Automated Privacy Risk Estimation of Limited Fixed Aggregate Statistics

Yifeng Mao*
Imperial College London
London, United Kingdom
y.mao22@imperial.ac.uk

Bozhidar Stevanoski*
Imperial College London
London, United Kingdom
b.stevanoski@imperial.ac.uk

Yves-Alexandre de Montjoye
Imperial College London
London, United Kingdom
demontjoye@imperial.ac.uk

Abstract

Empirical inference attacks are a popular approach for evaluating the privacy risk of data release mechanisms in practice. While an active attack literature exists to evaluate machine learning models or synthetic data release, we currently lack comparable methods for fixed aggregate statistics, in particular when only a limited number of statistics are released. We here propose an inference attack framework against fixed aggregate statistics and an attribute inference attack called DeSIA (Deterministic-Stochastic Inference Attack). DeSIA consists of two modules: a deterministic module, which verifies whether the target user's attribute can be uniquely inferred, and a stochastic module, which estimates the most likely attribute value when it cannot be uniquely inferred. We instantiate DeSIA against the U.S. Census PPMF dataset and show it to identify risks missed by reconstruction-based attacks. In particular, we show DeSIA to be highly effective in identifying vulnerable users, achieving a true positive rate of 0.14 at a false positive rate of 10^{-3}. We show DeSIA to outperform all reconstruction-based attacks even without access to a real-world auxiliary dataset. We show DeSIA to be robust to varying levels of noise addition and across varying numbers of released aggregate statistics. We perform an extensive ablation study of DeSIA and show how DeSIA can be successfully adapted to the membership inference task. Overall, our results show that (a) even a limited number of aggregate statistics can reveal sufficient information to expose users to risk from inference attacks, leaving some users disproportionately vulnerable, and (b) emphasize the need for formal privacy mechanisms and testing before aggregate statistics are released. [1]

Keywords

attribute inference attacks, aggregate statistics, privacy

1 Introduction

Empirical inference attacks are a popular tool to evaluate the privacy risks of data release mechanisms such as machine learning models and synthetic data. Membership inference and attribute inference attacks have been used extensively to evaluate the privacy of machine learning models and synthetic data in practice, showing models to memorize training data [1] and detecting privacy violations [2]. Data protection laws, such as GDPR [3], furthermore define anonymous data as the absence of empirical attacks.

We, however, currently lack a comparable inference attack framework and methods against fixed aggregate statistics, in particular when only a limited number of statistics are released. The literature on attacks against fixed aggregate statistics has instead focused on reconstruction attacks. As of today, the state-of-the-art reconstruction attacks against fixed aggregate statistics are CIP and RAP. CIP, proposed by the U.S. Census Bureau [4], models the attack as a constraint integer programming problem. RAP, proposed by Dick et al. [5], uses synthetic data to reconstruct tabular data from aggregate statistics.

Contributions. We here propose a framework and a method for attribute inference attacks against fixed aggregate statistics from tabular data. Our method, DeSIA (Deterministic-Stochastic Inference Attack), consists of two modules: a deterministic and a stochastic module. The deterministic module, which we instantiate as a modified constraint integer programming problem, deterministically verifies if there is only one feasible sensitive attribute to be inferred. The stochastic module, which we instantiate as a machine learning model trained on an auxiliary dataset, infers the most likely attribute when the deterministic module cannot verify a unique feasible value.

We instantiate our attack against the PPMF release from the U.S. Census Bureau and show DeSIA to strongly outperform state-of-the-art reconstruction-based attacks when instantiated on the attribute inference task. DeSIA effectively identifies highly vulnerable users, achieving a 0.14 true positive rate (TPR) at a false positive rate (FPR) of 10^{-3}. It also performs well with a synthetically-generated auxiliary data instead of a real-world auxiliary dataset. We then evaluate DeSIA under varying levels of per-query noise addition and varying numbers of released aggregate statistics. We also perform an extensive ablation study of DeSIA, showing all the components to be necessary for it to perform well. We finally show how DeSIA can be successfully adapted to the membership inference task (MIA); here again, strongly outperforming reconstruction-based baselines. Overall, our results show practical information leakage to occur even when a relatively small number of fixed aggregate statistics are being released, leaving some users particularly vulnerable to inference attacks, and emphasize the need for formal privacy mechanisms and testing before aggregate statistics are released.

The paper is organized as follows. Section 2 introduces the threat model and formalizes the framework. Section 3 introduces our method DeSIA. Sections 4 and 5 describe the experimental setup and present the results. Section 6 discusses the adaptation of DeSIA to the membership inference task and Section 7 discusses additional robustness analysis and efficiency of our method. Section 8

*Equal contribution
[1]The code for this paper is available at https://github.com/imperial-aisp/desia.

Proceedings on Privacy Enhancing Technologies 2026(3), 100–117

https://doi.org/10.56553/popets-2026-0073

presents the related work, Section 9 discusses the limitations of our framework, and finally, Section 10 concludes the paper.

2 Background

2.1 Fixed Aggregate Statistics

We consider a data curator handling a private tabular dataset D from a distribution $\mathcal{D}$. The dataset D consists of records for a set of s users U, $s = |U|$, over n attributes, $A = \{a_1, \ldots, a_n\}$. Each attribute $a_i \in A$ can take values from a set $\mathcal{V}_i, i \in \{1, \ldots, n\}$, with the set of all such sets being $\mathcal{V}$, $\mathcal{V} = \{\mathcal{V}_1, \ldots, \mathcal{V}_n\}$. We denote the record of user $u \in U$ as $r_u = (r_u^1, \ldots, r_u^n)$, where r_u^i is a value from the set $\mathcal{V}_i$, $r_u^i \in \mathcal{V}_i, i \in \{1, \ldots, n\}$. We denote the projection of the user record r_u over a subset of attributes $A' = \{a_{i_1}, \ldots, a_{i_k}\} \subseteq A$ as $r_u^{A'} = (r_u^{i_1}, \ldots, r_u^{i_k})$.

The data curator aims to release aggregate statistics from the dataset D, where each aggregate statistic counts the number of records in a particular subset of D. Formally, we define a counting aggregate statistic q as $q = (V_1^q, \ldots, V_n^q)$, given sets V_i^q of values corresponding to each attribute a_i, $V_i^q \subseteq \mathcal{V}_i$. We denote the subset of users who contribute to the statistic q as $U_q \subseteq U, U_q = \{u | r_u^1 \in V_1^q \wedge \cdots \wedge r_u^n \in V_n^q\}$ and we call U_q the userset of the aggregate statistic q. We denote the result of q evaluated on D by $q(D)$, $q(D) = |U_q| = \sum_{u \in U} \mathbb{1}(r_u^1 \in V_1^q \wedge \cdots \wedge r_u^n \in V_n^q)$. Equivalently, the aggregate statistic q can be expressed as a counting query using SQL notation as:

```
SELECT count(*) FROM D
WHERE a_1 IN V_1^q AND ... AND a_{n-1} IN V_{n-1}^q      (1)
      AND a_n IN V_n^q.
```

2.2 Threat Model: Attribute Inference Attacks Against Fixed Aggregate Statistics

We evaluate the privacy leakage of releasing fixed aggregate statistics from a tabular dataset in the context of attribute inference attacks (AIAs).

We assume the data curator releases a fixed set of m aggregate statistics, $Q = \{q_1, \ldots, q_m\}$ and their values on the dataset D, $Q(D) = \{q_1(D), \ldots, q_m(D)\}$. We assume that fewer aggregates are released than the number of users in D, $m < s$, i.e., that on average there is fewer than one aggregate per user, $\frac{m}{s} < 1$. We evaluate various values of $\frac{m}{s}$ in Section 5.4.

We assume that one of the attributes is a sensitive attribute. Without loss of generality, we assume a_n is sensitive and denote the set of non-sensitive attributes as $A' = \{a_1, \ldots, a_{n-1}\}$.

Attacker's Goal. The attacker's goal is to infer the value of the sensitive attribute a_n of a target user u^*, i.e., to infer $r_{u^*}^n$.

Attacker's knowledge. In line with the literature, we assume the attacker to have auxiliary knowledge about the dataset D and knowledge about the target user u^*.

The auxiliary knowledge of the dataset D consists of knowledge of its size s and access to an auxiliary dataset D_{aux} from a distribution $\mathcal{D}_{aux}$, similar to $\mathcal{D}$. We relax the assumption about access to an auxiliary dataset in Section 5.2. We also assume the attacker to know the possible values $\mathcal{V}_i$ for every attribute $a_i, i \in \{1, \ldots, n\}$.

Knowledge about the target user u^* consists of the values of the non-sensitive attributes A' of the target user u^*, $r_{u^*}^{A'} = (r_{u^*}^1, \ldots, r_{u^*}^{n-1})$ and the knowledge that the target user is unique in the dataset D with those values for the non-sensitive attributes[2], $\forall u' \in U, u' \neq u^*, r_{u'}^{A'} \neq r_{u^*}^{A'}$.

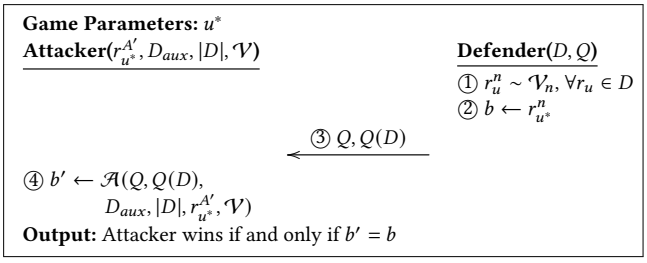

Figure 1: Privacy game for attribute inference attack against fixed aggregate statistics. The game measures risk from the released aggregates only, excluding imputation (see Section 2.3 for "in the wild" risk).

2.3 Framework for Attribute Inference Attack as a Privacy Game

We propose a formal framework for user-specific attribute inference attacks against fixed aggregates as a privacy game. The privacy game (Fig. 1) is played between two players, an attacker and a defender. The defender computes and releases aggregate statistics Q and their values on the dataset D, $Q(D)$, while an attacker aims to infer the value of the sensitive attribute of the target user u^*, $r_{u^*}^n$, using the released information from defender, the knowledge of u^*, and an auxiliary dataset. The game consists of a total of four steps. The defender plays the first three steps and the attacker plays the fourth one.

First, in line with the literature [6, 7], the defender samples the values of the sensitive attribute uniformly at random from $\mathcal{V}_n$ for all users U in the protected dataset D. This both provides a random baseline and circumvents the imputation issue (see Section 8), allowing us to measure the privacy risk posed solely by the release of the aggregate statistics.

Second, the defender saves the value sampled for the sensitive attribute of target user u^*, which we denote as b. Note that the value b is unambiguous as we assume, for simplicity and in line with the literature [6–8], that target user u^* is unique in D for $r_{u^*}^{A'}$.

Third, the defender evaluates the aggregate statistics Q on the protected dataset D *with randomized values* of the sensitive attribute and releases them to the attacker.

Fourth, the attacker aims to predict the sensitive value of the target user b by using the released aggregate statistics. The attacker runs an attack $\mathcal{A}$ to obtain a value b'.

The attacker wins the game if and only if the predicted value b' is equal to the value b of the target user's sensitive attribute in D, $b' = b$.

Measured vs. "in the wild" attribute inference risk. Note that the first step of the privacy game is specifically designed to

[2]We relax the assumption about the uniqueness of the target user's non-sensitive attributes in Appendix E, where, to keep the problem well-posed with an unambiguous ground-truth value, we extend our evaluation to value-unique users: those sharing both non-sensitive and sensitive values.

isolate the attacker's performance to the information leakage stemming from the released aggregates. In general, an attacker's inference success "in the wild" is shaped by three components: (a) the marginal distribution of the sensitive attribute, if skewed and known, (b) the correlations between sensitive and non-sensitive attributes, if strong and known, and (c) the information leakage induced by the released aggregates. By sampling the sensitive attribute uniformly at random from the set of possible values $\mathcal{V}_n$ in the first step of the game, we intentionally eliminate the effect of the first two components: the marginal distribution becomes uniform, and correlations between sensitive and non-sensitive attributes are broken, eliminating any advantage an attacker could gain from knowing either of these two. Consequently, the attacker must rely exclusively on the third component. In this sense, we evaluate a worst-case scenario from the attacker's perspective, in which they cannot exploit imputation (first two components), and we provide a lower bound to the overall risk "in the wild" if they could. In Appendix B, we consider an alternative game that preserves both marginal distributions and correlations and show that even without any aggregate release, a simple imputation attack using auxiliary data can achieve near-perfect inference performance.

2.4 State-of-the-art: Reconstruction-Based Attacks against Fixed Aggregate Statistics

Existing literature on empirical attacks against fixed aggregate statistics has primarily focused on reconstruction attacks. Reconstruction attacks aim to reconstruct the private dataset D by using the released statistics Q and their values on D, $Q(D)$. They search for a dataset D' such that the values of the released aggregate statistics Q on D', $Q(D')$, exactly or closely match the released values from the protected dataset D, $Q(D') \approx Q(D)$, with the ideal goal being perfect equality $Q(D') = Q(D)$. The dataset D' is called a tentative reconstructed dataset.

The state-of-the-art attacks are an attack that uses constraint integer programming [4] (CIP) and an attack that uses synthetic data generation techniques [5] to search for a tentative reconstructed dataset (RAP).

2.4.1 CIP: Constraint Integer Programming Attack. Abowd et al. [4] proposed an attack, which we refer to as CIP, that aims to find a tentative reconstructed dataset D' by formulating a constraint integer programming task. The main idea of CIP is to represent a dataset D' by using the multiplicity of all potential records $(v_1, \ldots, v_n)$, $\forall v_i \in \mathcal{V}_i$ in D' as integer variables $x_{(v_1,\ldots,v_n)}$ [3]. For example, a record that is not a member of D' has a multiplicity of 0.

Formally, the constraint integer programming task is composed of three main components: a set of unknown variables X, sets of possible values for each unknown variable (also called domain sets) S, and constraints C that specify allowable combinations of unknown variables.

Each **unknown variable**, $x_{(v_1,\ldots,v_n)} \in X$, represents the multiplicity of a record $(v_1, \ldots, v_n), v_i \in \mathcal{V}_i$ in D'. Note that although there are $\prod_{i=1}^{n} |\mathcal{V}_i|$ unknown variables, many of them may have values of 0 since they can represent records that are not members of the dataset D'.

Each **domain**, $s_{(v_1,\ldots,v_n)} \in S$, contains all possible values for the unknown variable $x_{(v_1,\ldots,v_n)}$. Here, a domain contains all possible integers between 0 and the size of the protected dataset D, inclusively, $x_{(v_1,\ldots,v_n)} \in \{0, 1, \ldots, s\}$

Each **constraint** $c_q \in C$, represents an allowable combination of variables given a released aggregate statistic $q = (V_1^q, \ldots, V_n^q)$, $q \in Q$. The constraint c_q limits the possible values for the multiplicity of records by taking the aggregate value $q(D)$:

$$c_q : \sum_{\forall r \in \mathcal{V}_1^q \times \cdots \times \mathcal{V}_n^q} x_r = q(D).$$

The solution to the constraint integer programming task is an assignment function that assigns values to all unknown variables. We denote the assignment function as $\sigma : X \to \mathbb{Z}$. Abowd et al. use a third-party solver [9] to obtain a tentative reconstructed dataset D', which we also use.

We consider two variants of this attack, depending on the initialization of the third-party solver. The attack originally proposed by Abowd et al. [4], which we refer to as **CIP-rand**, where the third-party solver initializes all unknown variables to zeros by default. For a fair comparison, we also use an adaptation of the attack, which we refer to as **CIP-init**, that uses the access of the auxiliary dataset D_{aux} to initialize the unknown variables $x_{(v_1,\ldots,v_n)}$ in the third-party solver with the multiplicity of the records in D_{aux}.

2.4.2 RAP: Synthetic Data Attack. Dick et al. [5] proposed an attack, which we refer to as RAP, that produces a tentative reconstructed dataset D' by using a synthetic data generator model.

The attack trains a synthetic data generator model M that takes as input aggregate statistics Q and their values $Q(D)$ on the protected dataset D and outputs a tentative reconstructed dataset D', The model M is trained to minimize the L2-distance between the aggregate values on the synthetic dataset D' and the protected dataset D, $\min \|Q(D') - Q(D)\|_2$.

We use the two variants of this attack proposed by Dick et al., depending on the initialization of model M: **RAP-rand** where M is initialized with a uniformly at random sample from $\mathcal{V}_1 \times \cdots \times \mathcal{V}_n$, and **RAP-init** where M is initialized with auxiliary dataset D_{aux}.

We would like to note first that while a perfect reconstruction attack would fully reconstruct the private dataset D and thus be a perfect attribute inference attack, this is rarely the case in practice, including when a large number of aggregate statistics are released, e.g., by the U.S. Census Bureau. Second, we assume, in line with the broader literature on inference attacks [5–8, 10–15], an attacker weaker than the strong Differential Privacy (DP) attacker but with access to a real-world auxiliary dataset. For fair comparison, we adapt the CIP attack to take advantage of the auxiliary dataset. The RAP attack already proposes a variant leveraging an auxiliary dataset, which we use here. Note that we relax this assumption in Section 5.2 and show DeSIA to perform well without it. Third, we assume an attacker with access to at least some information about the target record, but note that, while this is necessary for any attribute or membership inference attack, some threat models using reconstruction attacks do not need this assumption, e.g., identifying

[3] Note that while Abowd et al. [4] have implemented an equivalent formulation using binary variables, for simplicity, we here present and use the integer variable formulation.

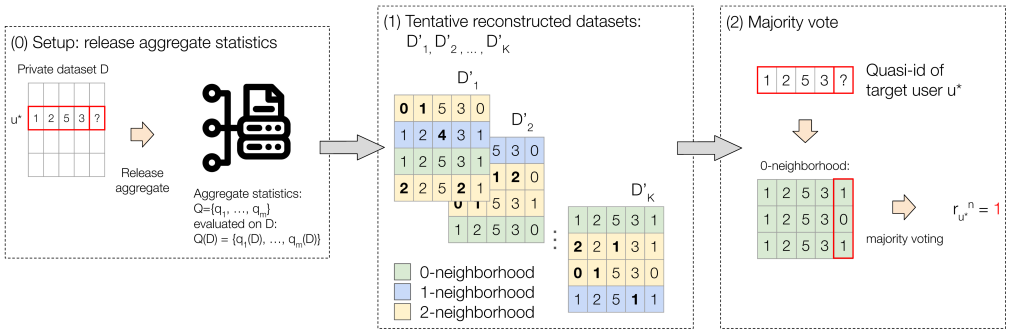

Figure 2: Adaptation of reconstruction attacks for attribute inference.

that an individual with a certain characteristic lives in a block. We do not consider this threat here.

2.5 Instantiating Reconstruction Attacks for Attribute Inference

In most cases, in particular when a limited number of aggregate statistics are released, there is more than one tentative reconstructed dataset D' [4]. We adapt existing reconstruction attacks to the attribute inference task by (a) generating K tentative reconstructed datasets $D'_1, \ldots, D'_K$ and (b) taking a majority vote after combining the values of the target user's sensitive attribute from all K datasets. Figure 2 illustrates both steps.

Obtaining tentative reconstruction datasets. For RAP, we follow the approach by Dick et al. [5] and train K synthetic data generator models, each with a different random seed, and generate a tentative reconstructed dataset using each of the K models. For CIP, we follow a similar approach by applying the solver K times, each time with a different seed and order of constraints.

Majority vote in smallest existing neighborhood. We define $L-neighborhood$ as the records from all K tentative reconstructed datasets that differ at L values from the values of the non-sensitive attributes of the target user, $r_{u^*}^{A'}$:

$$L{-}neighborhood(r_{u^*}^{A'}, D'_1, \ldots, D'_K) = \{r | \sum_{i=1}^{n-1} \mathbb{1}(r^i \neq r^i_{u^*}) = L, r \in \bigcup_{j\in\{1,\ldots,K\}} D'_j\}.$$

For example, the 0-neighborhood is the set of records from all K tentative reconstructed datasets that have the same values for the non-sensitive attributes A' as the target user, $0{-}neighborhood = \{r | r^{A'} = r_{u^*}^{A'} r \in \bigcup_{j\in\{1,\ldots,K\}} D'_j\}$.

For both RAP and CIP, we start from the 0-neighborhood of the target user. The attacker infers the sensitive attribute of the target user, $r^n_{u^*}$, by predicting the most frequent value for attribute a_n of all records in the 0-neighborhood.

$$r^n_{u^*} = \arg\max count(r^n),\ s.t.\ r \in 0{-}neighborhood(r_{u^*}).$$

If the $0{-}neighborhood$ is empty, we extend to the $1{-}neighborhood$. Following the same principle, if the $(L-1){-}neighborhood$ is empty, we extend to $L{-}neighborhood$, until $L = n-1$, when we take all records in all tentative reconstructed datasets $\bigcup_{i\in\{1,\ldots,K\}} D'_i$.

If there is a tie for the most frequent value between two (or more) values in a given $L-neighborhood$, we select one uniformly at random from them.

3 Methodology

Our method, DeSIA (Deterministic-Stochastic Inference Attack) is composed of two modules: a deterministic module $\mathcal{M}_D$ and a stochastic module $\mathcal{M}_S$. The deterministic module $\mathcal{M}_D$, which we instantiate as a modified constraint integer programming problem, aims to identify if one and only one value for the sensitive attribute of the target user is deterministically feasible, given the released statistics, and to output that value. If there is more than one feasible value, the stochastic module $\mathcal{M}_S$ is then used to estimate the likelihood of the feasible values and to output the most likely one.

$$\text{DeSIA} = \begin{cases} \mathcal{M}_D, & \mathcal{M}_D(\cdot) \neq \varnothing \\ \mathcal{M}_S, & otherwise \end{cases}$$

3.1 Deterministic Attribute Inference

The deterministic module $\mathcal{M}_D$ first finds a feasible value $v_n^* \in \mathcal{V}_n$ for the sensitive attribute of the target user, $r^n_{u^*}$, and second, deterministically verifies that the value v_n^* is the only feasible value. We instantiate both tasks as constraint integer programming tasks (see Algorithm 1).

Finding a feasible value. We use constraint integer programming to obtain a possible value $v_n^* \in \mathcal{V}_n$ for the target user u^*. In particular, we define the unknown variables x_r for record $r = (v_1, \ldots, v_n), v_i \in \mathcal{V}_i$ following the CIP attack.

We optimize the execution time of the method by updating the domain of each unknown variable to lower the range of possible values, using information from the aggregate statistics. We set the maximal integer s_r^{max} in the domain set of the unknown variable x_r to be the minimal value of all relevant aggregate statistics:

$$s_r^{max} = min(s, \min_{q\in Q} q(D)).$$
$$v_i \in V_i^q, \forall i \in \{1,\ldots,n\}$$

We crucially also extend the set of constraints by adding a constraint specific to the target user u^*, which we call a uniqueness constraint. The constraint reflects the attacker's knowledge of (a) the values of the nonsensitive attributes of the target user, $r_{u^*}^{A'}$, and

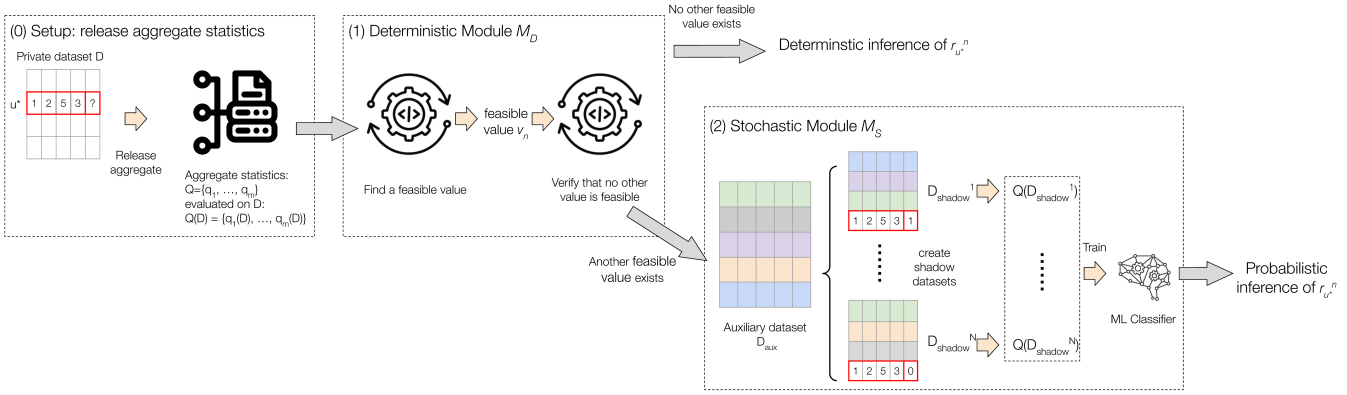

Figure 3: Our proposed attack. It uses two modules: (1) $\mathcal{M}_D$ deterministically checking if there is only one possible value for the target user's sensitive attribute, and (2) $\mathcal{M}_S$ stochastically predicting the most likely value.

(b) the uniqueness of the in D given $r_{u^*}^{A'}$:

$$c_t : 1 = \sum_{v_n \in \mathcal{V}_n} x_{(r_{u^*}^1,\ldots,r_{u^*}^{n-1},v_n)}.$$

We then use a third-party solver [9] to solve the constraint integer problem. The solver returns the assignment of values σ as a solution, which maps from an unknown variable x_r to its assigned value $\sigma(x_r)$.

We select the variables $\{x_{(r_{u^*}^1,\ldots,r_{u^*}^{n-1},v_n)} \mid \forall v_n \in \mathcal{V}_n\}$, which correspond to records that share the same values on non-sensitive attributes as the target user. Because of the uniqueness constraint, σ only assigns one of the selected variables to be larger than zero. We denote the value of the sensitive attribute of that variable as $\hat{v_n}$.

Verifying the feasible value. The feasible value $\hat{v_n}$ found for the sensitive attribute of the target user $r_{u^*}^n$ might not be unique. We here formulate another constraint integer task to verify whether the obtained value $\hat{v_n}$ is the only feasible value for $r_{u^*}^n$, explicitly disallowing $\hat{v_n}$ as a possible value for $r_{u^*}^n$ through a null constraint:

$$c_{t'} : 0 = x_{(r_{u^*}^1,\ldots,r_{u^*}^{n-1},\hat{v_n})}.$$

If we do not obtain any feasible assignment function σ', $\mathcal{M}_D$ verifies that the value $\hat{v_n}$ is the only deterministically possible value for $r_{u^*}^n$. If this is the case, we consider the target user u^* to be deterministically vulnerable and assign it a certainty of 1. If it is not, we use the stochastic module $\mathcal{M}_S$ to infer an attribute value.

3.2 Stochastic Attribute Inference

The stochastic module $\mathcal{M}_S$ estimates the likelihood of the feasible values for $r_{u^*}^n$ and outputs the most likely one (see Algorithm 2). In particular, it uses a machine learning model trained on aggregate statistics from datasets similar to the protected dataset D. Note that $\mathcal{M}_S$ learns only from the released aggregate statistics and does not leverage correlations in the original protected dataset D. The stochastic module is composed of three steps.

First, we create N datasets, $D_{shadow}^1, \ldots, D_{shadow}^N$, which we call shadow datasets. Each shadow dataset D_{shadow}^i consists of $s-1$ users sampled without replacement from the auxiliary dataset D_{aux} and the target user u^*. We project their records on non-sensitive

Algorithm 1: Deterministic attribute inference

Input: Aggregate statistics $Q = \{q_1, \ldots, q_m\}$
Aggregate values $Q(D) = \{q_1(D), \ldots, q_m(D)\}$
Dataset size s
Attribute domains $\{\mathcal{V}_1, \ldots, \mathcal{V}_n\}$
Non-sensitive attributes $(r_{u^*}^1, \ldots, r_{u^*}^{n-1})$
Output: Prediction on $r_{u^*}^n$ if deterministically possible, otherwise NaN.

Unknown Variable Set $X \leftarrow \emptyset$, Domain Set $S \leftarrow \emptyset$, Constraint Set $C \leftarrow \emptyset$ // Initialize CIP problem
for $r \in \mathcal{V}_1 \times \mathcal{V}_2 \times \cdots \times \mathcal{V}_n$ **do**
  Define x_r // Define the new variable
  $s_r^{max} \leftarrow \min(s, \min_{\forall q_i \in Q, r \in \mathcal{V}_1^{q_i} \times \cdots \times \mathcal{V}_n^{q_i}} q_i(D))$ // Obtain the minimal upper bound for the new variable
  $X \leftarrow X \cup \{x_r\}, S \leftarrow S \cup \{0, \ldots, s_r^{max}\}$ // Update the variable and domain set
for $q_i \in Q$ *and* $q_i(D)$ **do**
  $c_i : \sum_{r \in \mathcal{V}_1^{q_i} \times \cdots \times \mathcal{V}_n^{q_i}} x_r = q_i(D)$ // Define constraint of q_i
  $C \leftarrow C \cup \{c_i\}$ // Update the constraint set
$c_t : \sum_{v_n \in \mathcal{V}_n} x_{(r_{u^*}^1,\ldots,r_{u^*}^{n-1},v_n)} = 1$ // Define uniqueness constraint
$C \leftarrow C \cup \{c_t\}$
$\sigma \leftarrow$ **RunSolver**(X, S, C)
if $\exists\ \sigma$ **then** // If there is at least one feasible solution
  for $v_n \in \mathcal{V}_n$ **do**
    if $\sigma(x_{(r_{u^*}^1,\ldots,r_{u^*}^{n-1},\hat{v_n})}) = 1$ **then**
      $c_{t'} : x_{(r_{u^*}^1,\ldots,r_{u^*}^{n-1},\hat{v_n})} = 0$ // Define null constraint
      $C' \leftarrow C \cup \{c_{t'}\}$ // Update the constraint set
  $\sigma' \leftarrow$ **RunSolver**(X, S, C') // If no feasible solution
  if $\nexists\ \sigma'$ **then**
    return $\hat{v_n}$ // Return the only possible value of sensitive attribute
return $\varnothing$

Algorithm 2: Stochastic attribute inference

Input: Aggregate statistics $Q = \{q_1, \ldots, q_m\}$
Aggregate values $Q(D) = \{q_1(D), \ldots, q_m(D)\}$
Dataset size s
Attribute domains $\{\mathcal{V}_1, \ldots, \mathcal{V}_n\}$
Non-sensitive attributes $(r^1_{u^*}, \ldots, r^{n-1}_{u^*})$
Auxiliary dataset D_{aux}
Number of shadow datasets N

Output: Prediction on $r^n_{u^*}$, based on the probability of each possible value in $\mathcal{V}_n$

for $i = 1, \ldots, N$ **do**
  $D^i_{shadow} \leftarrow \emptyset$, $R^i = \emptyset$ // Initialize shadow dataset and the set of selected records
  for $j = 1, \ldots, s-1$ **do**
    $r_{i,j} \sim (D_{\text{aux}} \setminus R^i)$ // Sample one record without replacement from auxiliary dataset
    $z_{i,j} \sim \mathcal{V}_n$ // Sample sensitive value at uniform random
    $R^i \leftarrow R^i \cup \{r_{i,j}\}$ // Add the sampled record to selected record set
    $D^i_{shadow} \leftarrow D^i_{shadow} \cup \{(r^{A'}_{i,j}, z_{i,j})\}$ // Update shadow dataset
  $z_{i,*} \sim \mathcal{V}_n$
  $D^i_{shadow} \leftarrow D^i_{shadow} \cup \{(r^{A'}_{u^*}, z_{i,*})\}$ // Add shadow record of target user to shadow dataset
  $Q(D^i_{shadow}) \leftarrow \{q_1(D^i_{shadow}), \ldots, q_m(D^i_{shadow})\}\}$ // Evaluate the aggregate statistics on shadow dataset
$M \leftarrow \{(Q(D^1_{shadow}, z_{1,*}), \ldots, (Q(D^N_{shadow}, z_{N,*})\}$ // Train meta-classifier
$r^n_{u^*} \leftarrow M(Q(D))$ // Predict the sensitive value of target user using trained meta-classifier
return $r^n_{u^*}$

attributes $A' = \{a_1, \ldots, a_{n-1}\}$ to obtain $\{r^{A'}_{i,1}, \ldots, r^{A'}_{i,s-1}\}$. Then, we sample $s-1$ values from $\mathcal{V}_n$ uniformly at random, denoted as $\{z_{i,1}, \ldots, z_{i,s-1}\}$. We append $z_{i,j}$ to $r^{A'}_{i,j}$, such that $z_{i,j}$ is the value for the sensitive attribute of the record $(r^{A'}_{i,j}, z_{i,j})$. We obtain a shadow record of the target user by appending a value $z_{i,*}$ sampled at uniform random from $\mathcal{V}_n$ to the values of non-sensitive attributes $r^{A'}_{u^*}$. The resulting shadow dataset D^i_{shadow} is as follow:

$$\{(r^{A'}_{i,1}, z_{i,1}), \ldots, (r^{A'}_{i,s-1}, z_{i,s-1}), (r^{A'}_{u^*}, z_{i,*})\}.$$

Second, we obtain the fixed aggregate statistics from the N shadow datasets $Q(D^1_{shadow}), Q(D^2_{shadow}), \ldots, Q(D^N_{shadow})$ by evaluating the released aggregate statistics Q on each shadow dataset.

Third, we train a machine learning model M which takes $Q(D^1_{shadow}), Q(D^2_{shadow}), \ldots, Q(D^N_{shadow})$ as input and the values $z_{1,*}, \ldots, z_{N,*}$ as labels. The model M aims to predict the sensitive attribute of the target user's record. We input the aggregates $Q(D)$ of the protected dataset D to the trained machine learning model to infer a predicted value of the sensitive attribute for the target user in D, $r^n_{u^*}$.

4 Experimental setup

4.1 Dataset and Aggregates

Privacy-Protected Microdata File (PPMF). In line with the literature [5], we evaluate our method on the 2020-05-27 vintage Privacy-Protected Microdata File (PPMF) [16]. The PPMF dataset contains record-level data from the U.S. Census Bureau for users across the U.S., partitioned by census block. We use the same subset of the released block-level aggregates as [5]. In particular, as the U.S. Census Bureau releases the aggregate statistics as contingency tables, we also use the same block-level contingency tables as the literature [5]. We list and briefly describe them in Table 1.

Table 2 presents the attributes, their types, and possible values in the PPMF dataset. We consider the attribute "Hispanic" to be the sensitive attribute and the other attributes of the target user to be known by the attacker. Note that we randomize the values of the sensitive attribute so as to avoid the imputation problem. This allows us to establish a random guess baseline as a random coin flip since the sensitive attribute "Hispanic" is binary.

We focus on the scenario where only a limited number of aggregates are released. In particular, we assume that the number of released aggregates, m, is smaller than the number of users s in the protected dataset D, $\frac{m}{s} < 1$ and randomly sample m aggregate statistics without replacement from Table 1. If not otherwise specified, we consider $\frac{m}{s} = 2^{-2}$.

The majority of the blocks are small, with 99% of them containing less than 454 records. We assume that the protected dataset D is in the top 1 percent largest datasets. In order to create D_{aux} and D, we select the 10 smallest blocks larger than 4540 users. We partition them randomly into two parts of sizes 10% and 90%, representing the private dataset D and the auxiliary dataset D_{aux}, respectively. We

Table name	Description
P1.	total population
P6.	race (total races tallied)
P7.	Hispanic/Latino by race (total races tallied)
P9.	Hispanic/Latino by race
P11.	Hispanic/Latino by race for the population aged 18+
P12.	sex by age
P12A.	sex by age for only race White
P12B.	sex by age for only race Black or African American
P12C.	sex by age for only race American Indian and Alaska native
P12D.	sex by age for only race Asian
P12E.	sex by age for only race native Hawaiian and other Pacific Islander
P12F.	sex by age for only other races
P12G.	sex by age for only two or more races
P12H.	sex by age for only Hispanic/Latino
P12I.	sex by age for only race White and not Hispanic/Latino

Table 1: Block-level contingency tables that we use. The same contingency tables were also used by RAP [5].

Attribute name	Type	Domain
Block ID	Integer	A constant number for each block
Gender	Boolean	{Male, Female}
Age	Integer	{0, 1, …, 115}
Race	Integer	{0, 1, …, 63} [1]
Hispanic	Boolean	{Hispanic, Not-Hispanic}

[1] Each number represents one race category defined by the U.S. Office of Management and Budget Standards. [19]

Table 2: Description of attributes in 2010 Census Decennial Microdata.

don't use D_{aux} when we relax the assumption of auxiliary dataset access in Section 5.2.

American Community Survey We conduct secondary experiments on the American Community Survey (ACS) dataset [17][4]. The ACS dataset is partitioned by Public Use Microdata Areas (PUMAs), with a PUMA being the smallest region to contain record-level data. We choose similar non-sensitive attributes of the ACS dataset as those of the PPMF dataset, in particular PUMA-id, age, sex, and race. We take the attribute describing whether the income level of a person is over $50000 as the sensitive attribute. Analogously to PPMF, we take the three smallest PUMAs with more than 4540 users, we split each dataset 90 – 10 and randomize the values of the sensitive attribute.

In line with prior work [5], we consider $k-way$ marginals as aggregate statistics on the ACS dataset. In particular, we consider all one-way and two-way marginals, such that we bucketize the age attribute into 5-year intervals.

4.2 Evaluation Metrics

Area Under Receiver Operating Curve (AUC). We use the area under the receiver operating curve, denoted as AUC, to measure the strength of the attacks on average across all target users.

True positive rate (TPR) at low false positive rate (FPR). As privacy is not an average-case metric [10], we measure–in line with the literature–the attacks' success to reliably infer the sensitive value of the most vulnerable users using True Positive Rate (TPR) at k False Positive Rate (FPR), where k is small. We denote this metric as TPR@kFPR.

Execution time. We compare the execution time of the methods using Python's *time* library on a Linux Virtual machine with an Intel Xeon Gold 6248 CPU and an NVIDIA Tesla V100S (32GB) GPU under Ubuntu 24.04.2 LTS. While RAP uses a GPU to train the synthetic data generator, DeSIA and CIP run exclusively on the CPU, each using 20 cores.

4.3 Attack Parameters

Parameters specific to state-of-the-art methods. For both CIP and RAP, we obtain $K = 100$ tentative reconstructed datasets, $D'_1, \ldots, D'_K$, each of size s, as recommended by the authors of RAP [5]. For RAP, we generate each of the tentative reconstructed datasets with their proposed synthetic data generator model with an internal dimension of 1000, and we allow each generator model to train over 1000 iterations.

Parameters specific to DeSIA. We instantiate the number of shadow datasets in the stochastic module $\mathcal{M}_S$ to be $N = 20000$, $\frac{2}{3}$ of them are used as training shadow datasets, and $\frac{1}{3}$ as validation shadow datasets. We evaluate various numbers of shadow datasets in Appendix J. We build the meta-classifier with logistic regression under L2 penalty. To reduce overfitting, we apply cross-validation combined with grid search to choose the best L2 penalty coefficient for the logistic regression. We allow the meta-classifier to train for maximally 1000 iterations.

5 Results

5.1 Attack Performance

Figure 4 (main) shows DeSIA to outperform all reconstruction-based methods at the attribute inference task, reaching an AUC of 0.75. RAP-init is the second-best method with an AUC of 0.64. Interestingly, while initialization using the auxiliary dataset D_{aux} helps the RAP method, lifting the AUC from 0.59 to 0.64, it has no noticeable impact on CIP; both CIP-rand and CIP-init have an AUC of 0.63. Our results show that releasing even a limited number of aggregates leaks sensitive information for the target users, a risk that existing state-of-the-art attacks substantially underestimate. To confirm the robustness of these results, we perform a one-sided two-sample t-test comparing DeSIA's performance to each of the reconstruction-based methods (see Appendix A for details). The tests indicate that DeSIA achieves a statistically significant improvement over the baselines, with $p < 0.05$.

In line with the literature [10], we also compare the methods' performance focusing on the most vulnerable users. Figure 4 (inset) shows DeSIA to strongly outperform all reconstruction-based methods here too, achieving a TPR@10^{-3}FPR of 0.14 while all other methods struggle to reliably predict the sensitive attribute of the most vulnerable users. Overall, DeSIA successfully identifies highly vulnerable users even when only a small number of aggregates are released, identifying risks that state-of-the-art attacks miss.

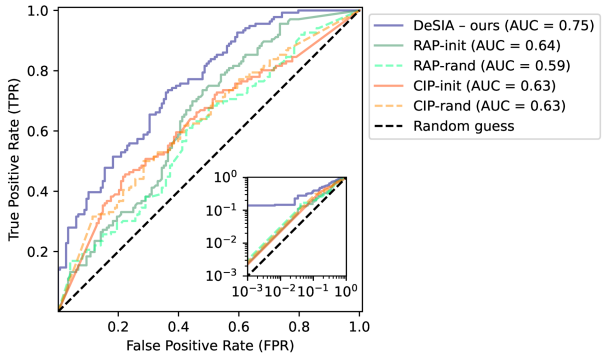

Figure 4: Performance on the PPMF dataset of our attack and the baseline reconstruction-based attacks at predicting the value of the sensitive attribute for all target users. The inset plot shares the same axes as the main plot.

[4] In line with Dick et al. [5], we use the folktable Python package [18] to access the 1-year ACS dataset.

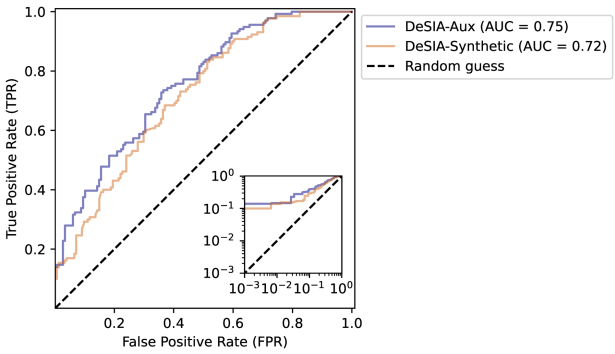

Figure 5: Performance of our attack instantiated with real-world auxiliary datasets and synthetic auxiliary datasets. The inset plot shares the same axes as the main plot.

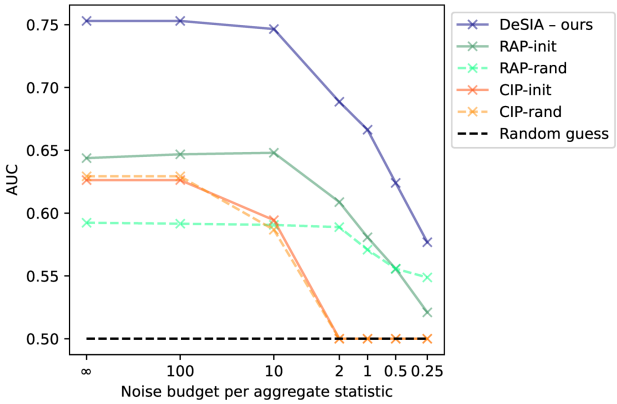

Figure 6: Robustness of the attacks to noise added to released aggregates. We report the mean AUC across 3 independent noisy aggregate statistic releases. The evaluation is performed on the PPMF dataset.

5.2 Relaxing the Assumption of Access to Auxiliary Dataset

While commonly accepted as a realistic assumption in the literature [5–8, 10–15], a real-world auxiliary dataset might not always be available for the attacker. The RAP attack already proposes a variant leveraging an auxiliary dataset, whereas CIP, as originally proposed [4], does not. Here, for a fair comparison, we relax that assumption by generating a synthetic dataset only using the released aggregates. In particular, we replace the auxiliary dataset with synthetic data generated only from the released aggregates, using the same generator [20] as RAP-rand [5].

Figure 5 shows DeSIA to have a minimal degradation in its performance when replacing the auxiliary dataset with a synthetically generated one: the AUC decreases from 0.75 to 0.72, and the TPR@10^{-3}FPR from 0.14 to 0.10. Statistical analysis (see Appendix A) shows no significant difference between DeSIA with a real-world auxiliary dataset and DeSIA-Synthetic ($p > 0.05$). Crucially, DeSIA-Synthetic achieves a higher AUC than the other state-of-the-art methods, including those with auxiliary data access. This result indicates that DeSIA, without requiring access to a real-world auxiliary dataset, can be instantiated to effectively evaluate the attribute inference risk against limited aggregate statistics.

5.3 Robustness to Laplace Noise Addition

In line with the literature [4, 5], we have assumed that exact values of the aggregate statistics are released. Here, we perturb the released aggregate statistics by adding different levels of Laplace noise using a simple per-aggregate privacy budget. Given a per-aggregate budget of ε, we perturb the aggregates by adding Laplace noise $n_i \sim Laplace(\frac{1}{\varepsilon})$ to each aggregate statistic q_i. For simplicity and in line with U.S. Census Bureau's approach [16], we round the answer to the nearest integer $round(q_i(D) + n_i), \forall i \in \{1, \ldots, m\}$.

Figure 6 shows DeSIA to outperform all other methods and shows both DeSIA and RAP to be robust to even a significant level of noise addition; albeit at the cost of an expected drop in the performance of both methods. CIP handles noise deterministically: for some users, the added noise does not create conflicts and inference succeeds, while for others, conflicts arise and CIP fails. As a result, CIP can handle some level of noise, with performance varying across users, but it struggles to perform better than the random guess baseline when faced with larger levels of noise. One-sided t-tests at each noise level confirm that DeSIA significantly outperforms the baselines ($p < 0.05$; see Appendix A). Figure 16 in Appendix F additionally shows that DeSIA is the only method achieving non-zero TPR@10^{-3}FPR, which gradually decreases with higher noise levels.

5.4 Impact of the Number of Aggregates

We have so far evaluated the methods assuming that there are four times more users than the aggregate statistics in the protected dataset ($\frac{m}{s} = 0.25$). We here test the performance of DeSIA and reconstruction-based methods when different numbers of aggregate statistics are released. We sample different numbers of aggregate statistics at uniformly random from the same set of block-level contingency tables (see Table 1) with different ratios of $\frac{m}{s}$. We evaluate the aggregate statistics on the same protected dataset D.

Figure 7 shows DeSIA to outperform all reconstruction-based methods across all numbers of aggregate statistics, with t-tests for each achieving small p-values, $p < 0.05$ (see Appendix A) for all methods and aggregate releases, except for $\frac{m}{s} = 2^{-4}$ when comparing DeSIA with CIP-rand. As expected, increasing the number of released aggregates increases the risk on average and lifts the performance of all methods. The state-of-the-art methods perform similarly among all sizes of aggregate statistics, with RAP being slightly better than CIP in general. Similar conclusions can be drawn for TPR@10^{-3}FPR, with DeSIA being the only method with positive values across different number of aggregate releases (Figure 17 in Appendix G).

We further investigate the marginal contribution of releasing complete contingency tables (a) in isolation and (b) in combination with other related tables. As shown in Appendix C, releasing a

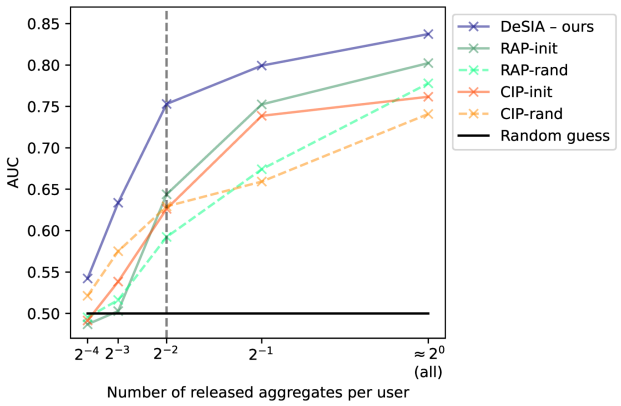

Figure 7: Impact of the number of released aggregates on the performance on the PPMF dataset of our attack and the baseline reconstruction-based attacks. The vertical dotted line indicates the default value taken for the other experiments.

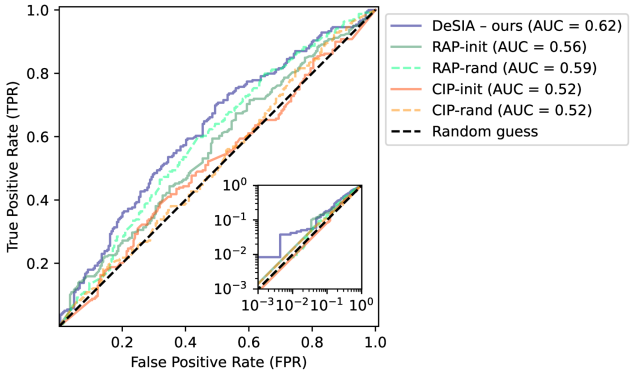

Figure 8: Performance on the ACS dataset of our attack and the baseline reconstruction-based attacks. The inset plot shares the same axes as the main plot.

single table in isolation already produces non-zero privacy leakage, though at a substantially lower level than the full release, and each additional related table further increases the inferred privacy risk, as reflected by higher AUC values.

5.5 Ablation

To better understand the contribution of each element of DeSIA, we perform an ablation study by systematically removing individual elements one by one and evaluating the impact on the method's performance. DeSIA has four main elements, three in the deterministic module $\mathcal{M}_D$ and one in the stochastic module $\mathcal{M}_S$. The three main elements of the deterministic module are: (1) finding a feasible value using either a solver or synthetic data generated by RAP, (2) adding the uniqueness constraint c_t specific to the target user, and (3) verifying the uniqueness of the found feasible value. The inclusion of the stochastic module $\mathcal{M}_S$ is the fourth element.

Table 3 shows that all four elements influence DeSIA's performance and are necessary for the method to perform well. In particular, the removal of the stochastic module, replacing it with a random guess, and the removal of the verification of the feasible value substantially hurt the performance on AUC and TPR@kFPR (with $p < 0.05$; see Appendix A). Our results emphasize the reliance of DeSIA on both the deterministic and the stochastic modules.

5.6 Attack Performance on the ACS Dataset

We have so far empirically evaluated DeSIA and state-of-the-art methods on the PPMF dataset. Here, we show their performance on another dataset released by the U.S. Census Bureau, the ACS dataset. In line with prior work [5], we evaluate attacks on k-way marginal statistics. These are known to be a harder problem as the queries are less informative on average than the aggregate statistics from contingency tables carefully chosen by the U.S. Census Bureau (Table 1) on the PPMF dataset.

Figure 8 shows DeSIA to generalize well to the ACS dataset and outperforms all other methods both on overall AUC (main) and on TPR at low FPR (inset). As with the PPMF dataset, we confirm these results with one-sided two-sample t-tests yielding $p < 0.05$ (Appendix A). Because of the less informative aggregate statistics compared to those on the PPMF dataset, the performance of all methods on ACS is lower than the performance on PPMF. All the methods, however, exhibit the same overall trend on the ACS dataset as on the PPMF dataset when varying both the level of Laplace noise (Appendix H) and the number of released aggregates (Appendix I).

6 Extension to Membership Inference Attack

We have so far instantiated DeSIA on attribute inference attacks (AIAs), as we believe they are a particular concern for aggregate data release. Membership inference attacks (MIAs) have, however, been extensively used to measure information leakage in the machine learning and synthetic data context and, in some cases, can be a concern in practice [21].

We here extend DeSIA to membership inference attacks (MIAs). We describe how we modify the privacy game to MIA, then extend DeSIA to MIAs, and compare it to reconstruct-based methods.

6.1 Framework for Membership Inference Attack as a Privacy Game

We here modify the attribute inference attack privacy game to the membership inference case, randomizing the membership of the target users in the protected dataset instead of the sensitive attribute values. The game is as follows (Figure 9).

First, the defender samples a secret bit b uniformly at random from $\{0, 1\}$ to determine the membership of the target user u^*.

Second, the defender removes the target user from the dataset D, if $b = 0$, or removes another target user u' from D, if $b = 1$.

Third, the defender obtains the fixed aggregate statistics $Q(D)$ by evaluating the aggregate statistics Q on the protected dataset D. The defender sends the values $Q(D)$ and the size of the protected dataset $|D|$ to the attacker.

Fourth, the attacker aims to infer the membership of the target user. The attacker runs an attack $\mathcal{A}$ and obtains a membership prediction b'.

Method				AUC	TPR@kFPR	
Approach to find a feasible value	**Uniqueness constraint**	**Verification of feasible value**	**Stochastic module**		**k=10%**	**k=1%**
solver	yes	yes	yes	0.75	0.38	0.15
solver	yes	yes	no	0.63	0.09	0.09
solver	yes	no	no	0.59	0.0	0.0
synthetic data	yes	no	no	0.56	0.0	0.0
synthetic data	no	no	no	0.47	0.0	0.0

Table 3: Ablation study of the impact of different elements of our attack. The evaluation is performed on the PPMF dataset.

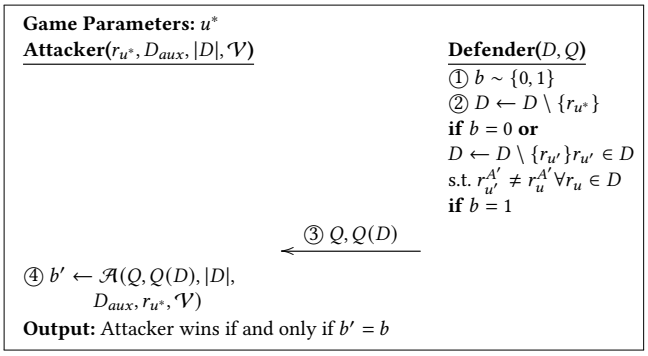

Figure 9: Privacy game for membership inference attacks against fixed aggregate statistics.

The attacker wins the game if and only if the predicted membership bit b' is equal to the secret bit b held by the defender, $b' = b$.

6.2 Extension of DeSIA to MIA

We adapt DeSIA to MIA with small modifications on both the deterministic $\mathcal{M}_D$ and stochastic $\mathcal{M}_S$ modules. Note that, for the deterministic module $\mathcal{M}_D$, the uniqueness constraint c_t is not applicable anymore, since the goal of $\mathcal{M}_D$ is to decide the deterministic membership. Furthermore, we verify the membership of the target user by finding a feasible multiplicity of target record $(x_{(r^1_{u^*},\dots,r^n_{u^*})}$ and modifying the null constraint c'_t to verify it:

$$c_{t'} : x_{(r^1_{u^*},\dots,r^n_{u^*})} = 1 - \sigma(x_{(r^1_{u^*},\dots,r^n_{u^*})}).$$

For the stochastic module $\mathcal{M}_S$, we slightly change the generation of the shadow datasets to resemble the first two steps of the privacy game. Before creating each shadow dataset D^i_{shadow}, we now sample a variable b^i uniformly at random from $\{0,1\}$. If $b^i = 1$, we consider the target user to be a member of D^i_{shadow} and sample $s-1$ users from the auxiliary dataset D_{aux}. If $b^i = 0$, we consider the target record to not be a member of D^i_{shadow} and sample s users from D_{aux}. We train the machine learning model M on the aggregate statistics from these shadow datasets $\{(Q(D^1_{shadow}, b^1), \dots, (Q(D^N_{shadow}, b^N)\}$ to predict the membership of target user $\hat{b^i}$ given the input $Q(D)$.

6.3 Adapting Reconstruction-Based Attacks to the MIA Task

Both reconstruction methods provide us with a number of tentative reconstructed datasets $D'_1, \dots, D'_K$. We adapt the AIA procedure (Section 2.5) to now perform a majority vote only from the 0-neighborhood of the target record r_{u^*} across all tentative reconstructed datasets. If the target user appears in more than $\frac{K}{2}$ tentative reconstructed datasets, we consider $\hat{b} = 1$ and $\hat{b} = 0$ otherwise.

6.4 Results

Figure 10 shows that DeSIA can be adapted to the MIA task, performing well both overall (AUC=0.85) and on the most vulnerable target record TPR@10^{-3}FPR of 0.10. The RAP method performs substantially worse than DeSIA, while a straightforward adaptation of CIP does not perform better than a random guess. One-sided t-tests confirm that DeSIA's improvement over the reconstruction-based methods is statistically significant ($p < 0.05$, see Appendix A). These results show that DeSIA effectively evaluates the membership inference risk of all target users and identifies those at high risk using only a small number of aggregate statistics.

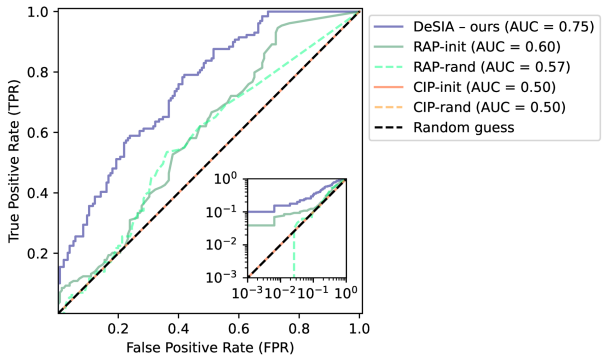

Figure 10: Membership inference attack performance on the PPMF dataset. The inset plot shares the same axes as the main plot. Interestingly, RAP-rand performs worse at TPR at low FPR on MIA than the random guess baseline. This is due to RAP incorrectly infers with high-confidence membership of users when their values for the non-sensitive attributes are relatively common on their own.

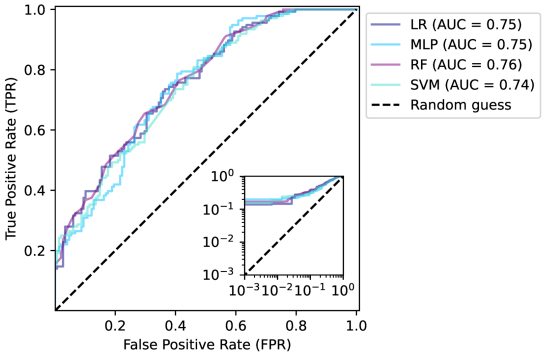

Figure 11: Performance on the PPMF dataset of our attack where the stochastic module $\mathcal{M}_S$ uses different machine learning models. The inset plot shares the same axes as the main plot.

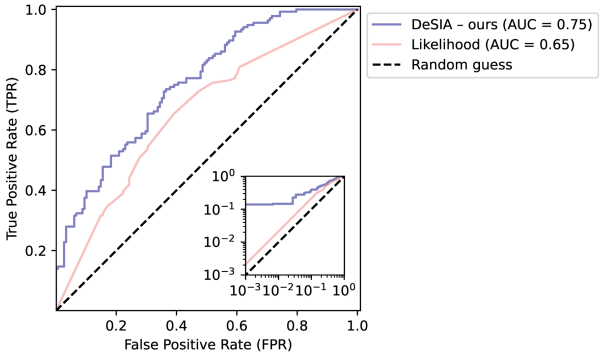

Figure 12: Performance on the PPMF dataset of our attack where the stochastic module $\mathcal{M}_S$ uses our standard machine learning model (Logistic Regression) and a likelihood attack. The inset plot shares the same axes as the main plot.

7 Discussion

7.1 Using a Different Machine Learning Model

To optimize for speed, the stochastic module $\mathcal{M}_S$ uses a logistic regression (LR) as a machine learning (ML) model to infer the sensitive attribute of the target user using aggregate statistics and its values. We here explore alternative options.

We evaluate three alternative choices of ML models:

- **Multi-Layered Perceptron (MLP)**: an MLP with two hidden layers of sizes 50 and 20.
- **Random Forest (RF)**: a random forest model with 100 trees. We set Gini impurity as the criterion measuring split quality and select a random subset of features at each split, where the subset contains square root of the total number of features.
- **Support Vector Machine (SVM)**: an SVM with radial-basis function kernels under hinge loss.

Figure 11 shows that the choice of ML model for the stochastic module does not have a substantial impact on the attack performance, with all four ML models achieving an AUC of around 0.75. One-sided two-sample t-tests confirm that differences across models are not statistically significant ($p > 0.05$), as expected.

7.2 Comparison to Likelihood Attack

We have thus far only considered using an ML model in the stochastic module $\mathcal{M}_S$. We here explore whether using an ML model is necessary or if it can be replaced with a simple likelihood attack.

We consider a simple likelihood attack to infer the value of the sensitive attribute for the target user by (a) inferring the most likely value of the sensitive attribute given each aggregate statistic independently and (b) performing a majority vote across aggregates to obtain the final inference.

To estimate the likelihood, we use the shadow datasets D^1_{shadow}, $\ldots$, D^N_{shadow} sampled from the auxiliary dataset D_{aux}. We group the datasets into $|\mathcal{V}_n|$ groups, where each group G_b contains shadow datasets with the same sensitive attribute value b of the target user, $G_b = \{D^i_{shadow} | z_{i,j} = b\}$. For an aggregate statistic q, we calculate the mean $\mu_{q,b}$ and standard deviation $\sigma_{q,b}$ of the values on the shadow datasets in each group G_b, $\{q(D^i_{shadow}) z_{i,j} = b\}$. Given a value $q(D)$ on the private dataset D, we estimate the likelihood of the sensitive value of the target user by assuming a Gaussian distribution of query answers and calculating:

$$\Pr[r^n_{u^*} = b] = \Phi\left(\frac{q(D) - \mu_{G_b}}{\sigma_{G_b}}\right),$$

where Φ is the probability density function of the standard normal distribution.

For an aggregate statistic q, we infer $r^n_{u^*}$ as the value b from the group with maximum likelihood. We obtain independent votes from all aggregates $q \in Q$ on which the membership of the target user u^* in their userset U_q, $u^* \in U_q$, depends only on the target user's sensitive value r^n_u, i.e., aggregates that condition on the sensitive attribute and do not exclude the target user's values in the conditions of the non-sensitive attributes, $Q_{u^*} = \{q \,|r^i_{u^*} \in V^q_i, \forall i \in \{1, \ldots, n-1\}$ and $V^q_n \neq \mathcal{V}_n\}$. We perform a majority vote on the predictions of all aggregates in Q_{u^*} to get the prediction of $r^n_{u^*}$.

Figure 12 shows that using an ML model is necessary and leads to a substantially better performing attack than the likelihood attack on both overall (AUC - main) and on highly vulnerable users (TPR@kFPR - inset). One-sided two-sample t-tests confirm that the ML-based stochastic module significantly outperforms the likelihood attack ($p < 0.05$), demonstrating the necessity of the ML model used in stochastic module $\mathcal{M}_S$.

7.3 Time Complexity Analysis

We analyze the time complexity of DeSIA in terms of asymptotic time complexity and empirical execution time measurements.

Asymptotic complexity. We here analyze the time complexity of DeSIA by analyzing the complexity of DeSIA's two modules: the deterministic $\mathcal{M}_D$ and the stochastic $\mathcal{M}_S$ module.

The time complexity of the deterministic module $\mathcal{M}_D$ is determined by the complexity of the underlying solver's branch-and-bound method [22]. Its worst-case time complexity is $O(V^{2.5}b^V)$ [23, 24], where V denotes the number of unknown variables and b

Method	AUC ↑	Execution time (s) ↓
DeSIA	0.75	
Deterministic $\mathcal{M}_D$		4.80 ± 0.47
Stochastic $\mathcal{M}_S$		181.61 ± 18.89
RAP-init	0.64	678.04 ± 6.41
RAP-rand	0.59	673.82 ± 8.99
CIP-init	0.63	97.48 ± 6.92
CIP-rand	0.63	95.63 ± 5.69

Table 4: Execution time for attacking a target user. We show the mean ± standard deviation of the execution time across all target users in the 10 blocks of the PPMF dataset.

denotes the branching factor. In the case of $\mathcal{M}_D$, there are $V = \prod_{i=1}^{n} |\mathcal{V}_i|$ variables and, assuming we branch one unknown variable at a time, the branching factor is the number of possible values for the variable, $b = \max_{i=1,\dots,n} \mathcal{V}_i$. Substituting these values, the worst-case time complexity of $\mathcal{M}_D$ to be $O(\prod_{i=1}^{n} |\mathcal{V}_i|^{2.5} \cdot (\max_{i=1,\dots,n} \mathcal{V}_i)^{\prod_{i=1}^{n} |\mathcal{V}_i|})$.

The time complexity of the stochastic module $\mathcal{M}_S$ is determined by the time complexity to (a) generate the N shadow datasets, $D^1_{shadow}, \dots, D^N_{shadow}$, $O(N \cdot n)$, (b) evaluate the m aggregate statistics Q on the shadow datasets, $O(N \cdot n \cdot m)$, and (c) train the machine learning model to infer $r^n_{u^*}$, $O(N \cdot m)$. The time complexity of $\mathcal{M}_S$ is thus $O(N \cdot n \cdot m)$.

Empirical Execution Time. Table 4 shows the execution time for attacking a target user for DeSIA and the reconstruction-based methods. DeSIA strikes the best balance between runtime and accuracy: it completes in 186 seconds on average, including 4.80 seconds for the deterministic module $\mathcal{M}_D$ and 181.61 seconds for the stochastic module $\mathcal{M}_S$, while achieving the highest AUC of 0.75. Although CIP requires only about 95 and 97 seconds, for CIP-rand and CIP-init respectively, it achieves substantially lower performance than DeSIA. RAP, on the other hand, is the slowest method due to GPU training, taking roughly 674 and 678 seconds for RAP-rand and RAP-init and still achieves worse AUC performance than DeSIA. Overall, the results highlight that DeSIA provides the most favorable runtime-performance trade-off among all methods.

8 Related Work

Empirical attacks have been proposed against a range of non-interactive data release mechanisms: fixed aggregate statistics, machine learning models, and synthetic data, as well as against interactive release mechanisms such as query-based systems.

8.1 Attacks Against Fixed Aggregates

8.1.1 Reconstruction Attacks. Dinur et al. [25] proposed the first linear reconstruction attack which aims to reconstruct the protected dataset by solving a linear system of equations. Dwork et al. [26] extended their attack by relaxing the assumption that the noise added to the aggregates is bounded and introduced linear programming with error correction codes to reconstruction attacks. Following works improved the linear reconstruction attack by decreasing the requirement on the number of released aggregates [27] and improved the reconstruction of categorical attributes [28]. These attacks, however, suffer from the so-called row-naming problem, which assumes that the users in the protected tabular dataset have unique identifiers that are both (a) known to the attacker and (b) used in the released aggregate statistics. This is a strong assumption that we do not make here. Instead, two new attacks remove this strong assumption: the U.S. Census Bureau introduced a constraint programming-based reconstruction attack [4] and Dick et al. proposed a synthetic data-based reconstruction attack [5] by solving a non-convex optimization problem [29]. We compare our work to these two attacks.

8.1.2 Inference Attacks. The (limited) literature on inference attacks against aggregate statistics predates shadow modeling-based methods and often uses strong and theoretical assumptions.

Homer et al. [30] introduced the first membership inference attack (MIA) against a large number of fixed aggregate statistics from genomic data. The attack has been improved [31] and extended to microRNA data [32]. Wang et al. [33] extended the attack to an attribute inference attack (AIA). All three attacks are specific to genomic data, and to the best of our knowledge, do not have an obvious extension to tabular data, which is the focus of our paper.

Kasiviswanathan et al. [34, 35] introduced an AIA against aggregate statistics for tabular data by (a) assuming the attacker to have access to the values of the non-sensitive attributes for all users in the protected dataset D, and (b) instantiating the reconstruction attack by Dinur et al. [25] to reconstruct the sensitive attribute. While the strong assumption of having access to all non-sensitive attributes for all users in the protected dataset allows the attacker to alleviate the row-naming problem, it is a very strong assumption that we do not make as we consider it to rarely hold in practice.

8.2 Attacks Against Machine Learning Models

An active field of research exists on empirical inference attacks against machine learning models. Attacks typically rely on the loss of the target model [36] and shadow models [11–13].

The first inference attacks against machine learning models were attribute inference attacks (also referred to in the literature as model inversion attacks [12]). Fredrikson et al. [37] proposed the first methodology and AIA against an ML model, inferring a patient's genetic marker from a model trained to predict the dosage of a drug. The attack was later extended from a black-box to a white-box attack by Mehnaz et al. [38]. The correlation between the genetic marker and the dosage of the drug has, however, raised concerns about the validity of some of the results and the extent to which it proved that an information leakage was happening [39]. Jayaraman et al. [40] formalized the issue and showed that popular black-box attacks often do not perform better than a baseline that uses these correlations to impute the values of the sensitive attribute. We refer to this as the imputation issue. Recent work on AIA against ML models, often image models, addressed this issue by evaluating the attack performance inferring the sensitive attributes with the model and without the model [41]. In line with AIAs against tabular datasets protected by query-based systems (below) we address this issue by uniformly randomizing the sensitive attributes.

Soon after the first AIA was proposed, Shokri et al. proposed the first MIA against machine learning models [11] and coined the term shadow models. This has become an active field of research with

attacks such as RMIA [42] and LiRA [10] and a focus on evaluating the performance of inference attacks on the most vulnerable users by reporting TPR at low FPR [10] which we follow.

Empirical attacks are often used to argue that models are not privacy-preserving by default, emphasizing the need for privacy-preserving measures, and to estimate the privacy risk in practice by models trained with DP privacy guarantees, typically using DP-SGD [43]. DP-SGD theoretical guarantees are indeed typically not tight, often overestimating the strength of the attacker [1, 44, 45]. The common approach when releasing machine learning models is thus to combine DP guarantees [1] with empirical attacks.

8.3 Attacks Against Synthetic Tabular Data

Contrary to machine learning models, synthetic data generators lack a notion of a loss for a given user. A separate line of research has developed on attacks against released synthetic data. More recently, this line of work has also targeted synthetic data generators themselves, including diffusion-based models. Attacks against synthetic data typically rely on the same shadow modeling technique and use the distance aggregate statistics as a signal, e.g., for membership. Stadler et al. [14] proposed the first MIA against synthetic data. Houssiau et al. [15] and then Meeus et al. [46] successively extended the attack by expanding the set of aggregate statistics used. Most recently, MIAs targeting directly diffusion synthetic data generation models were proposed [47]. An AIA against synthetic data was introduced by Annamalai et al. [48] by using a linear reconstruction attack. The AIA by Annamalai et al. [48], however, assumes the attacker to have access not only to the released synthetic dataset, but also, similarly to Kasiviswanathan et al. [34, 35], access to all non-sensitive attributes of all users in the original private dataset, which (as discussed in Section 8.1.2 above) is a much stronger assumption than the ones we consider.

8.4 Attacks Against Interactive Systems

Query-based systems (QBSes) are interactive interfaces that allow analysts to send queries and retrieve aggregate answers about a protected dataset. Two broad types of QBSes have been implemented in practice: interfaces, such as TableBuilder [49], a special-purpose QBS for national census data by the Australian Bureau of Statistics, and general-purpose QBSes, such as Diffix [50]. QBSes offer a much larger attack surface as, contrary to non-interactive data release mechanisms, they allow the attacker to select the queries.

Cohen et al. [51] and Joseph et al. [52] both proposed reconstruction attacks against Diffix [50], both of which are based on the earlier work of Dinur et al. [25]. Pyrgelis et al. [53] proposed an MIA against a query-based system protecting spatio-temporal data, based on their earlier work [54]. Chipperfield et al. [55] and Rinott et al. [56] proposed AIAs against TableBuilder, exploiting the seeded bounded noise. Gadotti et al. [8] proposed an AIA exploiting Diffix's noise structure. Cretu et al. [6] proposed an automated approach to discovering AIAs against query-based systems showing it to outperform the best known manual attacks. Stevanoski et al. [7] then further improved the speed of the attack and the space of sets of queries that can be searched.

9 Limitation

Our framework estimates the inference risk posed by the released aggregates, which is one of the three components driving the inference risk "in the wild" (see Section 2.3). While our framework provides a principled proxy for estimating the overall "in the wild" inference risk, it does not provide monotonicity guarantees for it in general: if one release exhibits higher inference risk than another under our framework, this does not guarantee that the same ordering will hold for the overall "in the wild" risk against all attackers with different goals, background knowledge, or capabilities. This limitation is not unique to our framework. Prior work, based on quantitative information flow which studies information leakage using information-theoretic techniques, has formalized such comparisons through the notion of leakage order [57] and shown that differential privacy mechanisms from different families, even when ordered by their privacy budget parameter ε, do not preserve a monotonic ordering of risk across all possible attackers [58].

10 Conclusion

We propose a framework for inference attacks against fixed aggregate statistics. We introduce DeSIA, a deterministic stochastic attribute inference attack against fixed aggregate statistics.

We first show DeSIA to be highly effective at evaluating the attribute inference risk. Using DeSIA, we show that even a limited number of aggregate statistics reveal sufficient information to enable attribute inference against highly vulnerable users, with DeSIA achieving a $TPR@10^{-3}FPR$ of 0.14. It also outperforms all other methods, indicating that existing approaches substantially underestimate the risk.

Second, we show DeSIA to perform well without access to a real-world auxiliary dataset, outperforming all reconstruction-based attacks, including those that assume auxiliary dataset access. Replacing the auxiliary dataset with synthetically generated data using only the released aggregates, DeSIA again effectively evaluates the risk posed by weak attackers and identifies highly vulnerable users.

Third, we evaluate DeSIA's robustness to varying numbers of released aggregates and levels of noise addition. In both cases, DeSIA has a better performance than reconstruction-based methods.

Fourth, we perform an ablation study of the different components of our method, and show all components to be important to the overall performance. In particular, we show the stochastic module to be important for both the overall AUC and TPR at low FPR, and the verification by the deterministic module to be important for identifying highly vulnerable users with only one possible value for their sensitive attribute.

Fifth, we extend DeSIA to membership inference attack and show it to also substantially outperform all other methods both on overall AUC and on identifying highly vulnerable users.

Our results show how even a limited number of aggregate statistics can expose users in the underlying dataset to risk from inference attacks, with some users being highly vulnerable. Taken together, they strongly emphasize the need to implement formal privacy mechanisms and to empirically test the privacy of the data release.

Acknowledgments

This work has been partially supported by the CHEDDAR: Communications Hub for Empowering Distributed ClouD Computing Applications and Research funded by the UK EPSRC under grant numbers EP/Y037421/1 and EP/X040518/1 and by the Agence E-Santé Luxembourg [5]. We acknowledge computational resources provided by the Imperial College Research Computing Service. [6]

We thank the editor and the reviewers for their insightful questions and valuable feedback, which helped improve the paper.

We used AI-based tools to polish writing. This help was on a level of a spelling and grammar checker.

References

[1] N. Ponomareva, H. Hazimeh, A. Kurakin, Z. Xu, C. Denison, H. B. McMahan, S. Vassilvitskii, S. Chien, and A. G. Thakurta, "How to dp-fy ml: A practical guide to machine learning with differential privacy," *Journal of Artificial Intelligence Research*, vol. 77, pp. 1113–1201, 2023.

[2] M. S. M. S. Annamalai, G. Ganev, and E. De Cristofaro, ""what do you want from theory alone?" experimenting with tight auditing of differentially private synthetic data generation," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 4855–4871.

[3] "Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation) (text with eea relevance)," pp. 1–88, May 2016.

[4] J. M. Abowd, T. Adams, R. Ashmead, D. Darais, S. Dey, S. L. Garfinkel, N. Goldschlag, D. Kifer, P. Leclerc, E. Lew *et al.*, "The 2010 census confidentiality protections failed, here's how and why," National Bureau of Economic Research, Tech. Rep., 2023.

[5] T. Dick, C. Dwork, M. Kearns, T. Liu, A. Roth, G. Vietri, and Z. S. Wu, "Confidence-ranked reconstruction of census microdata from published statistics," *Proceedings of the National Academy of Sciences*, vol. 120, no. 8, p. e2218605120, 2023.

[6] A.-M. Cretu, F. Houssiau, A. Cully, and Y.-A. de Montjoye, "Querysnout: Automating the discovery of attribute inference attacks against query-based systems," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 623–637.

[7] B. Stevanoski, A.-M. Cretu, and Y.-A. de Montjoye, "Querycheetah: Fast automated discovery of attribute inference attacks against query-based systems," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 3451–3465.

[8] A. Gadotti, F. Houssiau, L. Rocher, B. Livshits, and Y.-A. De Montjoye, "When the signal is in the noise: Exploiting diffix's sticky noise," in *28th USENIX Security Symposium (USENIX Security 19)*, 2019, pp. 1081–1098.

[9] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2024. [Online]. Available: https://www.gurobi.com

[10] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer, "Membership inference attacks from first principles," in *2022 IEEE symposium on security and privacy (SP)*. IEEE, 2022, pp. 1897–1914.

[11] 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.

[12] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models," *arXiv preprint arXiv:1806.01246*, 2018.

[13] C. A. Choquette-Choo, F. Tramer, N. Carlini, and N. Papernot, "Label-only membership inference attacks," in *International conference on machine learning*. PMLR, 2021, pp. 1964–1974.

[14] T. Stadler, B. Oprisanu, and C. Troncoso, "Synthetic data – anonymisation groundhog day," in *31st USENIX Security Symposium (USENIX Security 22)*. Boston, MA: USENIX Association, Aug. 2022, pp. 1451–1468. [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/stadler

[15] F. Houssiau, J. Jordon, S. N. Cohen, O. Daniel, A. Elliott, J. Geddes, C. Mole, C. Rangel-Smith, and L. Szpruch, "Tapas: a toolbox for adversarial privacy auditing of synthetic data," *arXiv preprint arXiv:2211.06550*, 2022.

[16] U.S. Census Bureau, "Developing the das: Demonstration data and progress metrics," https://www.census.gov/programs-surveys/decennial-census/decade/2020/planning-management/process/disclosure-avoidance/2020-das-development.html, 2020.

[17] ——, "American Community Survey (ACS)," 2020, aggregated survey data used for trend analysis and public reporting. [Online]. Available: https://www.census.gov/programs-surveys/acs

[18] F. Ding, M. Hardt, J. Miller, and L. Schmidt, "Retiring adult: New datasets for fair machine learning," 2022. [Online]. Available: https://arxiv.org/abs/2108.04884

[19] Office of Management and Budget, "Revisions to the standards for the classification of federal data on race and ethnicity," 1997. [Online]. Available: https://obamawhitehouse.archives.gov/omb/fedreg_1997standards

[20] S. Aydore, W. Brown, M. Kearns, K. Kenthapadi, L. Melis, A. Roth, and A. A. Siva, "Differentially private query release through adaptive projection," in *International Conference on Machine Learning*. PMLR, 2021, pp. 457–467.

[21] M. Boorstein, M. Iati, and A. Shin, "Top U.S. Catholic Church official resigns after cellphone data used to track him on Grindr and to gay bars," 2021. [Online]. Available: https://www.washingtonpost.com/religion/2021/07/20/bishop-misconduct-resign-burrill/

[22] Gurobi Optimization, LLC, "Gurobi Mixed Integer Program Basics," 2024. [Online]. Available: https://www.gurobi.com/resource/mip-basics/

[23] D. R. Morrison, S. H. Jacobson, J. J. Sauppe, and E. C. Sewell, "Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning," *Discrete Optimization*, vol. 19, pp. 79–102, 2016.

[24] P. M. Vaidya, "Speeding-up linear programming using fast matrix multiplication," in *30th annual symposium on foundations of computer science*. IEEE Computer Society, 1989, pp. 332–337.

[25] I. Dinur and K. Nissim, "Revealing information while preserving privacy," in *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2003, pp. 202–210.

[26] C. Dwork, F. McSherry, and K. Talwar, "The price of privacy and the limits of lp decoding," in *Proceedings of the thirty-ninth annual ACM Symposium on Theory of Computing*, 2007, pp. 85–94.

[27] C. Dwork and S. Yekhanin, "New efficient attacks on statistical disclosure control mechanisms," in *Advances in Cryptology–CRYPTO 2008: 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings 28*. Springer, 2008, pp. 469–480.

[28] K. Choromanski and T. Malkin, "The power of the dinur-nissim algorithm: breaking privacy of statistical and graph databases," in *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI symposium on Principles of Database Systems*, 2012, pp. 65–76.

[29] T. Liu, G. Vietri, and S. Z. Wu, "Iterative methods for private synthetic data: Unifying framework and new methods," *Advances in Neural Information Processing Systems*, vol. 34, pp. 690–702, 2021.

[30] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J. V. Pearson, D. A. Stephan, S. F. Nelson, and D. W. Craig, "Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays," *PLoS genetics*, vol. 4, no. 8, p. e1000167, 2008.

[31] K. B. Jacobs, M. Yeager, S. Wacholder, D. Craig, P. Kraft, D. J. Hunter, J. Paschal, T. A. Manolio, M. Tucker, R. N. Hoover *et al.*, "A new statistic and its power to infer membership in a genome-wide association study using genotype frequencies," *Nature genetics*, vol. 41, no. 11, pp. 1253–1257, 2009.

[32] M. Backes, P. Berrang, M. Humbert, and P. Manoharan, "Membership privacy in microrna-based studies," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 319–330. [Online]. Available: https://doi.org/10.1145/2976749.2978355

[33] R. Wang, Y. F. Li, X. Wang, H. Tang, and X. Zhou, "Learning your identity and disease from research papers: information leaks in genome wide association study," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, 2009, pp. 534–544.

[34] S. P. Kasiviswanathan, M. Rudelson, and A. Smith, "The power of linear reconstruction attacks," in *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 2013, pp. 1415–1433.

[35] S. P. Kasiviswanathan, M. Rudelson, A. Smith, and J. Ullman, "The price of privately releasing contingency tables and the spectra of random matrices with correlated rows," in *Proceedings of the forty-second ACM symposium on Theory of computing*, 2010, pp. 775–784.

[36] 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.

[37] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An {End-to-End} case study of personalized warfarin dosing," in *23rd USENIX security symposium (USENIX Security 14)*, 2014, pp. 17–32.

[38] S. Mehnaz, S. V. Dibbo, R. De Viti, E. Kabir, B. B. Brandenburg, S. Mangard, N. Li, E. Bertino, M. Backes, E. De Cristofaro *et al.*, "Are your sensitive attributes private? novel model inversion attribute inference attacks on classification models," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 4579–4596.

[39] F. McSherry, "Statistical inference considered harmful," https://github.com/frankmcsherry/blog/blob/master/posts/2016-06-14.md, 2016, [Accessed 13-04-2025].

[40] B. Jayaraman and D. Evans, "Are attribute inference attacks just imputation?" in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications*

[5] https://www.esante.lu/

[6] http://doi.org/10.14469/hpc/2232

Security, 2022, pp. 1569–1582.
[41] B. Z. H. Zhao, A. Agrawal, C. Coburn, H. J. Asghar, R. Bhaskar, M. A. Kaafar, D. Webb, and P. Dickinson, "On the (in) feasibility of attribute inference attacks on machine learning models," in *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2021, pp. 232–251.
[42] S. Zarifzadeh, P. Liu, and R. Shokri, "Low-cost high-power membership inference attacks," *arXiv preprint arXiv:2312.03262*, 2023.
[43] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 308–318.
[44] M. Nasr, S. Songi, A. Thakurta, N. Papernot, and N. Carlin, "Adversary instantiation: Lower bounds for differentially private machine learning," in *2021 IEEE Symposium on security and privacy (SP)*. IEEE, 2021, pp. 866–882.
[45] P. Stock, I. Shilov, I. Mironov, and A. Sablayrolles, "Defending against reconstruction attacks with r\'enyi differential privacy," *arXiv preprint arXiv:2202.07623*, 2022.
[46] M. Meeus, F. Guepin, A.-M. Creţu, and Y.-A. de Montjoye, "Achilles' heels: vulnerable record identification in synthetic data publishing," in *European Symposium on Research in Computer Security*. Springer, 2023, pp. 380–399.
[47] X. Wu, Y. Pang, T. Liu, and S. Wu, "Winning the midst challenge: New membership inference attacks on diffusion models for tabular data synthesis," *arXiv preprint arXiv:2503.12008*, 2025.
[48] M. S. M. S. Annamalai, A. Gadotti, and L. Rocher, "A linear reconstruction approach for attribute inference attacks against synthetic data," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 2351–2368.
[49] C. M. O'Keefe, S. Haslett, D. Steel, and R. Chambers, "Table builder problem-confidentiality for linked tables," 2008.
[50] P. Francis, S. Probst Eide, and R. Munz, "Diffix: High-utility database anonymization," in *Privacy Technologies and Policy: 5th Annual Privacy Forum, APF 2017, Vienna, Austria, June 7-8, 2017, Revised Selected Papers 5*. Springer, 2017, pp. 141–158.
[51] A. Cohen and K. Nissim, "Linear program reconstruction in practice," *arXiv preprint arXiv:1810.05692*, 2018.
[52] D. Privacy, "Reconstruction Attacks in Practice," https://differentialprivacy.org/diffix-attack/, [Accessed 01-02-2024].
[53] A. Pyrgelis, "On Location, Time, and Membership: Studying How Aggregate Location Data Can Harm Users' Privacy," https://www.benthamsgaze.org/2018/10/02/on-location-time-and-membership-studying-how-aggregate-location-data-can-harm-users-privacy/, [Accessed 01-02-2024].
[54] A. Pyrgelis, C. Troncoso, and E. De Cristofaro, "Knock knock, who's there? Membership inference on aggregate location data," *arXiv preprint arXiv:1708.06145*, 2017.
[55] J. Chipperfield, D. Gow, and B. Loong, "The australian bureau of statistics and releasing frequency tables via a remote server," *Statistical Journal of the IAOS*, vol. 32, no. 1, pp. 53–64, 2016.
[56] Y. Rinott, C. M. O'Keefe, N. Shlomo, and C. Skinner, "Confidentiality and differential privacy in the dissemination of frequency tables," *Statistical Science*, vol. 33, no. 3, pp. 358–385, 2018.
[57] 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.
[58] K. Chatzikokolakis, N. Fernandes, and C. Palamidessi, "Comparing systems: Max-case refinement orders and application to differential privacy," in *2019 IEEE 32nd Computer Security Foundations Symposium (CSF)*. IEEE, 2019, pp. 442–44 215.
[59] L. Rocher, J. M. Hendrickx, and Y.-A. de Montjoye, "Estimating the success of re-identifications in incomplete datasets using generative models," *Nature communications*, vol. 10, no. 1, p. 3069, 2019.
[60] J. Hua, Z. Xiong, J. Lowey, E. Suh, and E. R. Dougherty, "Optimal number of features as a function of sample size for various classification rules," *Bioinformatics*, vol. 21, no. 8, pp. 1509–1515, 2005.

A Statistical Analysis

Hypothesis and test description. We perform a one-sided two-sample t-test to evaluate whether DeSIA shows a statistically significant improvement in AUC over reconstruction-based methods. Specifically, we test the following hypotheses:

$$H_0 : AUC_{DeSIA} \leq AUC_{method}, \quad H_1 : AUC_{DeSIA} > AUC_{method}$$

We compute the p-value of the t-statistic and reject the null hypothesis H_0 in favor of H_1 if $p < 0.05$, indicating that DeSIA achieves a significantly higher AUC than the compared method.

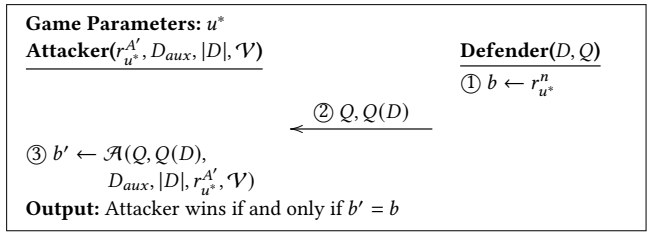

Figure 13: Privacy game for attribute inference attack against fixed aggregate statistics.

Experimental setup. To obtain sufficient statistical power, we run five random sampling trials of aggregate statistics (four additional trials), yielding multiple AUC values for each method. We assume that the AUC values obtained from different random samples follow a Gaussian distribution, and the evaluations of DeSIA and baseline methods are independent.

Results and interpretation. For the experiments in Sections 5.1, 5.3, 5.4, 5.6, 6, 7.1, and 7.2, almost all p-values are significantly small, demonstrating that DeSIA consistently outperforms the compared methods in terms of AUC. The only exception is the comparison between DeSIA and CIP-rand on the PPMF dataset with $\frac{m}{s} = 2^{-4}$, where the number of released aggregates is very small. In this case, the limited leakage makes it difficult to distinguish the performance of DeSIA from CIP-rand.

For the experiments in Sections 5.2 and 7.1, none of the p-values are significantly small. This indicates that we cannot reject the null hypothesis that DeSIA is better, supporting the conclusion that DeSIA maintains its performance when (a) the auxiliary dataset is replaced with synthetic data, and (b) a different machine learning model is used in the stochastic module $\mathcal{M}_S$.

B Attribute Inference Attack as a Privacy Game with Preserved Marginal Distribution and Correlations

While the privacy game used throughout the paper (Fig. 1) eliminates both the marginal distribution of the sensitive attribute and correlations between sensitive and non-sensitive attributes, we here look into the impact of preserving these two components on inferring the sensitive attribute. We propose an alternative privacy game (Fig. 13) in which the marginal distribution of the sensitive attribute and the correlations with non-sensitive attributes are preserved.

In this setting, an attacker may exploit the marginal distribution of the sensitive attribute and its correlations with non-sensitive attributes to infer the sensitive value of the target user. If these components are not known a priori, the attacker can use an auxiliary dataset to estimate them and perform imputation. Here, we train a Random Forest classifier on the auxiliary dataset D_{aux}, using the non-sensitive attributes A' as input features and the sensitive attribute a_n as the prediction target. The trained model is then applied to impute the sensitive attribute of the target user u^* based solely on their non-sensitive attributes. On the PPMF dataset, this straightforward imputation attack achieves an almost ideal AUC of 0.998, despite no aggregate statistics being released. This result shows that, when marginal distribution and correlations are preserved,

Released Tables	AUC ↑
P12 alone	0.52
P12 + P12h	0.64
P12 + P12h + P12i	0.69
All tables (for reference)	0.75

Table 5: Marginal contribution of releasing contingency tables to privacy leakage, measured as AUC for DeSIA. The table shows the incremental leakage as additional related tables are released.

the privacy risk might be dominated by imputation rather than by the aggregate release mechanism. Therefore, eliminating both the marginal distribution and the correlations enables a cleaner and easier estimation of the privacy risk caused solely by the released aggregates.

C Marginal Contributions of Complete Contingency Table Releases

We here evaluate the marginal privacy risk associated with releasing complete contingency tables. This analysis considers two scenarios: (a) release in isolation, when the target table is released without any other tables, and (b) release in complement, when the target table is released alongside other related tables, allowing us to measure the incremental leakage. We focus on the highly related contingency tables P12 (sex by age), P12h (sex by age for only Hispanic/Latino), and P12i (sex by age for only race White and not Hispanic/Latino). We select P12h and P12i, as they provide more fine-grained statistics than P12 and also include the sensitive attribute.

Table 5 shows that releasing a single table in isolation produces limited leakage, lower than the full release. The AUC of releasing P12 alone is not significantly higher than the random guess baseline, as confirmed by a one-sided, one-sample t-test against the null hypothesis that the AUC does not exceed 0.5 ($p > 0.05$). We observe that each additional related table can increase the privacy risk, as reflected in the higher AUC values, with the corresponding AUC values statistically significantly higher than a random guess ($p < 0.05$). Overall, this analysis confirms that our framework can quantify both isolated and incremental contributions of contingency tables, providing actionable insights for data curators before releasing aggregates.

D Attack Performance on Non-Deterministically Vulnerable Users

The deterministic module $\mathcal{M}_D$ identifies deterministically vulnerable users, those whose value of the sensitive attribute is uniquely determined by the released aggregate statistics. These users are likely among the easiest users to attack for all methods. We now compare the performance of all methods when excluding these target users.

Figure 14 (main) shows DeSIA to again outperform the reconstruction-based methods, reaching an AUC of 0.67 and maintaining a strong TPR@kFPR even for those more difficult target users (TPR@10^{-3}FPR of 0.06). Interestingly, RAP suffers more than CIP from this exclusion with initialization having a marginal impact on the overall performance of both CIP and RAP.

E Uniqueness of the target user

One of our assumptions is that the target user u^* is unique in the target dataset D with their values for the non-sensitive attribute. This assumption, in line with the literature [6–8], often holds in practice [59]. Importantly, this assumption ensures the problem is well-posed, meaning that there is only an unambiguous ground-truth value of $r^n_{u^*}$ for attribute inference.

Value-unique users, introduced by Gaddoti et al. [8], are the users sharing both non-sensitive values and the same sensitive value in D. Same as unique users, the ground truth value of value-unique users is also unambiguous in the context of attribute inference. We adapt DeSIA to value-unique users by reformulating the uniqueness constraint and enforcing value-uniqueness in the shadow datasets.

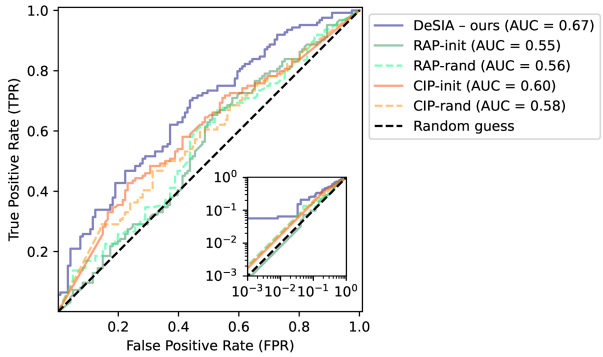

Figure 14: Performance on the PPMF dataset of our attack and the baseline reconstruction-based attacks at predicting the value of the sensitive attribute for only the target users who were not found to be deterministically vulnerable. The inset plot shares the same axes as the main plot.

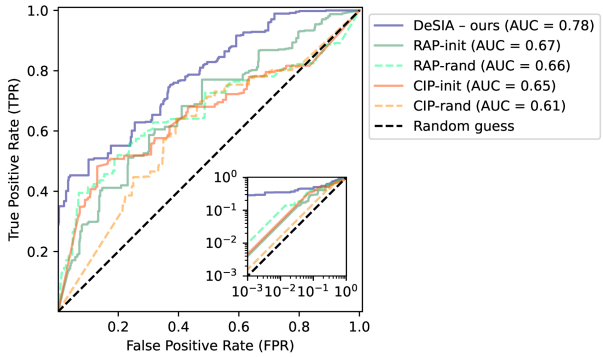

Figure 15: Performance on the PPMF dataset of our attack and the baseline reconstruction-based attacks at predicting the value of the sensitive attribute for value-unique users. The inset plot shares the same axes as the main plot.

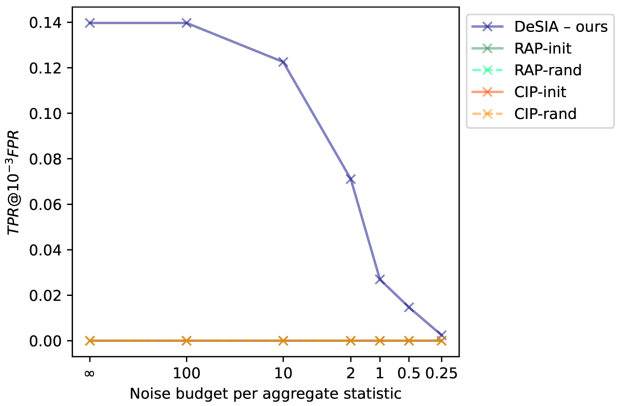

Figure 16: Robustness of the attacks to noise added to released aggregates on the most vulnerable users in the low-FPR regime. We report the mean AUC across 3 independent noisy aggregate statistic releases. The evaluation is performed on the PPMF dataset.

We reformulate the uniqueness constraint c_t into value-uniqueness constraints c_t^1, c_t^2. The constraint c_t^1 reflects that the multiplicity of the value-unique record $x_{(r^1_{u^*},\ldots,r^{n-1}_{u^*},v_n)}$ should be greater or equal to 1, while the constraint c_t^2 means the values of sensitive attributes among the value-unique users are the same.

$$c_t^1 : 1 \leq \sum_{v_n \in \mathcal{V}_n} x_{(r^1_{u^*},\ldots,r^{n-1}_{u^*},v_n)}.$$

$$c_t^2 : 0 = \sum_{v_n \in \mathcal{V}_n} x^2_{(r^1_{u^*},\ldots,r^{n-1}_{u^*},v_n)} - \Big(\sum_{v'_n \in \mathcal{V}_n \setminus \{v_n\}} x_{(r^1_{u^*},\ldots,r^{n-1}_{u^*},v'_n)}\Big)^2.$$

We enforce value-uniqueness in the shadow datasets by post-processing each shadow dataset after the randomization of the sensitive attribute.

Figure 15 shows DeSIA to again be an effective method for evaluating the attribute inference risk of this broader category of target users. It outperforms the other methods, reaching an AUC of 0.78, compared to 0.67 for the second-best RAP-init. DeSIA also identifies highly vulnerable value-unique users, achieving a TPR@10^{-3}FPR of 0.29.

F Robustness to Laplace Noise Addition in Low-FPR Regime

In line with the literature [10], we compare the attack performance of DeSIA with that of other reconstruction-based methods identifying the most vulnerable users under noisy aggregate statistics. Figure 16 shows that DeSIA is the only method achieving a non-zero TPR@10^{-3}FPR across noise levels, indicating that it can successfully infer the sensitive attributes of the most vulnerable users. In contrast, all reconstruction-based methods perform no better than random guessing in this regime, failing to correctly infer any sensitive attributes for these high-risk records.

G Impact of the Number of Aggregates in Low-FPR Regime

We next study how the number of released aggregate statistics affects the ability of attacks to identify the most vulnerable users. Figure 17 shows that DeSIA consistently outperforms reconstruction-based methods by achieving a positive TPR@10^{-3}FPR across all numbers of released aggregates. In contrast, all state-of-the-art methods exhibit zero true positives in this regime, indicating an inability to infer the sensitive attributes of the most vulnerable records. These results further demonstrate DeSIA's effectiveness not only in overall AUC, but also in identifying high-risk individuals under strict false-positive constraints.

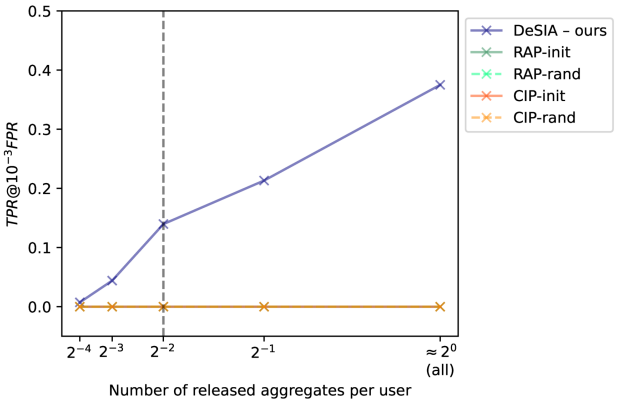

Figure 17: Impact of the number of released aggregates on the performance of inferring the sensitive attribute of the most vulnerable users on the PPMF dataset of all methods in the low-FPR regime. The vertical dotted line indicates the default value taken for the other experiments

H Robustness to Laplace Noise Addition on the ACS Dataset

Analogous to Section 5.3 on the PPMF dataset, we evaluate the robustness of DeSIA and state-of-the-art methods on the ACS dataset under varying levels of Laplace noise added to the released aggregates.

Figure 18 shows that DeSIA consistently outperforms reconstruction-based methods across all noise levels. Similar to the trends observed on the PPMF dataset, only DeSIA and RAP remain robust and achieve performance above the random-guess baseline under higher noise levels. In contrast, CIP can tolerate only limited noise: when noise does not induce conflicting constraints, inference succeeds, but performance degrades sharply once conflicts arise.

I Impact of the Number of Aggregates on the ACS Dataset

Following Section 5.4 on the PPMF dataset, we study how the number of released aggregate statistics affects attack performance on the ACS dataset.

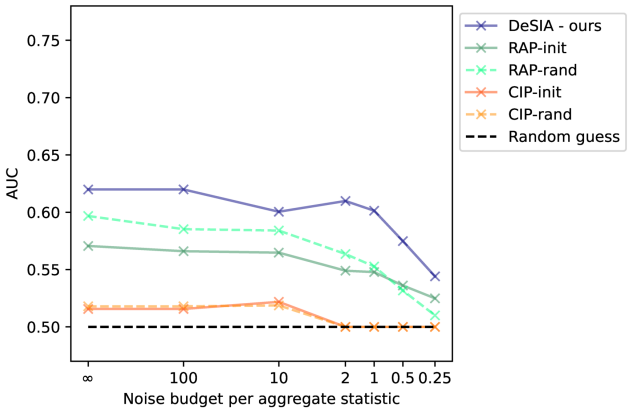

Figure 18: Robustness of the attacks to noise added to released aggregates on the ACS dataset. We report the mean AUC across 3 independent noisy aggregate statistic releases.

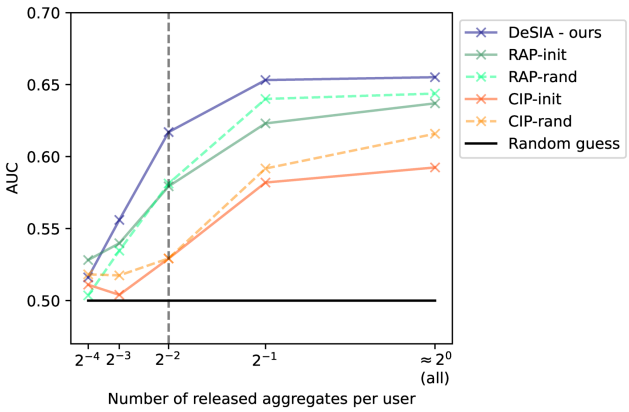

Figure 19: Impact of the number of released aggregates on the performance of DeSIA and other reconstruction-based methods inferring the sensitive attribute on the ACS dataset. The vertical dotted line indicates the default value taken for the other experiments

Figure 19 shows that DeSIA outperforms reconstruction-based methods across all numbers of released aggregates. As with the PPMF dataset, increasing the number of aggregates leads to higher AUC for all methods, indicating increased inference risk as more information is released. Among reconstruction-based attacks, RAP generally performs slightly better than CIP across most settings, mirroring the trends observed on PPMF.

J Impact of the Number of Shadow Dataset

We study the impact of the hyperparameter N, the number of shadow datasets, on DeSIA's attack performance. The deterministic module $\mathcal{M}_D$ does not have tunable hyperparameters, so we focus on $\mathcal{M}_S$, which trains a machine learning model using m features (aggregate statistics) and N samples (shadow datasets).

While [60] recommends, as a rule of thumb, to have an order of magnitude $\sqrt{n}$ attributes for a task with n data points, in our evaluation, we select a fixed value of $N = 20{,}000$ shadow datasets.

Figure 20 shows that attack performance plateaus after $N = 3{,}000$, indicating that $N = 20{,}000$ is sufficient to achieve stable and robust performance while remaining safe from potential variability due to sampling. Using fewer shadow datasets reduces AUC performance, while increasing N beyond this point yields no substantial gains.

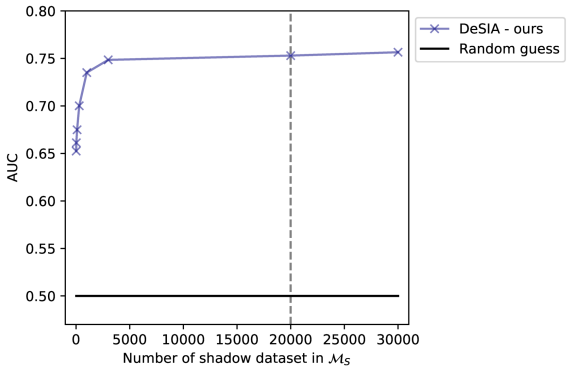

Figure 20: Performance on the PPMF dataset of our attack where the stochastic module $\mathcal{M}_S$ uses different number of shadow datasets.

K Extension to Multi-Valued Sensitive Attributes

In line with the literature [6–8, 47], and for simplicity, our evaluation focuses on binary sensitive attributes. Here, we discuss the extension of DeSIA to multi-valued categorical and continuous attributes.

Multi-valued categorical sensitive attributes. DeSIA can be instantiated out-of-the-box on multi-value categorical attributes. To show this, we instantiate DeSIA on the ACS dataset with the same non-sensitive attributes and following sensitive attributes: "Veteran service-connected disability" and "Number of times married", with 3 and 4 possible categorical values, respectively. In both cases, DeSIA achieves an AUC of 0.60.

Continuous sensitive attributes. Like other attack methods, DeSIA is not directly applicable to continuous sensitive attributes, where "exact inference" is inherently a coarse risk metric. For such cases, DeSIA can be extended in two ways. First, continuous attributes can be discretized into multi-valued categorical attributes, allowing DeSIA to be applied directly. Second, if discretization is not feasible, the stochastic module of DeSIA can be adapted by replacing the classification model with a regression model to predict the continuous value.