) tuples. The system designer does not know exactly how many original (X_p, X_u, X, Y) tuples can be collected by the adversary. Therefore, the best way would be to assume a range of values for such a collection of tuples. The smallest value of the range will result in an estimate of the upper bound (i.e., δ_U), and the largest value will result in the estimation of the lower bound (i.e., δ_L).

Let us discuss the effect of δ_L first. If $\delta_L = 0$, then it is possible for the adversary to have access to all the original (X_p, X_u, X, Y) tuples that were used to design the privacy mechanism and thus, have complete knowledge of the privacy mechanism. As a result, both $P_{Y|X}$ and $Q_{Y|X}$ refer to the same privacy mechanism [32]. In that case, we shall have the worst-case value of average objective leakage, which is min-entropy leakage (see Property 1). Additionally, both average subjective leakage and average confidence boost will be the same as the min-entropy leakage. However, in most practical circumstances, an adversary lacks this advantage which results in $\delta_L > 0$, and accordingly, *AOL* will be maximized when $Q_{Y|X}$ will be at exactly δ_L distance from $P_{Y|X}$.

Now let us analyze the effect of δ_U. For a fixed value of δ_L, a higher value of δ_U will result in a larger search space for $Q_{X_p|Y}$. A larger search space will result in a higher value for $\max_{Q_{X_p|Y}} Q_{X_p|Y}(x_2^*(y)|y)$. As both average subjective leakage and average confidence boost utilize $Q_{X_p|Y}(x_2^*(y)|y)$ in their definition, such a maximization will result in a higher value for both *ASL* and *ACB*.

Straightaway, we shall repeat the simulation where we let the adversary gather several input-output pairs, and based on the collection of (X_p, X_u, X, Y) tuples; they have the approximated privacy mechanism $Q_{Y|X}$. We have varied the number of collected tuples from 25 to 150. Of course, the system designer does not know how many samples the adversary gets. Hence, we assumed that the privacy system $P_{Y|X}$ is designed considering an adversary that can have anywhere between 25 and all of 150 (X_p, X_u, X, Y) tuples. The upper bound on the number of tuples implies that $\delta_L = 0$. For finding an estimate of δ_U, we initially generated Y using the noise addition mechanism of Sharma et al. [39], and let the adversary have access to random 25 (X_p, X_u, X, Y) tuples. By repeating the process several times, we found an estimate of $\delta_U = 0.75$.

Number of Samples	worst-case ASL	worst-case ACB
25	2.310	2.207
50	2.277	2.170
75	2.234	2.143
100	2.177	2.110
125	2.172	2.107
150	2.105	2.105

TABLE IV: Median of optimized worst-case metric values

Finally, we have solved the optimization problem, using $\delta_L = 0$ and $\delta_U = 0.75$, to achieve optimum $P_{Y|X}$. We compared the worst-case leakage values, obtained from the optimum $P_{Y|X}$, to the worst-case leakage values achieved by minimizing the mutual information between X_p and Y. The details of the median of the optimized worst-case leakage values are given in Table IV, and Figure 3 shows the comparative plot. Note that we have not solved the optimization problem when $\mathcal{L}(X_p, Y) = AOL$, as we have $\delta_L = 0$ meaning *AOL* will be maximized when $P_{Y|X} = Q_{Y|X}$.

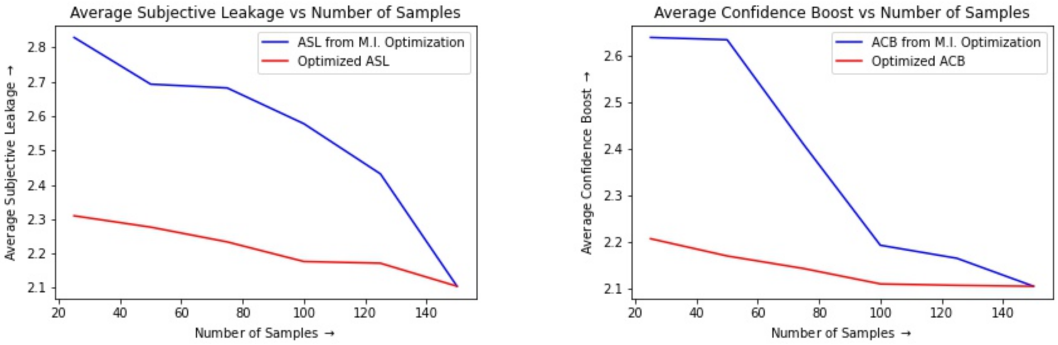
IX. Related Work

Information-theoretic measures, specifically Shannon entropy and mutual information based information leakage measures, have been studied extensively in former years [3]–[6]. Shannon entropy measures the average amount of information that a message contains, and mutual information between two random variables quantifies the amount of information gained about one variable by observing the other. Concisely, mutual information measures the correlation between two variables. Authors in [40] discussed how an attacker's belief change by observing the execution of a program whereas Hamadou et al. [41] unified the notion of belief and leakage for an adversary. In [40], the authors introduced a metric where an adversary observes the execution of a program and consequently updates their initial belief about the private variable. The authors did not consider the case where an adversary approximates the privacy mechanism. Authors in [41] nevertheless introduced the metrics to represent the belief of the attacker when they had different (and potentially wrong) initial beliefs regarding the distribution of the secret and consequently presented several properties to measure the accuracy and belief of the adversary. In the current paper, we assumed the approximation of the privacy mechanism. Moreover, we have also formulated an optimization problem that results in a privacy mechanism that minimizes the worst-case leakage. Such an optimization problem was not formulated in both [40] and [41]. Another prominent measure of information leakage is min-entropy leakage [8]–[10]. The definition of min-entropy leakage captures the reduction in uncertainty of guessing a secret once some information correlated with the secret is disclosed. Min-entropy is a specific case of Rényi entropy [15] with $\alpha = \infty$. Authors in [33] introduced *g-leakage*, a generalization of *min-entropy leakage*. The authors in [42] provided several axioms for information leakage. The notion of black-box estimation of leakage was introduced in [43]. In [44], the authors estimated the g-leakage via machine learning approaches and evaluated the performance of their approach via various experiments using k-nearest neighbors and neural network.

Authors in [45] introduced a single-shot measure of information leakage, known as maximal leakage. Additionally, in [7], they have also introduced various one-shot measures such as maximal realizable leakage, local differential privacy, maximal correlation, and maximal cost leakage. The authors in [46] and [47] showed that machine learning models are vulnerable to membership inference attacks. In [48], the authors measured the information leakage regarding the presence of an individual in a training dataset using conditional mutual information. Fisher information estimates the amount of information obtained about a parameter by observing a random variable whose characteristics depend on the said parameter. Hannun et al. [49] adopted Fisher information for defining information leakage, and later proposed a method to quantify the information leakage of the training data in a machine learning model. The authors in [50], [51] analyzed Fisher information as a privacy leakage measure to develop an optimum privacy-preserving policy. Various game-theoretic settings have been proposed to simulate the interactions between a utility provider and an adversary in the context of both information flow and differential privacy [52], [53], [54].

Differential privacy, introduced in [22], was formulated around two neighboring databases, namely two databases differing in a single entry. The tradeoff between utility and privacy, for a differentially private mechanism, has been studied extensively [55]–[60]. In a recent work, Desfontaines et al. [61] provided an extensive analysis on the variants of differential privacy. They divided the various notions of differential privacy into seven categories, depending on the aspect of the original definition that is being modified. The authors in [62] performed a similar extensive analysis of information leakage metrics.

The work of Chatzikokolakis et al. [13] is closely related to our framework. In the paper, the authors have forgone the assumption that the exact probabilities of the information disclosure mechanism are known and estimated the mutual information based on the collection of samples. Afterward, they provided an estimation of the channel capacity using the estimated mutual information. Several relevant works were performed in [11] and [12]. In both papers, the authors considered a database that comprises both private and public features, and the mechanism releases a distorted version of the public features. Thereafter, the authors devised a mechanism that minimizes the privacy-utility tradeoff. The authors utilized f-information as the measure of privacy in [11] and applied χ^2-information as a privacy measure in [12]. Note that both f-information and χ^2-information require that the joint distribution between public and disclosed variables are completely known. The authors, however, assumed that such assumptions might not hold in various applications and consequently provided an estimation of the privacy measures. Finally, they provided a bound on the error of the difference between the privacy measures, computed under exact and approximated mechanisms.



(a) Average subjective leakage comparison

(b) Average confidence boost comparison

Fig. 3: Plot of comparison of worst-case median leakage metrics with a varying number of samples for Iris dataset. The *blue* line shows the variation of the median value when the noise addition mechanism minimizes the mutual information between X_p and Y. The *red* line shows the variation of median values of the resultant worst-case leakage of the optimization problem.

X. Conclusion and Future Work

The traditional metrics of information leakage implicitly assume that the stochastic mechanism correlating the secret with the disclosed variable is known to the adversary. This assumption does not hold up in practice as most platforms do not publicly reveal their mechanisms for privatizing sensitive data. Therefore, the adversary can only approximate the true information disclosure mechanism. The conventional information leakage measures fail to compute the information leakage in these situations correctly. This paper introduces various information leakage measures to correctly compute the leakage when an adversary lacks complete statistical information about the true joint distribution of private, utility, and disclosed variables. These measures capture the various facets of the information leakage that result from the imperfect knowledge of the distribution. Furthermore, we have also considered distinct adversary characteristics and formulated optimization problems for each of these diverse adversaries. The solution to these optimization problems results in an optimized information disclosure mechanism that will minimize the worst-case maximization of any of the proposed metrics. Finally, we have simulated a case study where we observed that both the average subjective leakage and average confidence boost metric decrease monotonically with an increasing number of samples, whereas the average objective leakage increases gradually. These metrics converge to min-entropy leakage when the adversary is given access to all the samples. Furthermore, we have also solved the formulated optimization problems to achieve the optimized worst-case leakage values of our proposed metrics, and shown that such optimization, makes significant differences to the the worst-case leakages.

Acknowledgments

This work was partially supported by NIST CSAFE under Cooperative Agreements No. 70NANB15H176 and 70NANB20H019, NSF under grants No. CNS-1527579, CNS-1619201, CNS-1730275, DEB-1924178, ECCS-2030249, and Boeing Company. This publication was made possible by NPRP grant #12C-33905-SP-165 from the Qatar National Research Fund (a member of Qatar Foundation). The findings achieved herein are solely the responsibility of the authors. We appreciate anonymous reviewers for their valuable suggestions and comments.

References

[1] Statista, "Favorite music genres among consumers in the United States as of July 2018, by age group," https://www.statista.com/statistics/253915/favorite-music-genres-in-the-us/, 2018, [Online; accessed 23-August-2021].

[2] K. Zhang and X. Wang, "Peeping tom in the neighborhood: Keystroke eavesdropping on multi-user systems." in *USENIX Security Symposium*, vol. 20, 2009, p. 23.

[3] D. Gunduz, E. Erkip, and H. V. Poor, "Lossless compression with security constraints," in *2008 IEEE International Symposium on Information Theory*. IEEE, 2008, pp. 111–115.

[4] D. Rebollo-Monedero, J. Forne, and J. Domingo-Ferrer, "From t-closeness-like privacy to postrandomization via information theory," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 11, pp. 1623–1636, 2009.

[5] L. Sankar, S. R. Rajagopalan, and H. V. Poor, "Utility-privacy tradeoffs in databases: An information-theoretic approach," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 838–852, 2013.

[6] S. Li, A. Khisti, and A. Mahajan, "Information-theoretic privacy for smart metering systems with a rechargeable battery," *IEEE Transactions on Information Theory*, vol. 64, no. 5, pp. 3679–3695, 2018.

[7] I. Issa, A. B. Wagner, and S. Kamath, "An operational approach to information leakage," *IEEE Transactions on Information Theory*, vol. 66, no. 3, pp. 1625–1657, 2019.

[8] G. Smith, "On the foundations of quantitative information flow," in *International Conference on Foundations of Software Science and Computational Structures*. Springer, 2009, pp. 288–302.

[9] C. Braun, K. Chatzikokolakis, and C. Palamidessi, "Quantitative notions of leakage for one-try attacks," 2009.

[10] M. S. Alvim, M. E. Andrés, K. Chatzikokolakis, P. Degano, and C. Palamidessi, "Differential privacy: on the trade-off between utility and information leakage," in *International Workshop on Formal Aspects in Security and Trust*. Springer, 2011, pp. 39–54.

[11] H. Wang, M. Diaz, F. P. Calmon, and L. Sankar, "The utility cost of robust privacy guarantees," in *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2018, pp. 706–710.

[12] H. Wang, L. Vo, F. P. Calmon, M. Médard, K. R. Duffy, and M. Varia, "Privacy with estimation guarantees," *IEEE Transactions on Information Theory*, vol. 65, no. 12, pp. 8025–8042, 2019.

[13] K. Chatzikokolakis, T. Chothia, and A. Guha, "Statistical measurement of information leakage," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2010, pp. 390–404.

[14] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE mobile computing and communications review*, vol. 5, no. 1, pp. 3–55, 2001.

[15] A. Rényi, "On measures of entropy and information," in *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. University of California Press, 1961, pp. 547–561.

[16] T. M. Cover and J. A. Thomas, "Information theory and statistics," *Elements of Information Theory*, vol. 1, no. 1, pp. 279–335, 1991.

[17] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[18] Y. Deng, J. Pang, and P. Wu, "Measuring anonymity with relative entropy," in *International Workshop on Formal Aspects in Security and Trust*. Springer, 2006, pp. 65–79.

[19] A. R. Coble, "Formalized information-theoretic proofs of privacy using the hol4 theorem-prover," in *International Symposium on Privacy Enhancing Technologies Symposium*. Springer, 2008, pp. 77–98.

[20] F. du Pin Calmon and N. Fawaz, "Privacy against statistical inference," in *2012 50th annual Allerton conference on communication, control, and computing (Allerton)*. IEEE, 2012, pp. 1401–1408.

[21] A. Ly, M. Marsman, J. Verhagen, R. P. Grasman, and E.-J. Wagenmakers, "A tutorial on fisher information," *Journal of Mathematical Psychology*, vol. 80, pp. 40–55, 2017.

[22] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[23] D. Kifer and A. Machanavajjhala, "No free lunch in data privacy," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, 2011, pp. 193–204.

[24] A. Blum, K. Ligett, and A. Roth, "A learning theory approach to noninteractive database privacy," *Journal of the ACM (JACM)*, vol. 60, no. 2, pp. 1–25, 2013.

[25] E. Shi, T. H. Chan, E. Rieffel, R. Chow, and D. Song, "Privacy-preserving aggregation of time-series data," in *Proc. NDSS*, vol. 2. Citeseer, 2011, pp. 1–17.

[26] M. Jelasity and K. P. Birman, "Distributional differential privacy for large-scale smart metering," in *Proceedings of the 2nd ACM workshop on Information hiding and multimedia security*, 2014, pp. 141–146.

[27] K. Chatzikokolakis, M. E. Andrés, N. E. Bordenabe, and C. Palamidessi, "Broadening the scope of differential privacy using metrics," in *International Symposium on Privacy Enhancing Technologies Symposium*. Springer, 2013, pp. 82–102.

[28] Netflix, "How Netflix's Recommendations System Works," https://help.netflix.com/en/node/100639, [Online; accessed 28-April-2022].

[29] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *2008 IEEE Symposium on Security and Privacy (sp 2008)*. IEEE, 2008, pp. 111–125.

[30] K. Honda, A. Notsu, and H. Ichihashi, "Collaborative filtering by sequential extraction of user-item clusters based on structural balancing approach," in *2009 IEEE International Conference on Fuzzy Systems*. IEEE, 2009, pp. 1540–1545.

[31] S.-M. Choi and Y.-S. Han, "A content recommendation system based on category correlations," in *2010 Fifth International Multi-conference on Computing in the Global Information Technology*. IEEE, 2010, pp. 66–70.

[32] S. K. Sakib, G. T. Amariucai, and Y. Guan, "Information leakage metrics for adversaries with incomplete information: Binary privacy mechanism," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–7.

[33] S. A. M'rio, K. Chatzikokolakis, C. Palamidessi, and G. Smith, "Measuring information leakage using generalized gain functions," in *2012 IEEE 25th Computer Security Foundations Symposium*. IEEE, 2012, pp. 265–279.

[34] T. M. Cover and J. A. Thomas, *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, July 2006.

[35] I. Csiszár and J. Körner, *Information theory: coding theorems for discrete memoryless systems*. Cambridge University Press, 2011.

[36] P. Harsha, "12. Hellinger distance," https://www.tifr.res.in/~prahladh/teaching/2011-12/comm/lectures/l12.pdf, 2011, [Online; accessed 12-August-2022].

[37] T. Nishiyama, "A tight lower bound for the hellinger distance with given means and variances," *arXiv preprint arXiv:2010.13548*, 2020.

[38] U. M. Learning, "Iris species," https://www.kaggle.com/uciml/iris, 2016-09-27, accessed: 2022-03-21.

[39] C. Sharma, B. Mandal, and G. Amariucai, "A practical approach to navigating the tradeoff between privacy and precise utility," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[40] M. R. Clarkson, A. C. Myers, and F. B. Schneider, "Quantifying information flow with beliefs," *Journal of Computer Security*, vol. 17, no. 5, pp. 655–701, 2009.

[41] S. Hamadou, C. Palamidessi, and V. Sassone, "Quantifying leakage in the presence of unreliable sources of information," *Journal of Computer and System Sciences*, vol. 88, pp. 27–52, 2017.

[42] M. S. Alvim, K. Chatzikokolakis, A. McIver, C. Morgan, C. Palamidessi, and G. Smith, "Axioms for information leakage," in *2016 IEEE 29th Computer Security Foundations Symposium (CSF)*. IEEE, 2016, pp. 77–92.

[43] G. Cherubin, K. Chatzikokolakis, and C. Palamidessi, "F-bleau: fast black-box leakage estimation," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 835–852.

[44] M. Romanelli, K. Chatzikokolakis, C. Palamidessi, and P. Piantanida, "Estimating g-leakage via machine learning," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 697–716.

[45] I. Issa, S. Kamath, and A. B. Wagner, "An operational measure of information leakage," in *2016 Annual Conference on Information Science and Systems (CISS)*. IEEE, 2016, pp. 234–239.

[46] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 3–18.

[47] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*. IEEE, 2018, pp. 268–282.

[48] F. Farokhi and M. A. Kaafar, "Modelling and quantifying membership information leakage in machine learning," *arXiv preprint arXiv:2001.10648*, 2020.

[49] A. Hannun, C. Guo, and L. van der Maaten, "Measuring data leakage in machine-learning models with fisher information," *arXiv preprint arXiv:2102.11673*, 2021.

[50] F. Farokhi and H. Sandberg, "Fisher information privacy with application to smart meter privacy using hvac units," in *Privacy in Dynamical Systems*. Springer, 2020, pp. 3–17.

[51] ——, "Ensuring privacy with constrained additive noise by minimizing fisher information," *Automatica*, vol. 99, pp. 275–288, 2019.

[52] M. S. Alvim, K. Chatzikokolakis, Y. Kawamoto, and C. Palamidessi, "Information leakage games," in *International Conference on Decision and Game Theory for Security*. Springer, 2017, pp. 437–457.

[53] ——, "Information leakage games: Exploring information as a utility function," *arXiv preprint arXiv:2012.12060*, 2020.

[54] N. Alon, Y. Emek, M. Feldman, and M. Tennenholtz, "Adversarial leakage in games," *SIAM Journal on Discrete Mathematics*, vol. 27, no. 1, pp. 363–385, 2013.

[55] H. Brenner and K. Nissim, "Impossibility of differentially private universally optimal mechanisms," in *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*. IEEE, 2010, pp. 71–80.

[56] A. Ghosh, T. Roughgarden, and M. Sundararajan, "Universally utility-maximizing privacy mechanisms," *SIAM Journal on Computing*, vol. 41, no. 6, pp. 1673–1693, 2012.

[57] M. Gupte and M. Sundararajan, "Universally optimal privacy mechanisms for minimax agents," in *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2010, pp. 135–146.

[58] C. Li, M. Hay, V. Rastogi, G. Miklau, and A. McGregor, "Optimizing linear counting queries under differential privacy," in *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2010, pp. 123–134.

[59] K. Kalantari, L. Sankar, and A. D. Sarwate, "Robust privacy-utility tradeoffs under differential privacy and hamming distortion," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2816–2830, 2018.

[60] T. Xiao and A. Khisti, "Maximal information leakage based privacy preserving data disclosure mechanisms," in *2019 16th Canadian Workshop on Information Theory (CWIT)*. IEEE, 2019, pp. 1–6.

[61] D. Desfontaines and B. Pejó, "Sok: differential privacies," *Proceedings on Privacy Enhancing Technologies*, vol. 2020, no. 2, 2020.

[62] I. Wagner and D. Eckhoff, "Technical privacy metrics: a systematic survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 3, pp. 1–38, 2018.