Privacy Attacks on Schedule-Driven Data

Stephan A. Fahrenkrog-Petersen[1,2], **Arik Senderovich**[3], **Alexandra Tichauer**[1], **Ali Kaan Tutak**[1], **J. Christopher Beck**[4], **Matthias Weidlich**[1]

[1]Humboldt-Universität zu Berlin
Unter den Linden 6, 10117 Berlin, Germany
{fahrenks,tichauea,tutakali,matthias.weidlich}@hu-berlin.de
[2] Weizenbaum Insitute for the Networked Society
Hardenbergstraße 32, 10623 Berlin, Germany
[3] York University
4700 Keele St, Toronto, ON M3J 1P3, Canada
sariks@yorku.ca
[4] University of Toronto
5 King's College Rd, Toronto ON M5S 3G8, Canada
jcb@mie.utoronto.ca

Abstract

Schedules define how resources process jobs in diverse domains, reaching from healthcare to transportation, and, therefore, denote a valuable starting point for analysis of the underlying system. However, publishing a schedule may disclose private information on the considered jobs. In this paper, we provide a first threat model for published schedules, thereby defining a completely new class of data privacy problems. We then propose distance-based measures to assess the privacy loss incurred by a published schedule, and show their theoretical properties for an uninformed adversary, which can be used as a benchmark for informed attacks. We show how an informed attack on a published schedule can be phrased as an inverse scheduling problem. We instantiate this idea by formulating the inverse of a well-studied single-machine scheduling problem, namely minimizing the total weighted completion times. An empirical evaluation for synthetic scheduling problems shows the effectiveness of informed privacy attacks and compares the results to theoretical bounds on uninformed attacks.

Introduction

Schedule-driven systems are pervasive in our lives in areas such as outpatient clinics , production lines, and public transportation systems. To investigate improvements in system performance, it is common for an analyst to have access to the schedules. For example, Zhang et al. (2019) use data mining techniques to infer root-causes for failures based on historical schedules, Li and Olafsson (2005) use schedules to derive dispatching rules to solve future problem instances, and Kim and Nembhard (2013) use past schedules to facilitate real-time decision making.

Publishing the schedule, as in any data publishing scenario (Fung et al. 2010), may result in loss of private information. Specifically, the schedule of jobs depends on their features that are partially private and may constitute sensitive information, such as a medical priorities of patients in a hospital. As we shall demonstrate, given an *optimal* schedule, an adversary can potentially infer such private information, especially in the presence of even minimal background knowledge about the jobs in the published data (Narayanan and Shmatikov 2008).

Despite the pervasiveness of the analysis of schedule-driven data, the induced privacy risks have not yet been studied in the literature. Techniques for privacy-aware recommender systems (Yang, Qu, and Cudré-Mauroux 2018), which may seem related, are not directly applicable, due to fundamental differences in the problem structure: in schedules, certain features are unknown for *all* jobs, whereas common attacks on recommender systems are based on the full set of features being available for *some* items.

In this paper, we provide a first study of privacy attacks on schedule-driven data. To this end, we contribute a threat model that clarifies the knowledge to be used by an adversary. Next, we focus on the privacy gain that an adversary may achieve and develop measures for the privacy loss incurred by a published schedule.

Turning to the actual privacy attack, we first consider an *uninformed* adversary, who ignores the schedule and guesses randomly. For this baseline, we study formal properties of our loss functions and derive bounds on the expected value of the total privacy loss. For effective attacks, we present a framework for an *informed adversary*, who uses the published schedule for an attack based on *inverse scheduling* (Brucker and Shakhlevich 2009). For an important, basic scheduling problem, namely minimizing the total weighted completion time in a single-machine setting (Pinedo 2016, Chap. 3.1), we formulate the attack as a constraint satisfaction problem and explore its computational complexity.

We summarize our contributions as follows:

1) We present a threat model for schedules based on the public and private features used to derive an optimal schedule.
2) We devise measures for the privacy loss incurred by a published schedule.
3) We analyze the properties of our privacy loss measures

for a randomly guessing adversary.

4) We present a framework for informed privacy attacks on schedules following the idea of inverse scheduling and provide complexity results.

Experiments with synthetic schedules indicate that informed privacy attacks indeed pose a threat and allow adversaries to make inference on private attributes of jobs.

Background

Privacy Preservation. Privacy-preserving release of datasets has been widely studied (Fung et al. 2010; Wagner and Eckhoff 2018), with *differential privacy* (Dwork 2008) emerging as the most prominent privacy guarantee. Usually this guarantee is achieved by adding noise to the results of queries, so that the information that an adversary can learn is limited by bounding the impact one individual has on the query result.

For privacy-preserving publishing of a dataset independent of a specific query, guarantees such as *k-anonymity* (Sweeney 2002) and *l-diversity* (Machanavajjhala et al. 2007) have been proposed. The former ensures that at least k individuals are indistinguishable from each other in a published dataset, whereas the latter extends the guarantee to sensitive information for a group of individuals. Both types of strategies guarantee a level of privacy without bounding the generally inevitable loss in utility of the published data for other purposes (Brickell and Shmatikov 2008). As such, techniques to optimize the resulting utility are an active field of research (LeFevre, DeWitt, and Ramakrishnan 2006; Fioretto, Hentenryck, and Zhu 2021) with approaches tailored to specific types of data and analysis purposes. However, no work has investigated the privacy of published schedules.

Although this paper is the first one to study privacy attacks on published schedules, there is previous work on privacy-aware scheduling. For example, some work focussed on the construction of collective schedules maintaining privacy (Herlea et al. 2001; Wallace and Freuder 2005; Bilogrevic et al. 2011). Kadloor and Kiyavash analyze privacy in the context of shared event schedulers, studying the trade-off between utility and privacy of the schedules (Kadloor and Kiyavash 2015). Unlike our paper, these approaches generate a protected schedule, while considering privacy loss to be part of the scheduling problem. This results in sub-optimal, yet private schedules. Furthermore, these approaches focused on protecting the privacy between participants in the scheduling process itself. In our approach, the original scheduling problem is solved to optimality and we consider the privacy loss implications of making the resulting schedule public.

Scheduling Models. We discuss the general class of problems for single-resource scheduling, before turning to the problem types that we examine throughout this work. We consider scheduling problems that comprise a set of n jobs to be processed by a single resource. Each job is assigned a vector of m input features (e.g., release times, due dates and processing times). The resulting schedule must satisfy a set of constraints and attempt to minimize a given objective function. Formally, a single-resource scheduling problem is a tuple $\Pi = (J, X, C, \phi)$ where:

- $J = \{j_i\}_{i=1}^n$ is the set of jobs to be processed;
- X is an $n \times m$ job feature matrix with $x_{i,j}$ being the j-th feature of the i-th job;
- $C \subseteq 2^{\mathbb{R}^n}$ is the set of constraints imposed; and
- ϕ is the objective function measuring schedule quality.

A solution, or schedule, is a vector of start times for the n jobs, $s \in \mathbb{R}^n$. We denote the domain of all possible job feature combinations as $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_m$ with $\mathcal{X}_j$ representing the domain of feature $j = 1, \ldots, m$. For instance, if feature j is the processing time, $\mathcal{X}_j = [10, 80]$ would define that jobs run for at least 10 and at most 80 time units.

Every constraint $c \in C$ is defined as a subset of schedules, with every schedule in c satisfying a set of conditions that may depend on features X. The set of feasible schedules for problem Π is denoted $\mathcal{S}_\Pi$, and it is the intersection of the constraints in C. For example, single-resource scheduling problems require the constraint that job executions do not overlap. Further constraints, such as precedence constraints between specific jobs, can be added. We only consider problems that assume non-preemptive schedules. That is, jobs cannot be paused or stopped once their execution has started.

To assess the quality of a schedule, we define an objective function, $\phi : \mathbb{R}^n \times \mathcal{X} \to \mathbb{R}$ that assigns a real value to a given schedule and job features. An optimal solution to Π is a schedule, $s^* \in \mathbb{R}^n$, that satisfies the constraints, while $\phi(s^*)$ is minimal among all vectors that satisfy the constraints. By $\sigma^* \in \mathbb{N}^+$, we denote the job permutation derived from s^*, e.g., if s_i^* is the smallest start time in s^* then $\sigma_1^* = i$.

TWCT Scheduling. As a specific type of the above class of problems, we consider total weighted completion time scheduling. $\Pi = (J, X, C, \phi)$ is defined by an arbitrary number of jobs, each being described by two features: its processing time and its weight. Formally, X is an $n \times 2$ matrix, where $x_{i,1} = p_i > 0$ is the processing time, and $x_{i,2} = w_i > 0$ is the weight of job i. All features are assumed to have a finite, integer-valued domain. The only constraints applied to the problem are non-overlapping and non-preemptive job executions. The objective function is the sum of weighted completion times of all jobs.

The optimal schedule for TWCT can be found in polynomial time using the Weighted Shortest Processing Time (WSPT) rule (Pinedo 2016). That is, in an optimal solution, the jobs are ordered in a non-increasing manner based on the ratios w_i/p_i. Note that optimal TWCT solutions will always be earliest-start schedules (jobs that can start are not delayed), since any such delay would incur an increase in the objective value.

Inverse Scheduling. Inverse scheduling is a special case of inverse optimization problems, where the forward optimization problem is a scheduling problem (Brucker and Shakhlevich 2009). Using our notation, the inverse scheduling problem (ISP) considers $\Pi = (J, X, C, \phi)$ that has an optimal solution s_0. The goal is to find a new set of job features (e.g., processing times or due dates), X^*, close to X under some norm $|| \cdot ||$, that would yield a target schedule

s^*. In the TWCT scenario, an inverse scheduling problem would be to find a new set of weights w_i^* that would achieve a target schedule s^*.

Privacy Attacks on Schedules

Motivating Example

Consider a physician's clinic in a European country that prioritizes patients according to the quality of their insurance policy, rated $1, \ldots, 5$ (5 being the highest quality insurance policy). On a typical day, the clinic administrator must schedule $n = 10$ patients, with each patient ($i \in \{1 \ldots, 10\}$) requiring a service duration (p_i), and having an insurance ranking (w_i). The aim of the administrator is to minimize the sum of completion times, while taking into account the fact that high quality patients should be served earlier (naturally leading to the TWCT problem). The clinic wishes to publish the schedule to inform patients and assure timely arrivals. Uploading the data to the website exposes it to a threat. Alternatively, an attacker might observe the sequence in which patients are served. The adversary aims to learn private patient data to sell to other health insurance agencies. Similar situations can arise in other domains such as the delivery scheduling, where high-value customers are served before to low-priority customers.

Threat Model

Our threat model involves an adversary with insights into the scheduling problem that was solved to obtain the published schedule s. In particular, the adversary knows the set of jobs J, the constraint set C, the objective function ϕ, and a subset X_{pub} of the job features that govern the scheduling problem, i.e., X_{pub} is an $n \times m'$ matrix, with $m' < m$ and m being the total number of features.

The adversary lacks full knowledge of the features and wishes to determine the values of X that are not given by X_{pub}, which we denote by X_{priv}. It holds that $X = [X_{pub}\ X_{priv}]$ and we write $\mathcal{X}_{priv}$ for the domain of all possible combinations of job feature values not known by the adversary. Under this model, the adversary is rather powerful, a realistic assumption in many applications. For instance, in the case of the clinic from our motivating example, the number of patients can be directly derived from the published schedule; many constraints are common knowledge (e.g., non-overlapping treatments); the objective is given through prevalent operational principles (e.g., total weighted completion time); and some job features can be estimated based on the schedule (e.g., treatment times).

Let $\Pi(X_{priv})$ denote the single-resource scheduling problem with fixed J, C, ϕ, X_{pub} (information known to the adversary), and varying X_{priv} be the matrix of private features (e.g., job weights). In addition, we define a solver f for $\Pi(X_{priv})$ as a function of X_{priv}: it solves $\Pi(X_{priv})$ and returns an optimal schedule $s^* \in \mathbb{R}^n$. In this work, we assume that in case of a tie between optimal schedules, f chooses one of them in a deterministic pre-defined manner.

The idea of the threat is illustrated in Figure 1. On the left-hand-side, we see that the true values of private features lead to the optimal, published schedule. Yet, this schedule denotes an opportunity for inferring a set of possible values for the private features (see right-hand-side of the figure), thus exposing X_{priv}. Later, we discuss specific realizations of the attack function, g. Note that the left-most and right-most ovals of Figure 1 denote the same feature space.

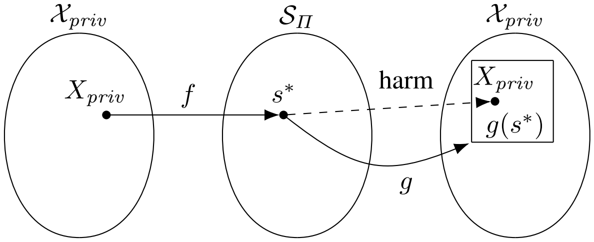

Figure 1: The threat model for a published schedule s^*. The three ovals correspond to the domain of private features, the space of feasible schedules, and again, the set of private features, respectively. The rectangle denotes the set of values of private features $g(s^*)$ inferred through the attack.

Privacy Loss in Published Schedules

Next, we turn to quantifying the privacy risk for the aforementioned threat. We first define a distance-based loss (DBL) that assesses the similarity between the true X_{priv} and a set of candidate values $\mathbb{Y}$ generated by the adversary. The set is based on the knowledge that the adversary possesses (as discussed in the threat model) and an inference procedure that we purposely keep vague at this point. Based on the DBL, we then define two privacy loss functions, namely the local privacy loss (LPL) that reflects the fact that losing privacy of a single feature j for a single job i matters, and the total privacy loss (TPL) that aggregates LPL by taking the maximum across multiple jobs and features.

Privacy Loss Principles. To derive privacy loss functions, we set the following three principles:

P1 The smaller the distance between $\mathbb{Y}$ and X_{priv}, the more successful the attack and the higher the loss.

P2 Gaining any information about a subset of the jobs is considered a (partially) successful attack; fully successful, if all feature values of all jobs are retrieved.

P3 The privacy loss for the entire schedule cannot be lower than the highest privacy loss among the jobs.

Returning to our example, P1 states that there is a positive privacy loss if the adversary finds weights that are close (in some metric) to the actual weights. P2 states that inferring private information about a subset of patients and features is a negative outcome. Lastly, P3 implies that leaking private information about a single patient is as negative as leaking information about all patients. In the remainder of the section, we build upon the three principles when defining our privacy loss measures.

Distance-Based Loss. Let Y^k be a single candidate matrix of feature values from the set $\mathbb{Y}$, with $k = 1, \ldots, N$ and $N = |\mathbb{Y}|$. Further, let $Y^k_{i,j}$ be the j-th feature of the i-th job in the k-th guess, and let $Y_{i,j} = [Y^1_{i,j}, \ldots, Y^N_{i,j}]$ be the set of

N candidate values for job i and feature j. Note that $Y_{i,j}$ is a multi-set, i.e., feature values may appear more than once.

We define d to be a metric that measures the distance between two feature values. The metric can be adapted according to the feature domain. For example, for numeric domains one can use the absolute distance, i.e.,

$$d(x_{i,j}, Y^k_{i,j}) = |x_{i,j} - Y^k_{i,j}|, \tag{1}$$

while for discrete domains one can use the discrete metric, namely

$$d(x_{i,j}, Y^k_{i,j}) = \begin{cases} 0 & \text{if } x_{i,j} = Y^k_{i,j}, \\ 1 & \text{otherwise.} \end{cases} \tag{2}$$

Moreover, let $\hat{p}_Y(x)$ be the (empirical) frequency of x in the multi-set $Y_{i,j}$ (with $\hat{p}_Y(x) = 0$ if $x \notin Y_{i,j}$).

Definition 1 (Distance-based Loss (DBL))**.** Given a job-feature value $x_{i,j}$, a published schedule s, and $Y_{i,j}$, the distance-based loss (DBL) is defined as

$$D(x_{i,j}, Y_{i,j}) = \int_{x \in \mathcal{X}_j} d(x_{i,j}, x)\hat{p}_Y(x)\, \mathrm{d}x, \tag{3}$$

with the integral interpreted as the empirical Lebesgue measure, meaning that it turns into a sum for discrete domains.

The distance-based loss is equivalent to the expected value under $\hat{p}_Y$ of the distance between a random value of a feature and the domain $\mathcal{X}$.

Normalizing DBL. DBL will return smaller values when the adversary guesses values in $\mathbb{Y}$ that are closer to X_{priv} (thus adhering to P1). However, the value of the DBL may differ depending on the underlying feature domain. Therefore, to compute the privacy loss across multiple feature domains of different sizes, we must first normalize the DBL. Intuitively, the closer the set $\mathbb{Y}$ of some feature j to the entire domain $\mathcal{X}_j$, the smaller the threat, since the adversary has gained little information by obtaining $\mathbb{Y}$. We integrate this intuition into the normalized distance-based loss (NDBL). Specifically, replacing $Y_{i,j}$ with $\mathcal{X}_j$ in Eq. (3) leads to the following normalization factor for discrete domains:

$$D(x_{i,j}, \mathcal{X}_j) = \frac{\sum_{x \in \mathcal{X}_j} d(x_{i,j}, x)}{|\mathcal{X}_j|}. \tag{4}$$

We can derive a similar expression for continuous domains. The normalized DBL for a single feature-job pair can be written as,

$$D_N(x_{i,j}, Y_{i,j}) = \frac{D(x_{i,j}, Y_{i,j})}{D(x_{i,j}, \mathcal{X}_j)}.$$

Local Privacy Loss. The local privacy loss (LPL) quantifies the loss for a single feature and single job. Intuitively, if the values of $\mathbb{Y}$ resemble those of the domains of X_{priv} the privacy loss is considered low. However, if the values in $\mathbb{Y}$ are close to the true values, the privacy loss should be considered high.

Definition 2 (Local Privacy Loss (LPL))**.** Given a feature j of job i and a published schedule s, the local privacy loss $\xi_{i,j}$ is defined as

$$\xi_{i,j}(s) = 1 - D_N(x_{i,j}, Y_{i,j}). \tag{5}$$

Using LPL we can generate a privacy loss value for every entry of the job-feature matrix X_{priv}. Note that due to normalization, the farther (closer) $D(x_{i,j}, \mathbb{Y})$ from (to) a uniform distribution, the closer the LPL to 1 (0).

Total Privacy Loss. The LPL satisfies P1 and P2, since it is based on the distance between the true value and the values collected by the adversary and partial success is taken into account. Next, we derive the total privacy loss (TPL) of a schedule across all jobs and features, taking into account the third principle.

Definition 3 (Total Privacy Loss (TPL))**.** The total privacy loss of a published schedule s is defined as

$$\boldsymbol{\xi}(s) = \max_{i,j \in \mathcal{X}_{priv}} \xi_{i,j}(s). \tag{6}$$

TPL satisfies all three principles (P1-P3) as it is based on the distance-based privacy loss, takes into account partial adversarial success, and the loss of the overall schedule is as high as that of the job with the highest loss.

Loss Properties for Uninformed Attacks

Any privacy protection mechanism would strive to publish a schedule that would result in a failed attack on the private features. Therefore, we require an attack that would serve a benchmark: if the TPL of a schedule is similar to that benchmark, we will consider the schedule to be well-protected.

In this part, we consider the failed attack benchmark to be the total privacy loss under an *uninformed attack*, an attack in which the adversary randomly guesses one value from the domain of the private feature. We explore the theoretical properties of TPL under an uninformed attack and establish the following sequence of results: (1) the expected value and the variance of the TPL, (2) a limit theorem for the behavior of the TPL for an increasing size of the candidate set $\mathbb{Y}$, and (3) lower and upper bounds on the TPL. The third result, which builds upon the first two, enables us to devise failure benchmarks for published schedules.

Expectation, Variance, and a Limit Theorem

Since we assume that jobs and their features are independent we consider the case of a single job and a single feature, $n = m = 1$, which can be easily extended to multiple jobs with multiple features.

Expectation and Variance of TPL. Let $X \in \mathcal{X}$ be a random variable that corresponds to a single true feature-value of a single job. Further, let Y be the random variable of a single guess that the uninformed adversary makes. In an uninformed attack the adversary does not observe the schedule, Y is independent of s. Furthermore, since the adversary is aware of the domain of X, they sample Y from $\mathcal{X}$. Thus, we can write the total privacy loss as

$$\xi(X, Y) = 1 - \frac{d(X, Y)}{D(X, \mathcal{X})}. \tag{7}$$

We shall now show that the expected total privacy loss for a single random guess of the adversary is 0 and provide an expression for the variance of the TPL.

For simplicity, we assume discrete domains; the concepts are however easily extensible to continuous domains. We assume that both X and Y are independently drawn from $\mathcal{X}$ using uniform sampling (probability of $1/|\mathcal{X}|$ is assigned to each value) we can write,

$$\mathbb{E}\left[\frac{d(X,Y)}{D(X,\mathcal{X})}\right] = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{X}}\left(\frac{d(x,y)}{d(x,\mathcal{X})}\cdot P(X=x)P(Y=y)\right) =$$

$$= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{X}}\left(d(x,y)\cdot\frac{|\mathcal{X}|}{\sum_{w\in\mathcal{X}}(x,w)}\cdot\frac{1}{|\mathcal{X}|^2}\right) =$$

$$= \sum_{x\in\mathcal{X}}\left(\frac{\sum_{y\in\mathcal{X}}(x,y)}{\sum_{w\in\mathcal{X}}(x,w)}\right)\cdot\frac{1}{|\mathcal{X}|} = 1, \tag{8}$$

and thus,

$$\mathbb{E}[\xi(X,Y)] = \mathbb{E}\left[1-\frac{d(X,Y)}{D(X,\mathcal{X})}\right] = 0. \tag{9}$$

Moreover, the variance of the random variable is derived as

$$Var[\xi(X,Y)] = Var\left[\frac{d(X,Y)}{D(X,\mathcal{X})}\right] =$$

$$= \mathbb{E}\left[\left(\frac{d(X,Y)}{D(X,\mathcal{X})}\right)^2\right] - \mathbb{E}\left[\frac{d(X,Y)}{D(X,\mathcal{X})}\right]^2 =$$

$$= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{X}}\left(d(x,y)^2\cdot\frac{|\mathcal{X}|^2}{\left(\sum_{w\in\mathcal{X}}(x,w)\right)^2}\cdot\frac{1}{|\mathcal{X}|^2}\right) - 1 =$$

$$= \sum_{x\in\mathcal{X}}\left(\frac{\sum_{y\in\mathcal{X}}d(x,y)^2}{\left(\sum_{w\in\mathcal{X}}(x,w)\right)^2}\right) - 1. \tag{10}$$

Returning to the motivating example, we have an integer weight domain $\mathcal{X} = \{1,...,5\}$. Thus, using the absolute difference as a distance measure yields a privacy loss with expected value of 0 and variance of 0.49.

A Limit Theorem for TPL. Assume the adversary produces a set of candidates $\mathbb{Y} = \{Y^1, Y^2, ...\}$ for a single true feature value X. Similar to Eq. (7), we now get

$$\xi(X,\mathbb{Y}) = 1 - \frac{D(X,\mathbb{Y})}{D(X,\mathcal{X})}. \tag{11}$$

Applying the definition of privacy loss, we can carry out the following transformation:

$$\frac{D(X,\mathbb{Y})}{D(X,\mathcal{X})} = \frac{1}{|\mathbb{Y}|}\frac{\left(\sum_{Y\in\mathbb{Y}}d(X,Y)\right)}{D(X,\mathcal{X})}$$

$$= \frac{1}{|\mathbb{Y}|}\sum_{Y\in\mathbb{Y}}\left(\frac{d(X,Y)}{D(X,\mathcal{X})}\right). \tag{12}$$

This expression is equivalent to calculating the mean of sample size $|\mathbb{Y}|$, where the single entries of the sample are drawn from the exact same distribution whose expected value we determined in Eq. (8). From the above we immediately arrive at the following result.

Theorem 1. As $|\mathbb{Y}| \to \infty$, the distribution of $\xi(s)$ converges to a normal distribution with mean 0 and variance $Var[\xi(X,\mathbb{Y})]$ given by

$$Var[\xi(X,\mathbb{Y})] = \frac{Var[\xi(X,Y)]}{|\mathbb{Y}|}.$$

The proof is straightforward from applying the central limit theorem (Feller 1945).

Discussion. At this point, there are several important observations. First, the expected privacy loss for an uninformed adversary is indeed 0, coinciding with our intuitive notion that privacy is preserved when the adversary cannot do better than simply guessing the private features. Second, the total privacy loss can assume negative values. Hence, the uninformed adversary has equal chances of "doing better" and "doing worse" than average. Note that a negative privacy stems from our definition of distance-based loss, which is not bounded below. It does not imply a "privacy gain".

Bounds on Total Privacy Loss

In this part, we continue working with an uninformed adversary to explore bounds on the expectation of the total privacy loss. The aim is to derive reasonable privacy loss benchmarks for a failed attack.

We have already shown that the total privacy loss for an uninformed adversary over a set of candidates converges to a normal random variable. Thus, we now write down the lower and upper bounds on the expectation of the maximum of a sample coming from a normal distribution.

Theorem 2 (Kamath 2015). Let $\boldsymbol{\xi} = \max_{1\le i\le n}\ \xi_i$ where $\xi_i \sim \mathcal{N}(0,\sigma^2)$ are i.i.d. random variables. Then

$$\frac{1}{\sqrt{\pi\log 2}}\sigma\sqrt{\log n} \le \mathbb{E}[\boldsymbol{\xi}] \le \sqrt{2}\sigma\sqrt{\log n}.$$

Since the bounds are valid only for random variables drawn from a single distribution, which is not necessarily the case when considering different features, the theorem is limited to the maximum over one specific feature, and not over the entire feature matrix. The overall bounds for all features could be found by taking the maximum and minimum of the individual feature bounds. Note that the values of these bounds depend both on the number of jobs, n, and on the number of guessed candidates in $|\mathbb{Y}|$, for which we do not have a fixed value for the uninformed adversary.

Figure 2 serves to illustrate the meaning of the described bounds for the motivating example. Specifically, it shows the upper and lower bounds of the expected value of the total privacy loss for a range of different $|\mathbb{Y}|$. The number of jobs is set to 10, while for a given $|\mathbb{Y}|$, the variance will be $\sigma^2 = Var[\xi(X,\mathbb{Y})] = 0.49/|\mathbb{Y}|$ as our domain is $\mathcal{X} = \{1,\ldots,5\}$. The figure shows a steady decline in the bounds on the total privacy loss as the size of the candidate set, $|\mathbb{Y}|$, increases. This is expected as smaller candidate sets have fewer possible values, making random guessing easier.

Discussion. In practice, one can use the following procedure to achieve a benchmark for attempts to prevent privacy loss when publishing schedules. First, obtain the lower and upper bounds on the TPL under an uninformed attack. Set one of them to be the representative of a failed attack. Use an algorithm to alter the schedule with the aim of protecting it, and compare an informed attack on the schedule to the uninformed benchmark. If the gap is insignificant, conclude that the protection attempt was indeed successful.

Note that Figure 2 shows that even an uninformed adversary may still arrive at a positive TPL value. This is a consequence of sampling the random weights from a relatively small domain (compared to the number of jobs), which results in high probability of of having several samples that agree with the true private value. One must take this property of the TPL into account when using uninformed attacks to set a benchmark for low values of the privacy risk.

Inverse Scheduling Attacks

We now turn to present an informed attack on a published schedule based on the notion of *inverse scheduling*. Specifically, when presented with a published schedule, the adversary uses inverse scheduling to collect all possible private feature values that can lead to this optimal schedule (the adversary knows the scheduling problem including the objective function). We consider attacks that enumerate *all* possible feature values and leave other solutions to future work.

More formally, the adversary assumes that the published schedule s is the optimal schedule s^* that solves $\mathit{\Pi}(X_{priv})$. Based thereon, we define the attack as an inverse scheduling problem (ISP) to $\mathit{\Pi}(X_{priv})$. That is, the adversary aims at finding $X_{priv} \in \mathcal{X}_{priv}$, such that $f(X_{priv}) = s^*$. However, there may exist multiple matrices of feature values in $\mathcal{X}_{priv}$ that yield the published schedule. As shown on the right-hand-side of Figure 1, the attack can be written as a mapping,

$$g(s^*) \mapsto \{X' \in \mathcal{X}_{priv} \mid f(X') = s^*\}, \qquad (13)$$

that relates a given schedule to a set of private values. In the context of total privacy loss, we get that $\mathbb{Y} = g(s)$.

Inverse Scheduling Attack for TWCT

Focusing on the TWCT problem, we show how the above attack can be instantiated for a published schedule s. Recall that the TWCT scheduling problem defines two features for each job, i.e., processing times and weights. Under the above threat model, processing times are known by the adversary as they can be estimated directly from the published schedule (by observing start times). In contrast, the job weights are private information that is not entirely disclosed, i.e.,

$$X_{priv}^T = [w_{true} = (w_1, \ldots, w_n)]. \qquad (14)$$

With $\mathcal{X}_w$ being the domain of job weights, the ISP to find a set of candidate matrices of weights is given by

$$g_{twct}(s) = X = \{w \in \mathcal{X}_w \mid f(w) = s\}. \qquad (15)$$

For this setting, the inverse scheduling procedure, g_{twct}, can be represented as the following constraint satisfaction problem (CSP). The input parameters of the CSP are the job permutation σ (derived from s), the known feature values, namely the processing times p.

Moreover, understanding the problem solved to obtain the schedule, the adversary knows the domain of weights (which could also be estimated from the schedule, if unknown). The vector of weights that the adversary attempts to find is the only decision variable. Furthermore, the adversary knows that the forward problem is TWCT and the solver used is f. Hence, since they assume that the schedule is optimal, they know that the jobs must be sorted in a non-increasing order of their weight to processing time ratio, and they are aware of the deterministic tie-breaking mechanism in f that selects between multiple optimal solutions. Based thereon, the ISP can be written as follows:

Inputs:

σ: permutation inferred from s.
p: processing times inferred from s.
$\mathcal{X}_w$: weight domain.

Decision variables:

$w \in \mathcal{X}_w$: vector of weights.

Constraints:

$$\frac{w_{\sigma_i}}{p_{\sigma_i}} \geq \frac{w_{\sigma_{i+1}}}{p_{\sigma_{i+1}}} \qquad \forall\, i \in \{1, \ldots, n-1\}.$$

Output:

W: a set of candidate weight vectors.

Note that, $w_i = p_i$ is always a trivial solution to the problem.

TWCT ISP in our Motivating Example. Consider a TWCT schedule of our clinic with 3 patients having processing times $p = (5, 3, 1)$, published permutation $\sigma = (1, 2, 3)$, and weight domain of $\mathcal{X}_w = \{1, \ldots, 5\}$. The result of the inverse scheduling attack on schedule s is $W = \{(5, 3, 1)\}$, i.e., there is only a single weight vector that leads to s and the adversary is able to fully infer the true weight vector.

Computational Complexity of TWCT ISP

For general ISPs, one can generate solutions by searching through the feature domains and finding X' that lead to s. Finding a single solution may not be polynomial, and finding all solutions will typically require an exponential number of steps. In fact, finding one solution to the inverse problem may be polynomial or hard, depending on the norm $||\cdot||$ and the adjusted feature (Brucker and Shakhlevich 2009). For example, when $||\cdot||$ is Hamming distance and when adjusting processing times, the inverse of the maximum lateness problem is $\mathcal{NP}$-hard. Yet, for the same norm, if one adjusts the due dates, the problem of finding one solution is polynomial. In other words, a polynomial scheduling problem, may or may not result in a computationally hard inverse problem when generating a single solution.

Below, we show that while the number of steps to generate all solutions to the TWCT problem is indeed exponential, one can efficiently generate a sequence of solutions.

Theorem 3. For the TWCT ISP with n jobs that have durations $p_1, \ldots, p_n$ and a weight domain $\mathcal{X}_W$ of size m, the size of $|W|$ is $\mathcal{O}(m^n)$.

Proof. Denote w_0 some solution to TWCT ISP (we know that such solution exists, as we can assign $w_i = p_i, \forall i$).

Then, any vector $w >= w_0$ is also a solution. In the worst case, every element in w_0 can be $\min(\mathcal{X}_w)$, and hence all permutations of length n from domain $\mathcal{X}_w$ are also solutions to the problem. Thus, in the worst-case we may get, $|W| = m^n$, solutions. □

Even though it is practically infeasible to enumerate all solutions in W, generating subsequent solutions into W is polynomial due to the following result.

Theorem 4. Assuming that p_i are positive integers, a single solution w that satisfies the WSPT constraints for n jobs and weight domain size of $|\mathcal{X}_w| = m$ can be computed in $\mathcal{O}\left((n-1)m\right)$.

Proof. The problem of finding a single weight vector $w_1, \ldots, w_n$ can be solved via the following dynamic programming procedure.

1. Set $w_n = \max\left(\mathcal{X}_w\right)$.
2. Set w_{n-1} such that it satisfies the WSPT constraint, namely $w_{n-1} \geq w_n \frac{p_{n-1}}{p_n}$. If it does not, lower w_n (potentially until $w_n = p_n$, in which case $w_{n-1} \geq p_{n-1}$) and repeat Step 2 until w_{n-1} satisfies the constraint. If it does, set w_{n-1} and repeat Step 2 for w_{n-2}.

The procedure requires at most m operations per instance of Step 2 until w_{n-1} is set (in worst case to p_{n-1}), and terminates within at most $n-1$ such instances. □

Next, we define the notion of efficient enumeration.

Definition 4 (Efficient enumeration algorithms (Johnson, Yannakakis, and Papadimitriou 1988))**.** An efficient enumeration algorithm has to enumerate all solutions in such a way that the time between each pair of assignments, between the start of the algorithm and the first solution, and between the last solution and the termination of the algorithm is polynomial in the input size.

Corollary 1. Enumerating the solutions of TWCT ISP by finding new solutions that satisfy the WSPT rule is efficient.

The proof is immediate due to Theorem 4.

In practice Corollary 1 states that if one sets a bound on the number of solutions in $g(s)$, generating a reasonably large number of candidates is feasible in polynomial time.

Empirical Evaluation of Informed Attacks

Finally, we turn an empirical assessment of the introduced attacks on published schedules based on inverse scheduling[1]. In addition to the bounds on the expected TPL under an uninformed attack, Figure 2 shows a box-plot of the total privacy loss (TPL) under an informed attack. Specifically, an informed adversary attacks $10,000$ synthetically generated schedules that all match our motivating example. For each schedule, we solve the ISP by enumerating all possible solutions, and compute the TPL based on the returned candidate set by the attack. For example, if for a given schedule the candidate size was 10 and the TPL was 0.9, we add that result to the value of $x = 0.9$.

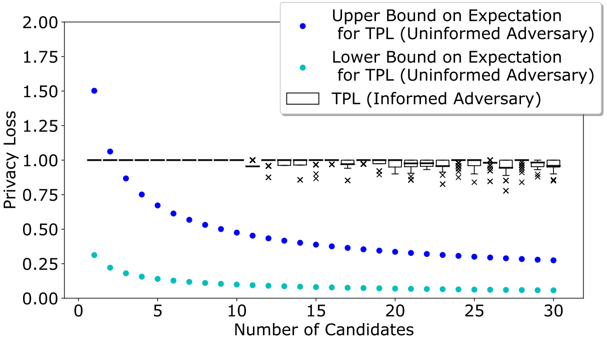

Figure 2: Upper and lower bounds for the expectation of total privacy loss (TPL) for an uninformed attack and a box-plot of TPL for an informed attack (when s is considered).

Our results illustrate a mild decrease, on average, for the informed TPL. However, the decrease is less steep than the two bounds. For relatively large candidate sets, the TPL is close to 1, so job privacy is indeed under threat.

Moreover, we note that the values of TPL are typically significantly above the upper bound on the expected loss of the uninformed attack. This observation validates our intuition that uninformed attacks are much less likely to lead to privacy loss than informed attacks.

Lastly, we see that TPL does not decrease dramatically as the candidate set grows. This behavior is due to the use of the maximum function to aggregate the local privacy losses of the jobs and features when computing the TPL.

When testing on larger domains, we consistently observed a significant privacy loss. The shape of privacy loss decline rate remained the same as in Figure 2 (bounded by the theoretical results). The efficiency of the attack often decreased as the attacker needs to solve larger problems.

Conclusion

We considered a setting where a published schedule may expose private information. To explore privacy attacks on schedules we started by formulating a threat model for published schedules. We defined several metrics for privacy loss and provided theoretical properties of these measures for uninformed attacks. These properties can be used to derive reasonable benchmarks for assessing published schedules. Then, we formulated a framework for informed attacks on schedules using the notion of inverse scheduling. We demonstrated the framework on a single-machine scheduling problem, and proved the computational complexity of the resulting inverse scheduling problem. Using synthetically generated schedules we demonstrated the effectiveness of informed attacks. In future work, we plan to explore privacy protection mechanisms that would be able to minimize the chances of successful inverse scheduling attacks. Moreover, we aim to show theoretical properties of informed ISP attacks and devise sampling schemes to make these attacks more efficient, which can be useful to efficiently test the effectiveness of protection mechanisms.

[1] https://github.com/samadeusfp/aaai2023_schedule_privacy

Acknowledgments

We thank Alexander Shleyfman for the fruitful discussion. Stephan Fahrenkrog-Petersen and Matthias Weidlich gratefully acknowledge the support of the Berlin Centre for Consumer Policies (BCCP). This work was supported by the German Federal Ministry of Education and Research (BMBF), grant number 16DII133 (Weizenbaum-Institute).

References

Bilogrevic, I.; Jadliwala, M.; Kumar, P.; Walia, S. S.; Hubaux, J.; Aad, I.; and Niemi, V. 2011. Meetings through the cloud: Privacy-preserving scheduling on mobile devices. *J. Syst. Softw.*, 84(11): 1910–1927.

Brickell, J.; and Shmatikov, V. 2008. The cost of privacy: destruction of data-mining utility in anonymized data publishing. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 70–78.

Brucker, P.; and Shakhlevich, N. V. 2009. Inverse scheduling with maximum lateness objective. *Journal of Scheduling*, 12(5): 475–488.

Dwork, C. 2008. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, 1–19. Springer.

Feller, W. 1945. The fundamental limit theorems in probability. *Bulletin of the American Mathematical Society*, 51(11): 800–832.

Fioretto, F.; Hentenryck, P. V.; and Zhu, K. 2021. Differential privacy of hierarchical Census data: An optimization approach. *Artif. Intell.*, 296: 103475.

Fung, B. C.; Wang, K.; Chen, R.; and Yu, P. S. 2010. Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys (Csur)*, 42(4): 1–53.

Herlea, T.; Claessens, J.; Preneel, B.; Neven, G.; Piessens, F.; and Decker, B. D. 2001. On securely scheduling a meeting. In *IFIP International Information Security Conference*, 183–198. Springer.

Johnson, D. S.; Yannakakis, M.; and Papadimitriou, C. H. 1988. On generating all maximal independent sets. *Information Processing Letters*, 27(3): 119–123.

Kadloor, S.; and Kiyavash, N. 2015. Delay-Privacy Tradeoff in the Design of Scheduling Policies. *IEEE Trans. Inf. Theory*, 61(5): 2557–2573.

Kamath, G. 2015. Bounds on the Expectation of the Maximum of Samples from a Gaussian. Technical report, University of Waterloo.

Kim, S.; and Nembhard, D. A. 2013. Rule mining for scheduling cross training with a heterogeneous workforce. *International Journal of Production Research*, 51(8): 2281–2300.

LeFevre, K.; DeWitt, D. J.; and Ramakrishnan, R. 2006. Mondrian multidimensional k-anonymity. In *22nd International conference on data engineering (ICDE'06)*, 25–25. IEEE.

Li, X.; and Olafsson, S. 2005. Discovering dispatching rules using data mining. *Journal of Scheduling*, 8(6): 515–527.

Machanavajjhala, A.; Kifer, D.; Gehrke, J.; and Venkitasubramaniam, M. 2007. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1): 3–es.

Narayanan, A.; and Shmatikov, V. 2008. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, 111–125. IEEE.

Pinedo, M. 2016. *Scheduling : theory, algorithms, and systems*. Cham : Springer, fifth edition edition. ISBN 9783319265803. Includes bibliographical references and indexes.

Sweeney, L. 2002. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05): 557–570.

Wagner, I.; and Eckhoff, D. 2018. Technical privacy metrics: a systematic survey. *ACM Computing Surveys (CSUR)*, 51(3): 1–38.

Wallace, R. J.; and Freuder, E. C. 2005. Constraint-based reasoning and privacy/efficiency tradeoffs in multi-agent problem solving. *Artificial Intelligence*, 161(1-2): 209–227.

Yang, D.; Qu, B.; and Cudré-Mauroux, P. 2018. Privacy-preserving social media data publishing for personalized ranking-based recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 31(3): 507–520.

Zhang, L.; Li, Z.; Królczyk, G.; Wu, D.; and Tang, Q. 2019. Mathematical modeling and multi-attribute rule mining for energy efficient job-shop scheduling. *Journal of Cleaner Production*, 241: 118289.