MCMC for Bayesian Estimation of Differential Privacy from Membership Inference Attacks

Ceren Yıldırım[1][0000−0002−8029−6520](✉), Kamer Kaya[1,2][0000−0001−8678−5467], Sinan Yıldırım[1,2][0000−0001−7980−8990], and Erkay Savaş[1][0000−0002−4869−5556]

[1] Sabancı University, Faculty of Natural Sciences and Engineering, İstanbul 34956, Turkey {cerenyildirim, kaya, sinanyildirim, erkays}@sabanciuniv.edu
[2] Sabancı University, VERIM, 34956, İstanbul, Turkey

Abstract. We propose a new framework for Bayesian estimation of differential privacy, incorporating evidence from multiple membership inference attacks (MIA). Bayesian estimation is carried out via a Markov Chain Monte Carlo (MCMC) algorithm, named `MCMC-DP-Est`, which provides an estimate of the full posterior distribution of the privacy parameter (e.g., instead of just credible intervals). Critically, the proposed method does *not* assume that privacy auditing is performed with the most powerful attack on the worst-case (dataset, challenge point) pair, which is typically unrealistic. Instead, `MCMC-DP-Est` jointly estimates the strengths of MIAs used *and* the privacy of the training algorithm, yielding a more cautious privacy analysis. We also present an economical way to generate measurements for the performance of an MIA that is to be used by the MCMC method to estimate privacy. We present the use of the methods with numerical examples with both artificial and real data.

Keywords: Differential Privacy · Membership Inference Attacks · Bayesian estimation · Markov Chain Monte Carlo

1 Introduction

Differential privacy (DP) has emerged as a gold standard for quantifying and guaranteeing privacy in data analysis and machine learning [10,12]. DP provides a mathematical framework to limit the impact of an individual's data on the output of a random algorithm, enabling robust privacy guarantees regardless of the adversary's knowledge other than that individual. DP is defined below.

Definition 1 (DP). *An algorithm $\mathcal{A}$ with input space $\bigcup_{n=1}^{\infty} \mathcal{Z}^n$ and output space Ω is (ϵ, δ)-DP if for every $n \in \mathbb{N}$, $D \in \mathcal{Z}^n$, $z \in \mathcal{Z}$, and $E \subseteq \Omega$, we have*

$$P(\mathcal{A}(D) \in E) \leq e^{\epsilon} P(\mathcal{A}(D \cup \{z\}) \in E) + \delta,$$
$$P(\mathcal{A}(D \cup \{z\}) \in E) \leq e^{\epsilon} P(\mathcal{A}(D) \in E) + \delta.$$

Quantifying the privacy of practical implementations remains a challenging task. Theoretical lower bounds for (ϵ, δ) have been extensively analyzed for a variety of mechanisms, such as noise-adding mechanisms (Laplace, Gaussian, etc.) [11,12], subsampling [4], and their composition [15]. Other theoretical definitions of DP have also been used to derive lower bounds of DP [5,9,21]. However, theoretical lower bounds are not tight for many practical algorithms whose revealed outputs result from a series of calculations involving randomness. An example is when a training algorithm outputs *only* its final model, possibly following randomizing steps like random initialization, random updates (e.g., due to subsampling and/or noisy gradients), or output perturbation. In such a case, a large gap between the theoretical bounds and the actual privacy is shown to exist [24]. The privacy-auditing, or privacy estimation, of such complex but practical algorithms, through numerical estimation of (ϵ, δ), has become an emerging research line [2,13,14,18,19,20,22,23,24,25,28,31]. This study follows this line by proposing a new framework for Bayesian privacy estimation.

Privacy auditing methods leverage the relation between DP and *Membership Inference Attacks* (MIA) [27,30,6,29] to estimate ϵ and δ. This is because DP particularly guarantees the protection of sensitive input data against MIAs. While there are different types of MIAs, by Definition 1, the leave-one-out attack (L-attack in [29]) is the one directly relevant to the DP of a private algorithm. In an L-attack, a data set D, a data point $z \in \mathcal{Z}$, and a random output of $\mathcal{A}$ are given to the attacker who uses the given information to infer whether D or $D \cup \{z\}$ was used by $\mathcal{A}$. For *any* D, z and *any* statistical test, the type I and type II error probabilities are lower-bounded by a curve determined by the (ϵ, δ) of $\mathcal{A}$, see Theorem 1.

Two main issues deserve caution in empirical privacy analysis based on MIAs.

1. *The attack strategy to audit privacy is typically not the strongest*: For a given pair (D, z), the strongest attack that decides based on $\mathcal{A}$'s output is known to be the likelihood ratio test (LRT). However, practical MIAs to audit privacy merely approximate the LRT [26] with a limited computational budget, e.g. using metrics that are loss-based [30,29,6], gradient-based [22], etc. On the other hand, an adversary can design more powerful attacks than the one used for auditing, provided computational budget. It is also difficult to analytically characterize the gap between the performances of a given MIA and the LRT. Therefore, treating a given MIA as *the* strongest attack may lead to overconfident estimates about the privacy of an algorithm ('overconfident', because weaker attacks imply stronger privacy; see Remark 2). This is particularly dangerous since, based on overconfident estimations, private data may be leaked to a greater extent than it is permitted.
2. *The challenge base* (D, z) *may not be the absolute "worst-case" challenge base.* Moreover, existing efforts such as [23,19,31,25] do not *theoretically* guarantee to find such a point. Therefore, *even if* the strongest attack is applied on the challenge base (D, z), its observed false positive and false negative counts should *not* directly be used to upper bound ϵ. This is because tests on another (D, z) challenge base could result in a larger upper bound on ϵ.

Considering the above issues, we propose a new Bayesian methodology for empirical privacy estimation. Our contributions are as follows.

1. **A new posterior sampling method for DP estimation:** We develop a joint probability distribution for the privacy parameter, attack strengths, and false negatives/positives involved in *block-box* auditing a private algorithm, where one has access only to the output of $\mathcal{A}$ and not its intermediate results. Suited to this model, we propose a Markov Chain Monte Carlo (MCMC) method named `MCMC-DP-Est` (Alg. 1), adopted from [3], for Bayesian estimation of privacy. The advantages of `MCMC-DP-Est` are as follows:
   - **Full posterior distribution:** Beyond credible intervals, such as in [31], it returns the *full posterior distribution* of the privacy parameters.
   - **Combining multiple results:** With the fully Bayesian treatment, the algorithm is able to combine the false negative and false positive counts from *multiple* (D, z) points (and possibly from multiple attack strategies). In particular, *no tried attacks need to be thrown away.* This also enables leveraging new (D, z) points or attack strategies to refine the privacy estimates coherently.
   - **Cautious treatment of attack strengths:** Related to the above discussion, the probabilistic model based on which MCMC is used in this work does *not* assume that the attack used is the strongest attack possible under the privacy constraint or it is performed on the worst-case (D, z) pair (though it can be adapted to include such cases). Instead, we parametrize the average strength of the applied tests/challenge bases by a parameter $s \in [0, 1]$, and use MCMC to jointly estimate both s and ϵ.
2. **A method for measuring MIA performance:** We present a parametric loss-based MIA, adopting LiRA [6], to feed privacy auditing methods (including ours) with informative error counts. We propose a computationally efficient way to measure the MIA performance. The measurements, the numbers of false positives and false negatives, are to be fed to the MCMC algorithm as observations.
3. **An extension of the joint probability model** that allows statistical dependencies among attacks is also discussed briefly in Remark 3 in Section 2.1 and more in detail in Section 3 of a separate **Supplementary File**. Attack results can be statistically dependent, for example, when a common z point is paired with different D datasets from the population.

Section 2 presents the joint probability model and `MCMC-DP-Est`. Section 3, presents an MIA and an experiment design to collect performance measures for the attack, which are to be fed to the MCMC algorithm as observations. Section 4 presents the experiments. Section 5 concludes the paper.

1.1 Related work

Privacy estimation methods exploit theoretical results and approach their guarantees empirically. For example, the privacy estimation in [13] estimates the privacy of stochastic gradient descent (SGD) based on the relationship between the sensitivity of the output and privacy. However, in most studies, the relation between MIAs and Definition 1 of DP has been exploited. For example, [14] derives Clopper-Pearson confidence intervals for ϵ from MIAs carefully designed for SGD with clipping. [23] uses Clopper-Pearson confidence intervals, too, but additionally finds the worst-case pairs (D, z) to improve the bound on the privacy estimates. In contrast to frequentist estimates in [14,23], Bayesian privacy estimation is proposed in [31], where Bayesian *credible intervals* are provided for ϵ. Privacy estimation has also been extended to other definitions of privacy [18,22,25] and to federated learning [2,20].

The quality of privacy estimation through MIAs depends heavily on the quality of MIAs, i.e., their power to distinguish membership and non-membership. Several MIAs (tests) have been proposed in the literature. If black-box privacy-auditing is performed, the loss function of the trained model is typically involved in the MIA decision rule and using the loss function is shown to be Bayes optimal in [26]. The LOSS attack [30] uses the loss function as its test statistic. This attack is also used as an approximation of the LRT under some conditions [29] that correspond to the output of the training model behaving like a sample from its posterior distribution. The LOSS attack has been reported to be weak in identifying memberships and strong in identifying non-memberships [6]. As an alternative, [6] proposes LiRA, a more direct approximation of the LRT that considers the distribution of the loss function under both hypotheses.

2 Bayesian estimation of privacy

We present the joint probability distribution for the variables regarding the privacy of $\mathcal{A}$, error probabilities of MIAs, and the observed false positive (FP) and false negative (FN) counts for each MIA. Then, we present the MCMC algorithm for the privacy estimation of $\mathcal{A}$ according to that joint probability distribution. First, we provide some preliminaries and introduce some concepts for clarity. Definition 2 assigns a specific meaning to the term 'challenge base' for the clarity of presentation.

Definition 2 (Challenge base). *A challenge base is a pair (D, z), where $D \in \bigcup_{n=1}^{n} \mathcal{Z}^n$ and $z \in \mathcal{Z}$ with $z \notin D$.*

Next, we define an MIA as a statistical test specified by a challenge base, a critical region, and a private algorithm whose output serves as the observation point for that test.

Definition 3 (MIA). *An MIA is a statistical test specified by $(D, z, \mathcal{A}, \phi, \alpha, \beta)$, where $\phi : \Omega \mapsto \{0, 1\}$ is a (possibly random) decision rule for the absence or presence of z in the input of $\mathcal{A}$ based on a random outcome θ from $\mathcal{A}$, and α, β are the deduced type I and type II error probabilities given by*

$$\alpha = P(\phi(\theta) = 1 | \theta \sim \mathcal{A}(D)), \quad \beta = P(\phi(\theta) = 0 | \theta \sim \mathcal{A}(D \cup \{z\})).$$

The theorem below from [16, Theorem 2.1] is central to the methodology presented in this paper. It sets an upper bound on the accuracy of MIAs whose sample is the output of an (ϵ, δ)-DP algorithm.

Theorem 1. *$\mathcal{A}$ is (ϵ, δ)-DP if and only if, for any $D \in \mathcal{Z}^n$ and $z \in \mathcal{Z}$, and a decision rule ϕ, the MIA $(D, z, \phi, \mathcal{A}, \alpha, \beta)$ satisfies $(\alpha, \beta) \in \mathcal{R}(\epsilon, \delta)$, where*

$$\mathcal{R}(\epsilon, \delta) := \left\{ (x, y) \in [0,1]^2 : \begin{array}{c} x + e^{\epsilon} y \geq 1 - \delta,\ y + e^{\epsilon} x \geq 1 - \delta, \\ y + e^{\epsilon} x \leq e^{\epsilon} + \delta,\ x + e^{\epsilon} y \leq e^{\epsilon} + \delta \end{array} \right\}.$$

See Figure 1 (top left) for an illustration of $\mathcal{R}(\epsilon, \delta)$.

Remark 1. In this work, we will assume $\delta \geq 0$ fixed and known, and we will focus on estimating the parameter ϵ. However, if an algorithm is (ϵ_1, δ) it is also (ϵ_2, δ) for any $\epsilon_2 > \epsilon_1$. To prevent ambiguity, by "estimating privacy", we specifically mean estimating $\epsilon := \inf\{\epsilon_0 \geq 0 : \mathcal{A} \text{ is } (\epsilon_0, \delta)\text{-DP}\}$.

2.1 Joint probabilistic model for privacy- and MIA-related variables

We present in detail the joint probability model illustrated in Figure 1 (bottom).

Observed error counts: We assume that there are $n \geq 1$ challenge bases (D_i, z_i), $i = 1, \ldots, n$. On each challenge base, an MIA $(D_i, z_i, \phi_i, \mathcal{A}, \alpha_i, \beta_i)$ is challenged $N_{i,0}, N_{i,1}$ times under H_0, H_1, respectively. More explicitly, for $j = 1, \ldots, N_{i,0}$, we challenge an MIA $(D_i, z_i, \phi_i, \mathcal{A}, \alpha_i, \beta_i)$ with $\theta_0^{(j)} \sim \mathcal{A}(D)$. We collect X_i, the number of false positives out of the $N_{i,0}$ challenges. Likewise, we challenge the same MIA $N_{i,1}$ times with $\theta_1^{(j)} \sim \mathcal{A}(D \cup \{z\})$. We collect Y_i, the number of false negatives out of the $N_{i,1}$ challenges.

We denote the conditional distribution of X_i, Y_i given α_i, β_i by $g(X_i, Y_i | \alpha_i, \beta_i)$. When the tests for each challenge base are independent, X_i and Y_i become independent binomials and their conditional distributions become

$$g(X_i, Y_i | \alpha_i, \beta_i) = \mathrm{Binom}(X_i | N_{i,0}, \alpha_i) \times \mathrm{Binom}(Y_i | N_{i,1}, \beta_i), \tag{1}$$

where α_i, β_i are the error probabilities of the i'th MIA. Other distributions may arise with dependent tests, e.g. because of using common shadow models to learn the null and alternative hypotheses. We discuss such a case in Section 3.2.

True error probabilities: The performance of an MIA depends on $\mathcal{A}$, the challenge base (D, z) as well as the decision rule ϕ. When we have little knowledge about the performance of a test, a convenient choice for its conditional prior distribution for (α_i, β_i) given (ϵ, δ) is the uniform distribution over $\mathcal{R}(\epsilon, \delta)$. However, uniformity over $R(\epsilon, \delta)$ may be a loose assumption for carefully designed attacks. Indeed, several works in the literature study the design of powerful MIAs by approximating the LRT [6,23,29,30] or finding the worst-case (or "best-case" from the attacker's point of view) tuple (D, z), or both. When such techniques are involved, the prior of $\alpha_{1:n}, \beta_{1:n}$ given ϵ, δ can be modified as

$$(\alpha_1, \beta_1), \ldots, (\alpha_n, \beta_n) | \epsilon, \delta \overset{\text{iid}}{\sim} \mathrm{Unif}(\mathcal{R}_s(\epsilon, \delta)), \quad \mathcal{R}_s(\epsilon, \delta) := \mathcal{R}(\epsilon, \delta) \backslash \mathcal{R}(s\epsilon, s\delta). \tag{2}$$

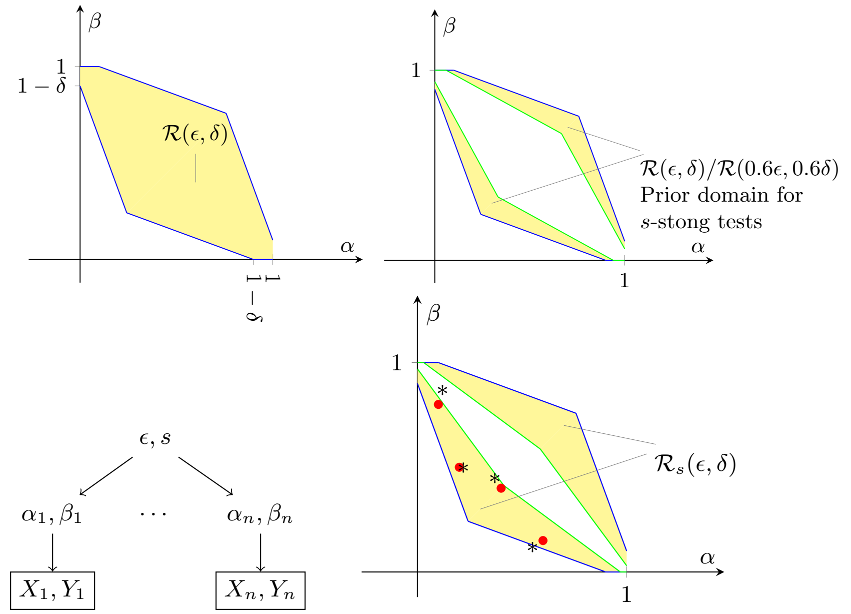

Fig. 1: **Top Left**: $\mathcal{R}(\alpha,\beta)$, the unconstrained prior domain ($s=0$) for α,β of an MIA. **Top Right**: $\mathcal{R}_{0.6}(\alpha,\beta)$, prior domain for $s=0.6$. **Bottom Left**: The dependency structure of the variables involved (a fixed δ is assumed). **Bottom Right:** Realization of the variables. ϵ and s set the blue and green lines, respectively; (α_i,β_i) and $(X_i/N_{i,0}, Y_i/N_{i,1})$ are shown with red and black points, respectively.

Here, $s \in (0,1)$ is a strength parameter for the test; the closer it is to 1, the stronger the test is expected. The parameter s itself can be modeled as a random variable, for example, as $s \sim \text{Beta}(a,b)$, and it can also be estimated from the results of multiple MIAs. The pdf of (α_i,β_i) in the modified case is

$$p_s(\alpha_i,\beta_i|\epsilon,\delta) = \mathbb{I}((\alpha,\beta) \in \mathcal{R}_s(\epsilon,\delta))/|\mathcal{R}_s(\epsilon,\delta)|, \qquad (3)$$

where $|\mathcal{R}_s(\epsilon,\delta)|$ is the area of $\mathcal{R}_s(\epsilon,\delta)$, given by

$$|\mathcal{R}_s(\epsilon,\delta)| = 2[(1-s\delta)^2 e^{-s\epsilon}/(1+e^{-s\epsilon}) - (1-\delta)^2 e^{-\epsilon}/(1+e^{-\epsilon})].$$

Figure 1 illustrates this prior for $s=0$ (top left) and $s=0.6$ (top right).

Remark 2 (The special case $s=1$). The choice $s=1$ corresponds to the assumption that the strongest possible attacks are used to generate (X_i,Y_i) pairs. Several studies make this assumption implicitly, relying on the quality of their MIAs [14,23,31]. In many cases, the assumption is too strong since determining the worst-case challenge base (D,z) *and* using the most powerful test for that couple is usually intractable. As a result, taking $s=1$ may lead to overconfident estimations about ϵ. (See Figure 2.) More concretely, the relation between ϵ and (α, β) can be written as

$$\{\epsilon \leq \epsilon_0\} \Leftrightarrow \{(\alpha, \beta) \in \mathcal{R}(\epsilon_0, \delta) \textit{ for any } \mathrm{MIA}(D, z, \mathcal{A}, \phi, \alpha, \beta)\}.$$

When a *particular* MIA is concerned, the above gives a one-way implication as

$$\textit{for any } \mathrm{MIA}\ (D, z, \mathcal{A}, \phi, \alpha, \beta), \quad \{\epsilon \leq \epsilon_0\} \Rightarrow \{(\alpha, \beta) \in \mathcal{R}(\epsilon_0, \delta)\},$$

which leads to

$$P(\epsilon \leq \epsilon_0) \leq P[(\alpha, \beta) \in \mathcal{R}(\epsilon_0, \delta)]. \tag{4}$$

Replacing the inequality in (4) with equality is equivalent to taking $s = 1$, which would be valid only when the LRT is applied exactly *and* on the worst-case challenge base (D, z), which is typically not guaranteed in practice. In the absence of that strong condition, $s = 1$ leads to early saturation of the cdf $P(\epsilon \leq \epsilon_0)$ vs ϵ_0 and results in overconfident (credible) intervals for ϵ_0. Section 4.1 numerically demonstrates the effect of s in privacy estimation.

Remark 3 (Dependent challenge bases and MIAs). In (2), the MIA performances are assumed conditionally independent given ϵ, δ. Statistical dependency can exist among the MIAs depending on how they are designed. Dependency can occur, for example, when a group of MIAs have distinct Ds but the *same* z point, or they have a common challenge base (D, z) but differ in their decision rules. Dependent MIAs can also be incorporated into the statistical model. Section 3 of the Supplementary File contains a modeling approach for dependent MIAs.

Priors for privacy and attack strengths: We assume a fixed δ and estimate ϵ (and s). For the priors of ϵ and s, we consider a one-sided normal distribution $\epsilon \sim \mathcal{N}_{[0,\infty)}(0, \sigma_\epsilon^2)$ and $s \sim \mathrm{Beta}(a, b)$ independently, with $\sigma_\epsilon^2 > 0$ and $a, b > 0$.

Joint probability distribution: Finally, the overall joint probability distribution of ϵ, s, $\alpha_{1:n}$, $\beta_{1:n}$, $X_{1:n}$, $Y_{1:n}$ can be written as

$$p_\delta(\epsilon, s, \alpha_{1:n}, \beta_{1:n}, X_{1:n}, Y_{1:n}) = p(\epsilon)p(s) \prod_{i=1}^{n} p_\delta(\alpha_i, \beta_i | \epsilon, s) g(X_i, Y_i | \alpha_i, \beta_i). \tag{5}$$

Figure 1 shows the hierarchical structure according to (5) (bottom left) and an example realization of the variables in the model (bottom right).

2.2 Estimating privacy via MCMC

Alg. 1 shows the MCMC method for the *joint posterior distribution*

$$p_\delta(\epsilon, s, \alpha_{1:n}, \beta_{1:n} | X_{1:n}, Y_{1:n}) \propto p_\delta(\epsilon, s, \alpha_{1:n}, \beta_{1:n}, X_{1:n}, Y_{1:n}). \tag{6}$$

We call this method `MCMC-DP-Est`. Although `MCMC-DP-Est` iterates for $(\epsilon, s, \alpha_{1:n}, \beta_{1:n})$, the ϵ (or (ϵ, s))-component of the samples can be used to estimate the marginal

Algorithm 1: `MCMC-DP-Est`: posterior sampling for (ϵ, s)

```
Input: $M$: Number of MCMC iterations,
$(\epsilon^{(0)}, s^{(0)})$: Initial values;
$X_{1:n}, N_{0,1:n}, Y_{1:n}, N_{1,1:n}$: FP and FN counts for each challenge base and numbers of challenges under $H_0, H_1$ for each challenge base;
$\sigma^2_{q,\epsilon}\sigma^2_{s,\epsilon}$: Proposal variances for $\log \epsilon$ and $s$;
$K$: Number of auxiliary variables in one iteration of MCMC.
Output: Samples $\epsilon^{(i)}, s^{(i)}$, $i = 1, \dots, M$ from $p_\delta(\epsilon, s, \alpha_{1:n}, \beta_{1:n} | X_{1:n}, Y_{1:n})$
for $i = 1 : M$ do
    Draw the proposal $\epsilon' \sim \log \mathcal{N}(\log \epsilon, \sigma^2_{q,\epsilon})$ and $s' \sim \mathcal{N}(s, \sigma^2_{q,s})$.
    for $j = 1 : n$ do
        Set $(\alpha_j^{(1)}, \beta_j^{(1)}) = (\alpha_j, \beta_j)$.
        Sample $\alpha_j^{(k)}, \beta_j^{(k)} \overset{\text{iid}}{\sim} \text{Unif}(0, 1)$ for $k = 2, \dots, K$.
        Calculate the weights
            $w_j^{(k)} = p_\delta(\alpha_j^{(k)}, \beta_j^{(k)} | \epsilon, s) g(X_j, Y_j | \alpha_j^{(k)}, \beta_j^{(k)}), \quad k = 1, \dots, K$
            $w_j^{\prime(k)} = p_\delta(\alpha_j^{(k)}, \beta_j^{(k)} | \epsilon', s') g(X_j, Y_j | \alpha_j^{(k)}, \beta_j^{(k)}) \quad k = 1, \dots, K$
    Acceptance probability:
        $A = \min \left\{ 1, \frac{p(s')p(\epsilon')\epsilon'}{p(s)p(\epsilon)\epsilon} \prod_{j=1}^{n} \frac{\sum_{k=1}^{K} w_j^{\prime(k)}}{\sum_{k=1}^{K} w_j^{(k)}} \right\}.$
    Accept/Reject: Draw $u \sim \text{Unif}(0, 1)$.
    if $u \leq A$ then
        Set $\epsilon = \epsilon', s = s'$, and $\bar{w}_{1:n}^{(1:K)} = w_{1:n}^{\prime(1:K)}$.
    else
        Keep $\epsilon, s$ and set $\bar{w}_{1:n}^{(1:K)} = w_{1:n}^{(1:K)}$.
    for $j = 1, \dots, n$ do
        Sample $k \in \{1, \dots, K\}$ w.p. $\propto \bar{w}_j^{(k)}$ and set $(\alpha_j, \beta_j) = (\alpha_j^{(k)}, \beta_j^{(k)})$.
    Store $\epsilon^{(i)} = \epsilon, s^{(i)} = s$.
```

posterior of ϵ (or (ϵ, s)). `MCMC-DP-Est` is a variant of the MHAAR (Metropolis-Hastings with averaged acceptance ratios) methodology in [3] developed for latent variable models. (Here the latent variables are the $(\alpha_{1:n}, \beta_{1:n})$.) `MCMC-DP-Est` has $\mathcal{O}(Kn)$ complexity per iteration. We state the correctness of `MCMC-DP-Est` in the following proposition. A proof is given in Section 1 of the Supplementary File and contains a strong allusion to [3].

Proposition 1. *For any $K > 1$, $\sigma^2_{q,\epsilon}$, and $\sigma^2_{q,s}$, `MCMC-DP-Est` in Alg. 1 targets exactly the posterior distribution in* (6)*, in the sense that it simulates an ergodic Markov Chain whose invariant distribution is* (6)*.*

3 The MIA attack and measuring its performance

In this section, we describe the MIA used in our experiments and equip it with an experimental design to measure its performance computationally efficiently.

3.1 The MIA design

The test statistic of LRT, the most powerful test, is the ratio of likelihoods $p_{\mathcal{A}}(\theta|D)/p_{\mathcal{A}}(\theta|D \cup \{z\})$. However, the likelihoods are usually intractable due to $\mathcal{A}$'s complex structure; therefore, approximations are sought. Loss-based attacks are a common way of approximating the LRT [6,26,29,30]. In particular, we consider a parametric version LiRA [6], a loss-based attack that uses the ratio $p_L(\ell^*|H_0)/p_L(\ell^*|H_1)$ evaluated at $\ell^* = L(z, \theta)$. The outline of a loss-based MIA is given in Alg. 2. The densities $p_L(\cdot|H_i)$ can be approximated via $M_i > 1$ shadow models $\theta_i^{(1)}, \ldots, \theta_i^{(N)}$ generated under H_i and fitting a distribution $p_L(\cdot|H_i)$ to the losses $L(z, \theta_i^{(1)}), \ldots L(z, \theta_i^{(M_i)})$. Finally, the critical region to choose H_1 is set $\{p_L(\ell|H_0)/p_L(\ell|H_1) < \tau\}$ and τ is adjusted to have a desired target type I error probability α^*.

```
Algorithm 2: $b = \texttt{MIA}(\theta, D, z, \mathcal{A}, M_0, M_1, \alpha^*)$
for $j = 1, \ldots, M_0$ do
    Obtain $\theta_0^{(j)} \sim \mathcal{A}(D)$, calculate $\ell_0^{(j)} = L(z, \theta_0^{(j)})$
for $j = 1, \ldots, M_1$ do
    Obtain $\theta_1^{(j)} \sim \mathcal{A}(D \cup \{z\})$, calculate $\ell_1^{(j)} = L(z, \theta_1^{(j)})$.
return $b = \texttt{LearnAndDecide}(D, z, \theta, \alpha^*, \{\ell_0^{(j)}\}_{j=1}^{M_0}, \{\ell_1^{(j)}\}_{j=1}^{M_1})$
```

```
Algorithm 3: $\texttt{LearnAndDecide}(D, z, \theta, \alpha^*, \{\ell_0^{(j)}\}_{j=1}^{M_0}, \{\ell_1^{(j)}\}_{j=1}^{M_1})$
for $i = 0, 1$ do                                      // Learn $H_0$ and $H_1$
    Fit normal distributions for $H_i$ as
    $\mu_i = \frac{1}{M_i} \sum_{j=1}^{M_i} \ell_i^{(j)}, \quad \sigma_i^2 = \frac{1}{M_i - 1} \sum_{j=1}^{M_i} (\ell_i^{(j)} - \mu_i)^2$
Calculate $\ell^* = L(z, \theta)$.                     // Compute $\ell^*$, $R$ and the decision
Calculate $R = \frac{\mu_0/\sigma_0^2 - \mu_1/\sigma_1^2}{1/\sigma_0^2 - 1/\sigma_1^2}$ and $\delta = \frac{\mu_0}{\sigma_0} + \frac{1}{\sigma_0} R$
return Decision
    $b = \begin{cases} 1 & \text{if } (\ell^* + R)^2 \leq \sigma_0^2 F_{1,\delta^2}^{-1}(\alpha^*) \text{ and } \sigma_0^2 > \sigma_1^2, \\ 1 & \text{if } (\ell^* + R)^2 \geq \sigma_0^2 F_{1,\delta^2}^{-1}(1 - \alpha^*) \text{ and } \sigma_0^2 < \sigma_1^2, \\ 0 & \text{otherwise.} \end{cases}$
    where $F_{d,\delta^2}^{-1}(u)$ is the inverse cdf of $\chi_{d,\delta^2}^2$, the non-central $\chi^2$ dist. with
    noncentrality parameter $\delta^2$ and degrees of freedom $d$, evaluated at $u$.
```

Learning H_0 and H_1 and deciding: Alg. 3 describes how we learn the distributions under both hypotheses and apply a decision. Firstly, for each $i = 0, 1$ we fit a normal distribution $\mathcal{N}(\mu_i, \sigma_i^2)$ using the sample $\ell_i^{(1:M_i)}$, where $\ell_i^{(j)} = L(z, \theta_0^{(j)})$. Then, LRT is applied to decide between $H_0 : \ell \sim \mathcal{N}(\mu_0, \sigma_0^2)$ and $H_1 : \ell \sim \mathcal{N}(\mu_1, \sigma_1^2)$ with a target type I error probability of α^*.

3.2 Measuring the performance of the MIA

When the primary goal of using MIA is to audit the privacy of an algorithm, one needs to perform the attack multiple times to estimate its type I and type II error probabilities. A direct way to do this is Alg. 4, where the MIA is simply run N_0 and N_1 times, each with an independent output θ and independent sets of M_0, M_1 shadow models for H_0, H_1. The cost of this procedure is proportional to $(N_0 + N_1)(M_0 + M_1)$, which can be prohibitive.

```
Algorithm 4: MeasureMIA($\mathcal{A}, D, z, N_0, N_1, M_0, M_1$)
  Set $D_0 = D \backslash \{z\}$ and $D_1 = D \cup \{z\}$.
  for $i = 0, 1$ do
      for $j = 1, \dots, N_i$ do
          Train $D_i$ and output $\theta \sim \mathcal{A}(D_i)$.
          Decide according to $\hat{d}_i^{(j)} = \mathtt{MIA}(\theta, D, z, \mathcal{A}, M_0, M_1)$.
  return $X = \sum_{j=1}^{N_0} d_0^{(j)}$, $Y = \sum_{j=1}^{N_1} 1 - d_1^{(j)}$
```

We present a cheaper alternative to Alg. 4, in which N models are trained from H_0 and H_1 and *cross-feed* each other as shadow models. For each $i = 0, 1$ and $j = 1, \dots, N$, the triple $(D, z, \theta_i^{(j)})$ is taken as the input of MIA and the rest $\{\theta_i^{(1:N)-j}, \theta_{1-i}^{(1:N)}\}$ are used as the shadow models. This is presented in Alg. 5.

Although the decisions obtained with Alg. 4 are independent (given the true α, β of the MIA), those obtained with Alg. 5 are *not* independent; they are correlated due to using the same set of shadow models. As a result, the Binomial distributions (1) no longer hold for X_i, Y_i pairs obtained from Alg. 5. One can incorporate that into the joint distribution by taking the conditional distributions of X_i, Y_i given α, β as correlated Binomial distributions [17]. An alternative, which is pursued here, is to use a *bivariate normal approximation* for (X_i, Y_i) as

$$\mathcal{N}\left(\begin{bmatrix} N\alpha_i \\ N\beta_i \end{bmatrix}, \begin{bmatrix} \alpha_i(1-\alpha_i)(N + N(N-1)\tau) & N^2\rho\sqrt{\alpha_i(1-\alpha_i)\beta_i(1-\beta_i)} \\ N^2\rho\sqrt{\alpha_i(1-\alpha_i)\beta_i(1-\beta_i)} & \beta_i(1-\beta_i)(N + N(N-1)\tau) \end{bmatrix} \right). \tag{7}$$

The parameters τ, ρ can be estimated jointly with ϵ, s by slightly modifying `MCMC-DP-Est`. The details of this extension are given in Section 2 of the Supplementary File.

Algorithm 5: `MeasureMIAFast`$(\mathcal{A}, D, z, N, \alpha^*)$

```
Set $D_0 = D$ and $D_1 = D \cup \{z\}$.
for $i = 0, 1$ do                 // $N$ correlated attacks for $D_0$ vs $D_1$
    for $j = 1, \ldots, N$ do
        Obtain $\theta_i^{(j)} \sim \mathcal{A}(D_i)$ and calculate the loss $\ell_i^{(j)} = L(z, \theta_i^{(j)})$
for $j = 1, \ldots, N$ do                                     // Decisions
        $d_0^{(j)} = \texttt{LearnAndDecide}(D, z, \theta_0^{(j)}, \alpha^*, \{\ell_0^{(i)}\}_{i=1, i \neq j}^{N}, \{\ell_1^{(i)}\}_{i=1}^{N})$
        $d_1^{(j)} = \texttt{LearnAndDecide}(D, z, \theta_1^{(j)}, \alpha^*, \{\ell_0^{(i)}\}_{i=1}^{N}, \{\ell_1^{(i)}\}_{i=1, i \neq j}^{N})$
return $X = \sum_{j=1}^{N} d_0^{(j)}$, $Y = \sum_{j=1}^{N} 1 - d_1^{(j)}$
```

4 Experiments

The code to replicate all the experiments in the section can be downloaded at `https://github.com/cerenyildirim/MCMC_for_Bayesian_estimation`.

4.1 Privacy estimation with artificial test performance results

Role of s in privacy estimation: This experiment is designed to show the effect of the prior specification for the attack strength. For simplicity, we took $n = 1$ and focused on $N_{0,1} = N_{1,1} = N > 1$ instances of a single attack. Also, we set $X = 0.4 \times N$ and $Y = 0.4 \times N$ to imitate an attack with $\alpha = \beta = 0.4$. We ran `MCMC-DP-Est` in Alg. 1 with varying values of s that are seen on the x-axis of the left plot in Figure 2. The conditional distribution $g(X_i, Y_i | \alpha_i, \beta_i)$ is set to (1). The 90% credible interval (CI) for ϵ for each run (different s) by computing the 5%- and 95%- empirical quartiles obtained from the last 10^6 samples of the MCMC algorithm (discarding the first 10^5 samples). A dramatic change is visible in the CI width as a function of s. CIs as narrow as those reported in [31] with the same observations are obtained when $s \geq 0.9$. However, CIs are significantly wider for smaller (and arguably more realistic) values of s. Those results indicate the critical role of s, hence the importance of its estimation when it is unknown.

Estimating ϵ and s: Here we show how `MCMC-DP-Est` estimates ϵ and s jointly from multiple attack results in different scenarios. We took $n = 10$ and $N_{0,i} = N_{1,i} = 1000$ for all $i = 1, \ldots, n$. We considered two scenarios.

- In the first scenario, we made the test strengths evenly spread over $\mathcal{R}(\epsilon, \delta)$ by generating $\alpha_i, \beta_i \overset{\text{iid}}{\sim} \text{Beta}(10, 10)$, for $i = 1, \ldots, n$. The counts X_i, Y_i were drawn as $X_i \overset{\text{iid}}{\sim} \text{Binom}(N_{0,1}, \alpha_i)$, $Y_i \overset{\text{iid}}{\sim} \text{Binom}(N_{0,1}, \beta_i)$, independently.
- In the second, we assumed relatively accurate attacks as

$X_{1:10}$	40	50	60	100	100	110	120	200	200	200
$Y_{1:10}$	250	200	150	100	120	100	100	80	70	60

.

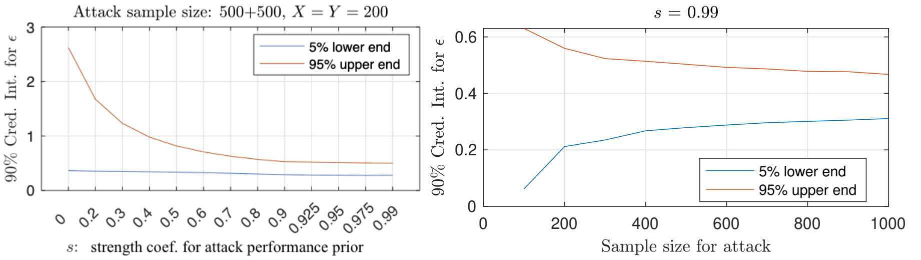

Fig. 2: **Left:** 90% CI for ϵ vs s. **Right:** 90% CI for ϵ vs N.

The observed error rates $(X_i/N_{0,i}, Y_i/N_{1,i})$ are shown on the left-most plot in Fig. 3. Alg. 1 was run to obtain 10^6 samples from $p(\epsilon, s|X_{1:n}, Y_{1:n})$. As previously, $g(X_i, Y_i|\alpha_i, \beta_i)$ is set to (1). The results in Fig. 3 indicate that our method can accurately estimate the attack strengths and ϵ together.

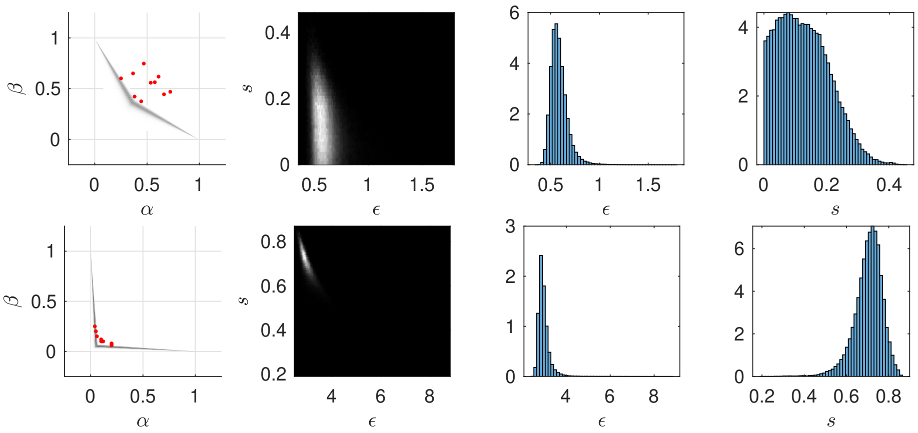

Fig. 3: Posterior distributions for ϵ, s from multiple attacks. **Top:** Weak attacks. **Bottom:** Strong attacks. The gray area in left-most plots are "histograms" of ϵ for the test according to the posterior distribution of ϵ (the symmetric counterpart is omitted).

4.2 Experiments with real data

We considered the MNIST dataset as the population, which contains 60,000 training examples [8]. Each example in the set contains a 28 × 28-pixel image and an associated categorical label in $\{0, \ldots, 9\}$. In the experiments, we construct D with a size of 999 (to avoid too small batches while training $D \cup \{z\}$). We generated $n = 20$ challenge bases and for each challenge base, we generated $N = 100$ challenges. Those challenges are also used as shadow models in a cross-feeding fashion, as described in Section 3.2.

Training algorithms and attacks: For $\mathcal{A}$, we considered a fully connected neural network with one hidden layer having 128 nodes and ReLU as its activation function. Meanwhile, the activation function of the output layer is softmax. We set the loss function $L(z, \theta)$ as categorical cross-entropy. To train the models, we use Keras and TensorFlow libraries [1,7]. For the optimizer, we use SGD with the momentum parameter 0.9 and learning rate 0.01.

We consider black-box auditing of four choices for $\mathcal{A}$ to audit their privacy. The algorithms differ based on the initialization and output perturbation: ($\mathcal{A}_1$): fixed initial, no output perturbation; ($\mathcal{A}_2$): random initialization, no output perturbation; ($\mathcal{A}_3$): fixed initial, output perturbation. ($\mathcal{A}_4$): random initialization, output perturbation. Output perturbation is performed by adding i.i.d. noise from $\mathcal{N}(0, \sigma^2)$ to components of the trained model and releasing the noisy model. All algorithms are run for 200 epochs with a minibatch size of 100.

Attack performances and privacy estimation: The attack performance of the MIA in Section 3 on the outputs of $\mathcal{A}_{1:4}$ are shown in Figure 4. The error counts are obtained with the procedure in Alg. 5 run for each algorithm. The algorithms with output perturbation used $\sigma = 0.1$. In each plot, each dot is a value $(X_i(\alpha^*), Y_i(\alpha^*))$, where α^* is the target type I error for the MIA. For the same challenge base (D_i, z_i), several points $(X_i(\alpha^*), Y_i(\alpha^*))$ are obtained by using the $\alpha^* \in \{0.01, 0.02, \dots, 0.99\}$, and those points are joined by a line.

We observe that the random initialization affects the performance of the attacks visibly when output perturbation is not used ($\mathcal{A}_1$ vs $\mathcal{A}_2$). However, the effect of random initialization significantly drops when output perturbation is used. We also see that some challenge bases (D_i, z_i) allow significantly better detection than others. This is expected since the challenge bases are drawn at random. Strategies to craft worst-case scenarios [23] can be used to eliminate those challenge bases for which the error lines are close to the $x + y = 1$ line.

We turn to privacy estimation. We choose a single (X, Y) point for each challenge base to feed `MCMC-DP-Est` in Alg. 1 with $n = 20$ observations; those points are $(X_i(0.1), Y_i(0.1))$ for each $i = 1, \dots, 20$. `MCMC-DP-Est` is run with $K = 1000$ auxiliary variables for 10^5 iterations, and the first 10^4 samples are discarded as burn-in. We used $\epsilon \sim \log \mathcal{N}_{[0,\infty)}(0, 10)$, one-sided normal distribution, and $s \sim \text{Unif}(0, 1) = \text{Beta}(1, 1)$ for the priors of ϵ and s. For $g(X_i, Y_i | \alpha_i, \beta_i)$, a bivariate normal approximation in (7) is used to account for the dependency among the decisions produced by Alg. 5. The two additional parameters τ, ρ are estimated within `MCMC-DP-Est`, as described in Section 2 of the Supplementary File.

The 2D histograms at the bottom row of Figure 4 are the posterior distributions of (ϵ, s) constructed from the samples provided by `MCMC-DP-Est`. The estimates of (ϵ, s) are as expected (e.g., randomness decreases ϵ) and are consistent with the attack performances. The estimates for s suggest that with more randomness in training, either the loss-based attack loses its power (relative to the best theoretical attack) or some challenge bases are no longer informative.

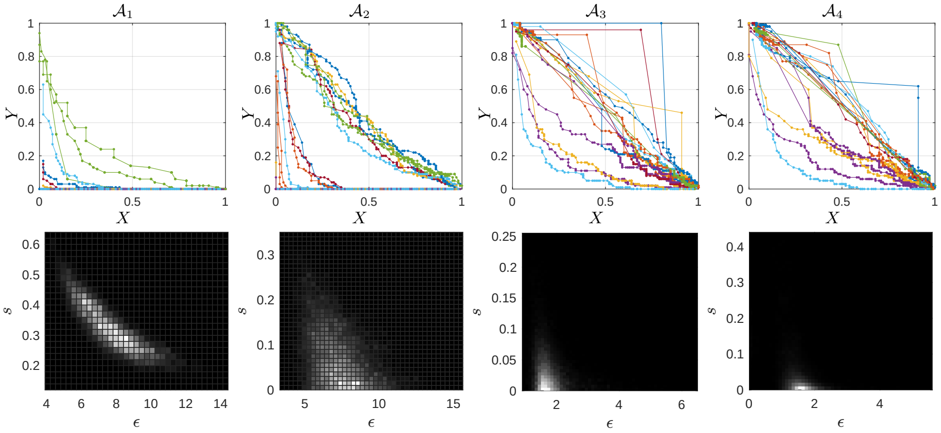

Fig. 4: (X, Y) counts for $\mathcal{A}_1, \ldots, \mathcal{A}_4$. For output perturbation, $\sigma = 0.1$ was used.

We repeat the experiments for $\mathcal{A}_4$ with $\sigma \in \{0.01, 0.05, 0.1\}$ for the output perturbation noise. Figure 5 shows that, as expected, increasing noise makes the attacks less accurate, which in turn causes smaller estimates for ϵ.

The estimates of `MCMC-DP-Est` across the audited algorithms are summarized in Table 1 with 90% CIs for ϵ. Figure 6 shows the sample autocorrelation functions (ACF) for ϵ-samples of all the runs of `MCMC-DP-Est`. The fast-decaying ACFs indicate a healthy (fast-mixing) chain. For further diagnosis, we also provide the trace plots of the samples for ϵ, s, τ, ρ in Section 2 of the Supplementary File.

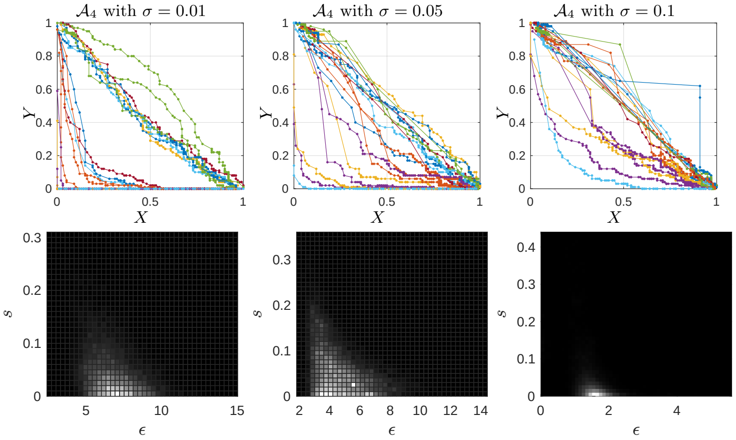

Fig. 5: Error counts and privacy estimation for $\mathcal{A}_4$ with $\sigma \in [0.01, 0.05, 0.1]$

Table 1: 90% Credible intervals for ϵ

	$\mathcal{A}_1$	$\mathcal{A}_2$	$\mathcal{A}_3$	$\mathcal{A}_4$	$\mathcal{A}_4$	$\mathcal{A}_4$
			$(\sigma = 0.1)$	$(\sigma = 0.1)$	$(\sigma = 0.05)$	$(\sigma = 0.01)$
Lower	5.52	4.95	1.29	1.00	2.80	4.61
Upper	10.52	10.00	2.53	2.12	7.68	9.62

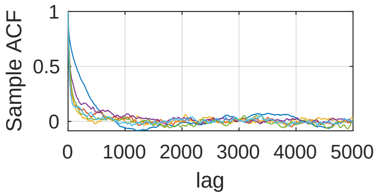

Fig. 6: Sample ACFs for the six algorithms in Table 1

5 Conclusion

In this work, we proposed a novel method for Bayesian estimation of differential privacy. Our algorithm leverages multiple (D, z) pairs and attacks to refine the privacy estimates and makes no assumptions about the strength of the attacks to avoid overconfident estimations. Beyond just credible intervals, the method provides the entire posterior distribution of the privacy parameter (as well as the average attack strength). Our experiments demonstrated that we can effectively estimate the privacy parameters of models trained under various randomness assumptions and that the resulting estimates align with attack performances.

We considered the "inclusion" versions of DP and MIAs where the pair of datasets differ by the inclusion/exclusion of a single point. The methodology similarly applies to the "replace" versions where dataset pairs are $D \cup \{z\}$ and $D \cup \{z'\}$ for a pair of z, z'.

In the real data experiments in Section 4.2, a single run of `MCMC-DP-Est` took ≈ 2.5 minutes on Matlab on a modern laptop, which is negligibly small compared to the time needed to collect the error counts. This suggests that `MCMC-DP-Est` can feasibly be used several times to estimate ϵ at different values of δ.

Limitations and future work: Given a challenge base (D, z), our statistical model considers the error counts for a single target value of type I error (i.e., a single point on each line of a plot in Figure 4). As suggested by [6], MIA performance tests would be utilized more effectively by using false negative and false positive counts at all decision thresholds to estimate its *profile function* $f(\alpha)$. Likewise, rather than (ϵ, δ)-DP, the f-DP [9] of $\mathcal{A}$ could be estimated as in [18,22], since f-DP has more complete information setting a lower bound on the profile functions of MIAs. Both extensions require prior specifications for random functions (e.g. *a la* Gaussian process priors), which we consider an important avenue for future work.

Acknowledgments. This work was supported by the European Union's Horizon Europe research and innovation program through the Twinning Project (enCRYPTON) under Grant 101079319 and the PHASE IV AI Project under Grant 101095384. This work is funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), `https://www.tensorflow.org/`, software available from tensorflow.org
2. Andrew, G., Kairouz, P., Oh, S., Oprea, A., McMahan, H.B., Suriyakumar, V.M.: One-shot empirical privacy estimation for federated learning. In: 12th Int. Conf. on Learning Rep. (2024)
3. Andrieu, C., Yıldırım, S., Doucet, A., Chopin, N.: Metropolis-hastings with averaged acceptance ratios (2020), `https://arxiv.org/abs/2101.01253`
4. Balle, B., Barthe, G., Gaboardi, M.: Privacy amplification by subsampling: tight analyses via couplings and divergences. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. p. 6280–6290. NIPS'18, Curran Associates Inc., Red Hook, NY, USA (2018)
5. Bun, M., Steinke, T.: Concentrated differential privacy: Simplifications, extensions, and lower bounds. In: Hirt, M., Smith, A. (eds.) Theory of Cryptography. pp. 635–658. Springer Berlin Heidelberg, Berlin, Heidelberg (2016)
6. Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., Tramèr, F.: Membership inference attacks from first principles. In: 2022 IEEE Symp. on Security and Privacy (SP). pp. 1897–1914 (2022)
7. Chollet, F., et al.: Keras. `https://keras.io` (2015)
8. Deng, L.: The mnist database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine **29**(6), 141–142 (2012)
9. Dong, J., Roth, A., Su, W.J.: Gaussian differential privacy. Journal of the Royal Statistical Society Series B: Statistical Methodology **84**(1), 3–37 (02 2022)
10. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) Automata, Languages and Programming. pp. 1–12. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
11. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) Theory of Cryptography. pp. 265–284. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
12. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science **9**(3–4), 211–407 (2014)

13. Hyland, S.L., Tople, S.: An Empirical Study on the Intrinsic Privacy of SGD (2022), https://arxiv.org/abs/1912.02919
14. Jagielski, M., Ullman, J., Oprea, A.: Auditing differentially private machine learning: how private is private sgd? In: Proc. of the 34th Int. Conf. on Neural Information Proc. Sys. NIPS '20, Curran Associates Inc., Red Hook, NY, USA (2020)
15. Kairouz, P., Oh, S., Viswanath, P.: The composition theorem for differential privacy. In: Bach, F., Blei, D. (eds.) Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 37, pp. 1376–1385. PMLR, Lille, France (07–09 Jul 2015)
16. Kairouz, P., Oh, S., Viswanath, P.: The composition theorem for differential privacy. IEEE Trans. Inf. Theor. **63**(6), 4037–4049 (Jun 2017)
17. Kupper, L.L., Haseman, J.K.: The use of a correlated binomial model for the analysis of certain toxicological experiments. Biometrics **34**(1), 69–76 (1978)
18. Leemann, T., Pawelczyk, M., Kasneci, G.: Gaussian membership inference privacy. In: Thirty-seventh Conference on Neural Information Processing Systems (2023)
19. Lu, F., Munoz, J., Fuchs, M., LeBlond, T., Zaresky-Williams, E.V., Raff, E., Ferraro, F., Testa, B.: A general framework for auditing differentially private machine learning. In: Oh, A.H., Agarwal, A., Belgrave, D., Cho, K. (eds.) Advances in Neural Information Processing Systems (2022)
20. Maddock, S., Sablayrolles, A., Stock, P.: CANIFE: Crafting Canaries for Empirical Privacy Measurement in Federated Learning. In: ICLR (2023)
21. Mironov, I.: Rényi Differential Privacy . In: 2017 IEEE 30th Computer Security Foundations Symposium (CSF). pp. 263–275. IEEE Computer Society, Los Alamitos, CA, USA (Aug 2017). https://doi.org/10.1109/CSF.2017.11
22. Nasr, M., Hayes, J., Steinke, T., Balle, B., Tramèr, F., Jagielski, M., Carlini, N., Terzis, A.: Tight auditing of differentially private machine learning. In: Proc. of the 32nd USENIX Conf. on Security Symposium. SEC '23, USENIX, USA (2023)
23. Nasr, M., Songi, S., Thakurta, A., Papernot, N., Carlin, N.: Adversary instantiation: Lower bounds for differentially private machine learning. In: 2021 IEEE Symposium on security and privacy (SP). pp. 866–882. IEEE (2021)
24. Nasr, M., Steinke, T., Balle, B., Choquette-Choo, C.A., Ganesh, A., Jagielski, M., Hayes, J., Thakurta, A.G., Smith, A., Terzis, A.: The last iterate advantage: Empirical auditing and principled heuristic analysis of differentially private SGD. In: The 13th Int. Conf. on Learning Rep. (2025)
25. Pillutla, K., Andrew, G., Kairouz, P., McMahan, H.B., Oprea, A., Oh, S.: Unleashing the power of randomization in auditing differentially private ml. In: Proc. of the 37th International Conference on Neural Information Processing Systems. NIPS '23, Curran Associates Inc., Red Hook, NY, USA (2023)
26. Sablayrolles, A., Douze, M., Schmid, C., Ollivier, Y., Jégou, H.: White-box vs black-box: Bayes optimal strategies for membership inference. In: International Conference on Machine Learning (2019)
27. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership Inference Attacks Against Machine Learning Models. In: 2017 IEEE Symposium on Security and Privacy (SP). pp. 3–18. IEEE Computer Society, Los Alamitos, CA, USA (May 2017). https://doi.org/10.1109/SP.2017.41
28. Steinke, T., Nasr, M., Jagielski, M.: Privacy auditing with one (1) training run. In: Thirty-seventh Conference on Neural Information Processing Systems (2023)
29. Ye, J., Maddi, A., Murakonda, S.K., Bindschaedler, V., Shokri, R.: Enhanced membership inference attacks against machine learning models. In: Proc. of the 2022 ACM SIGSAC Conf. on Computer and Comm. Security. p. 3093–3106. CCS '22, ACM, New York, NY, USA (2022). https://doi.org/10.1145/3548606.3560675

30. Yeom, S., Giacomelli, I., Fredrikson, M., Jha, S.: Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting . In: 2018 IEEE 31st Computer Security Foundations Symposium (CSF). pp. 268–282. IEEE Computer Society, Los Alamitos, CA, USA (Jul 2018). https://doi.org/10.1109/CSF.2018.00027
31. Zanella-Beguelin, S., et al.: Bayesian estimation of differential privacy. In: Proc. of the 40th Int. Conf. on Machine Learning. vol. 202, pp. 40624–40636. PMLR (2023)