Privacy-utility trade-off under continual observation

Murat A. Erdogdu
Department of Statistics
Stanford University, CA 94305 USA
e-mail: erdogdu@stanford.edu

Nadia Fawaz
Technicolor
Los Altos, CA 94022 USA
e-mail: nadia.fawaz@technicolor.com

***Abstract*—In the online setting, a user continuously releases a time-series that is correlated with his private data, to a service provider to derive some utility. Due to correlations, the continual observation of the time-series puts the user at risk of inference attacks against his private data. To protect the user's privacy, the time-series is randomized prior to its release according to a probabilistic privacy mapping. This mapping should be designed in a way that balances privacy and utility requirements over time. First, we formalize the framework for the design of utility-aware privacy mappings for time-series, under both online and batch models. We introduce two threat models, for which we respectively show that under the log-loss cost function, the information leakage can be modeled by the mutual or directed information between the randomized time-series and the private data. Second, we prove that the design of the privacy mapping can be cast as a convex optimization. We provide a sequential online scheme that allows to design privacy mappings at scale, that accounts for privacy risk from the history of released data and future releases to come. Third, we prove the equivalence of the optimal mappings under the batch and the online models, in the case of a Hidden Markov Model. Evaluations on real-world time-series data show that smart-meter data can be randomized to prevent disaggregation of per-device energy consumption, while maintaining the utility of the randomized series.**

I. Introduction

In the era of the Internet of Things, the collection of fine-grained time-series data raises privacy concerns, as such data is often correlated with sensitive information that users would like to keep private. Studies have shown that private information such as household composition, user behavior and lifestyle (appliance use, eating and sleeping patterns, presence or absence, household activities) [1], health status, or mobility patterns can be inferred from data collected by sensors in houses, business offices or cars, such as smart-meters, HVAC systems (NEST), temperature, light, or motion sensors; by health-monitoring devices (fitbit, jawbone); and by sensors on handheld devices such smartphones, tablets, game controllers. The entity collecting user data may make this data available to third parties with or without user knowledge or the possibility to opt-out. The trust boundary may lie either centrally at the aggregator side: the user may trust the entity aggregating data but not the third-parties with whom the aggregator may share data; or locally at the user side: the user may not entirely trust the aggregating entity in the first place, and may want to limit the amount of private information leaked by the data he releases to the aggregator. Data distortion was proposed as a countermeasure to protect user privacy in both cases: either locally at the user side by randomized response of the user data prior to its release, or in a centralized manner at the aggregator side by randomizing the answer to a query over a user database. In either case, the design of the distortion mechanism should satisfy formal privacy guarantees, but also maintain utility of the distorted data. Initially, data distortion approaches were devised for the static case, and when they were subsequently extended to the dynamic case of time-series, scalability challenges arose. First, as the sequence of distorted releases carries correlation across time, the amount of distortion introduced by the randomization procedure may grow with the sequence length, thus maintaining utility often becomes challenging. Second, the distortion mechanism balancing the privacy-utility trade-off over time is often obtained through algorithms or optimizations whose complexity may scale with the length of the sequence.

Contributions: We consider the online setting where a user would like to continuously release a time-series of data that is correlated with his private data, to a service provider to derive some utility. To protect the user from inference attacks on his private data, samples from the time-series are sequentially randomized prior to their release according to a stochastic process, called the privacy mapping. The privacy mapping should be designed in a way that balances privacy and utility requirements over time. Our contributions are threefold. First, we formalize the framework for the design of utility-aware privacy mappings for time-series, under both online and batch models. Our framework for time-series builds on and generalizes the static framework for privacy against statistical inference [2] to account for temporal correlations in the time-series, and for multiple sequential releases of data. We introduce two threat models, namely the adaptive and the instantaneous inference attacks, for which we respectively show that under the log-loss cost function, the information leakage can be modeled by the mutual information or the directed information between the randomized time-series and the private data. Second, we prove that the design of the privacy mapping can be cast as a convex optimization. We provide a sequential scheme that allows to design online privacy mappings at scale, that accounts for privacy risk from the history of released data and future releases to come. Third, we prove the equivalence of the optimal mappings under the batch and the online models, in the case where the time-series follow a Hidden Markov Model (HMM). Finally, evaluations over real-world time-series data show that smart-meter data can be randomized for privacy purposes to prevent disaggregation of per-device consumption, while maintaining the utility of the randomized series.

Related Work: The problem of preserving *differential privacy* when an analyst continually tracks statistics over a time-series [3] was studied for running sum of bits, for decayed sums of predicates, and for aggregate-sum queries over the time-series data of multiple users. However, these approaches do not account for temporal correlations that may exist between samples of the time-series. Moreover, in this paper, the quantity that is locally randomized prior to the release to the service provider is not an aggregate quantity over multiple users or over multiple time instants, on the contrary to aggregate queries in the centralized differential privacy setting for either user-level or event-level privacy under continual observation.

We locally distort and release an individual time-series of an individual privacy-conscious user. Approaches to protect user privacy for the specific case of smart-meter data [4] include battery-based solutions, data distortion [5], and cryptographic protocols. Privacy-utility tradeoffs for smart-meter data were studied in [5] under an information-theoretic framework. Assuming a stationary Gaussian Markov model, their privacy mapping is designed and applied offline once over the whole sequence prior to the release, and the privacy guarantees hold asymptotically through a single asymptotic constraint on the equivocation rate. In contrast, our approach considers the online setting where data is distorted and released sequentially under a sequence of constraints on the private information leakage at each time, and it is applicable to any stochastic model for the time-series.

II. A GENERAL FRAMEWORK FOR TIME-SERIES DATA

$[T]$ is the set of integers $\{1, \dots T\}$, and $X^T = \{X_1, X_2, \dots X_T\}$ a sequence of T random variables.

A. Setting and challenges

We consider the dynamic setting where at every time instant $t \in [T]$, a privacy-conscious user generates samples from two time-series of data: a sample $S_t \in \mathcal{S}$ of sensitive data that the user would like to keep private, and another sample of data $X_t \in \mathcal{X}$ that the user is willing to release to a service provider, to receive some utility. Assuming that the time-series $S^T = \{S_t\}_{t=1}^T$ and $X^T = \{X_t\}_{t=1}^T$ are correlated, the sequential observation of the samples from X^T by the service provider might allow him to adversarially perform inference attacks on the private sequence S^T. As a countermeasure to protect the user's privacy, the time-series X^T is not released as such, but is distorted according to a stochastic process called the *privacy mapping*, to generate a new time-series $\hat{X}^T = \{\hat{X}_t\}_{t=1}^T$, from which the user will sequentially release samples to the adversarial service provider. The privacy mapping should be designed in a way that balances privacy and utility requirements over time: the time-series should be altered dynamically in a way that renders inference attacks against the private sequence S^T harder at any instant, but not to the extent where the alteration would hinder extracting some utility from the distorted data.

In the dynamic setting of time-series, a natural question arises as to whether the privacy mapping can be designed and operated sequentially as data is generated online, or whether a batch scheme that designs and operates the mapping based on buffered sequences of data would be preferable. More precisely, the privacy mapping can be designed and can operate according to either of the following schemes:

Batch scheme: the batch privacy mapping is produced at time $t = T$ by an algorithm that generates a single joint distribution for the random vector $\hat{X}^T$ based on all the information available until time T (after observing all T samples).

Online scheme: the privacy mapping is produced sequentially by an online algorithm that at every time $t \in [T]$ generates a distribution for $\hat{X}_t$ based on all or a subset of the information available up to time t. The online privacy mapping thus consists of a sequence of distributions. Online schemes can be further categorized as *interactive* or *non-interactive*. An interactive scheme refers to an online scheme that at time t, leverages all the information $(\hat{X}^{t-1}, X^t, S^t)$ available up to time t to generate $\hat{X}_t$, whereas in the non-interactive scheme the distorted data is generated based only on current (X_t, S_t).

B. Threat Model

We define two inference attack models.

Adaptive inference attack: Under the adaptive model, at each time $t \in [T]$, the adversary selects a joint distribution $q \in \mathcal{P}_{S^t}$ on the whole sequence S^t, in order to minimize the average inference cost $C(S^t, q)$ at time t. If the adversary had not observed $\hat{X}^t$, he would choose q as the solution to

$$C_0(t)^* = \min_{q \in \mathcal{P}_{S^t}} \mathbb{E}_{S^t}[C(S^t, q)]. \tag{1}$$

However, after observing the sequence $\hat{X}^t = \hat{x}^t$, the adversary would chose q as the solution to the minimization

$$C_{\hat{X}^t}(t)^* = \min_{q \in \mathcal{P}_{S^t}} \mathbb{E}_{S^t|\hat{X}^t}[C(S^t, q)|\hat{X}^t = \hat{x}^t]. \tag{2}$$

At time t, the average gain in inference cost by the adversary after observing the sequence $\hat{X}^t = \hat{x}^t$ is thus

$$\Delta C(t) = C_0(t)^* - \mathbb{E}_{\hat{X}^t}[C_{\hat{X}^t}(t)^*]. \tag{3}$$

The inference cost gain $\Delta C(t)$ at time $t \in [T]$ represents how much the quality of the inference of the private sequence S^t improves thanks to the observation of the sequence $\hat{X}^t$.

In this model, at each time t, the adversary improves his inference of the whole sequence S^t. In particular, he improves his inference with respect to the previous time $t-1$ in which he only carried out an inference attack on S^{t-1} using $\hat{X}^{t-1}$. At time t, not only does the adversary use all observations $\hat{X}_1, \dots, \hat{X}_t$ up to time t to infer the latest private sample S_t, but he also uses the observation of the latest sample $\hat{X}_t$ along with the past sequence $\hat{X}^{t-1}$ to revise his inference on the past $S_1, \dots, S_{t-1}$. Thus at each time t, the inference of any past private data S_i, $i \in [t]$ is revised using its future $\hat{X}_{i+1}, \dots, \hat{X}_t$, contemporary $\hat{X}_i$ and past $\hat{X}_1, \dots, \hat{X}_{i-1}$ observations.

Instantaneous inference attack: In the instantaneous model, at each time $t \in [T]$, the adversary selects a marginal distribution $q \in \mathcal{P}_{S_t}$ over the variable S_t, in order to minimize the average instantaneous inference cost $C(S_t, q)$ at time t. If the adversary had not observed $\hat{X}^t$, he would choose q as the solution to the minimization

$$c_0(t)^* = \min_{q \in \mathcal{P}_{S_t}} \mathbb{E}_{S^t}[C(S_t, q)], \tag{4}$$

and the resulting aggregate inference cost up to time t is then

$$C'_0(t)^* = c_0(t)^* + C'_0(t-1)^* = \sum_{i=1}^{t} c_0(i)^*. \tag{5}$$

After observing the sequence $\hat{X}^t = \hat{x}^t$, the adversary would chose q as the solution to the minimization

$$c_{\hat{X}^t}(t)^* = \min_{q \in \mathcal{P}_{S_t}} \mathbb{E}_{S^t|\hat{X}^t}[C(S_t, q)|\hat{X}^t = \hat{x}^t], \tag{6}$$

and the resulting aggregate cost up to time t would be

$$C'_{\hat{X}^t}(t)^* = c_{\hat{X}}(t)^* + C'_{\hat{X}}(t-1)^* = \sum_{i=1}^{t} c_{\hat{X}}(i)^*. \tag{7}$$

At time t, the average gain in inference cost by an adversary performing an instantaneous inference attack, after observing the sequence $\hat{X}^t = \hat{x}^t$, is thus

$$\Delta C(t) = C'_0(t)^* - \mathbb{E}_{\hat{X}^t}[C'_{\hat{X}^t}(t)^*]. \tag{8}$$

The instantaneous model represents an adversary who, at each time t, tries to infer S_t based on his observation of the whole sequence $\hat{X}^t$ up to time t, but who does not have the possibility to later improve the quality of the inference of S_t using observation of future samples $\hat{X}_i$, $i > t$. In particular at time t, the adversary is not allowed to revise and improve his inference of past S^{t-1} using the latest observation $\hat{X}_t$. The instantaneous model represents a weaker adversary than the adaptive model, in terms of quality of the inference for S^T.

C. Privacy Metric

Under both the adaptive and the instantaneous attacks, the inference cost gain $\Delta C(t)$ at time t represents how much the quality of the inference of the private sequence S^t improves thanks to the observation of sequence $\hat{X}^t$. Thus, $\Delta C(t)$ as defined in either Eq. (3) or (8) will be used as a privacy metric, representing the private *Information Leakage* up to time t.

Definition 1. *The* Information Leakage *from $\hat{X}^t$ to S^t is defined as $\mathcal{J}(\hat{X}^t; S^t) = \Delta C(t)$. It quantifies the improvement in the inference of S^t after observing $\hat{X}^t$.*

Def. 1 captures a broad class of adversaries performing inference attacks on time-series, under either the adaptive or the instantaneous model.

Definition 2. *A sequence $X^T \in \mathbb{R}^T$ is ϵ^T-*private *with respect to a sequence S^T if $\forall t \in [T]$, the information leakage at time t is bounded by ϵ_t, i.e., $\forall t \in [T]$, $\quad \mathcal{J}(X^t; S^t) \leq \epsilon_t$.*

Def. 2 constrains the information leakage at each time $t \in [T]$. This is in contrast with prior works, [5] which constrain the average equivocation rate asymptotically as the sequence size grows large. The sequence of privacy constraints ϵ^T can be specified by the user or a privacy agent on his behalf.

D. Distortion Metric

The distortion metric $d : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ quantifies the proximity of the distorted sequence $\hat{X}^t$ to the original data X^t. We will assume that the distortion metric is separable: $d(X^t, \hat{X}^t) = \frac{1}{t}\sum_{\tau=1}^{t} d(X_\tau, \hat{X}_\tau)$. Examples of separable metrics include Hamming distance, and l_p-norms to the power p.

E. Privacy-Utility trade-off for time-series

The design of the privacy mapping should aim at minimizing the expected distortion between X^T and $\hat{X}^T$, $d(X^T, \hat{X}^T)$, while enforcing a privacy constraint ϵ_t at each time t, thus balancing the privacy-utility tradeoff over time. That is, the privacy mapping should generate a distorted version $\hat{X}^T$ of X^T which is ϵ^T-private and close to the original sequence X^T.

Batch Scheme: The batch scheme, shown in Alg. 1, minimizes the distortion between X^T and $\hat{X}^T$ over the joint distribution $p(\hat{x}^T|x^T, s^T)$ under T privacy constraints in a single run. It requires as input the joint distribution $p(x^T, s^T)$, since both the objective and the constraints can be written as functions of the input $p(x^T, s^T)$ and the variable $p(\hat{x}^T|x^T, s^T)$.

Online Scheme: The online scheme, shown in Alg. 2, operates sequentially by leveraging at time t all that has been accomplished by past steps from time 1 to time $t-1$, as illustrated in Fig. 1. More precisely, at every time $t \in [T]$, the online scheme minimizes the distortion between X_t and $\hat{X}_t$ under privacy constraint $J(\hat{X}^t; S^t) \leq \epsilon_t$ over the distributions $p(\hat{x}_t|x^t, s^t, \hat{x}^{t-1})$, which is conditioned over the history of past samples x^t, s^t and past randomization outputs $\hat{x}^{t-1}$. Moreover, at step t, the online scheme requires the joint distribution $p(\hat{x}^{t-1}, x^t, s^t)$ as an input. This is obtained in Eq. (10) by combining $p(\hat{x}^{t-1}, x^{t-1}, s^{t-1})$, which is an output from the previous step $t-1$, and $p(x_t, s_t|x^{t-1}, s^{t-1})$, which is assumed to be known at step t, and by using the conditional independence of $\hat{x}^{t-1}$ and (x_t, s_t) conditioned on (x^{t-1}, s^{t-1}). Finally, by definition, the information leakage at time t, $J(\hat{X}^t; S^t)$ measures the leakage of the whole sequence $\hat{X}^t$ with respect to the whole sequence S^t, and thus incorporates leakages due to earlier releases of distorted data $\hat{X}^{t-1}$. Thus, the constraint at time t both accounts for the leakage up to time $t-1$ and bounds the incremental leakage due to step t.

Definition 3. *The* regret *between the online and batch schemes is the difference between the optimal distortions they achieve.*

III. Analysis of the batch and online schemes

A. Information Leakage under the log-loss

We focus on the information leakage under the log-loss cost $C(s, q) = -\log q(s)$. The relevance and generality of the log-loss cost in the privacy metric were justified in [2], [6].

Lemma III.1. *Assuming an adaptive inference attack, the information leakage at time t under the log-loss cost function $C(s^t, q) = -\log q(s^t)$ is given by the mutual information between sequences S^t and $\hat{X}^t$, i.e., $\mathcal{J}(\hat{X}^t; S^t) = I(\hat{X}^t; S^t)$.*

Mutual information has been previously introduced to quantify information leakage in the static setting in [2]. A related metric, the equivocation rate $\frac{1}{t}H(S^t|\hat{X}^t)$, was used in [7], [8] to quantify the level of privacy in the asymptotic regime of large sequences under a batch scheme.

Lemma III.2. *Assuming an instantaneous inference attack, the information leakage at time t under the log-loss cost function $C(s_t, q) = -\log q(s_t)$ is given by the directed information [9] from the sequence $\hat{X}^t$ to the sequence S^t, i.e. $\mathcal{J}(\hat{X}^t; S^t) = I(\hat{X}^t \to S^t) = H(S^t) - H(S^t\|\hat{X}^t)$.*

$H(S^t\|\hat{X}^t)$ denotes the *causally conditional entropy* [10]. The directed information [9] $I(\hat{X}^t \to S^t)$ from sequence $\hat{X}^t$ to sequence S^t is an asymmetric measure of how much $\hat{X}^t$ is relevant for temporally causal inference of S^t. Temporal causality here does not mean causation, but how past samples $\hat{X}^t$ up to time t affect the inference of elements of S^t, while future samples $\hat{X}_i$, $i > t$ are not relied upon to infer S^t.

B. Convexity of the optimizations

Theorem III.3. *Assuming finite alphabets $\mathcal{S}, \mathcal{X}, \widehat{\mathcal{X}}$, and that the information leakage metric is either mutual or directed information, then Optimizations (9) and (11) are convex.*

Theorem III.3 (proof in appendix) allows for the use of efficient convex optimization techniques. However, without any model assumption, the number of variables of the convex program grows exponentially with T. Simplifying model assumptions, such as HMM or time-window dependency, allow to decrease the problem size. For instance, for independent samples, the size of the online problem scales linearly with T.

Algorithm 1 Batch scheme $\mathcal{A}_b$

Input: $p(x^T, s^T)$, ϵ^T

Solve optimization $\mathcal{A}_b(T)$**:**

$$p^*(\hat{x}^T|x^T, s^T) = \underset{p(\hat{x}^T|x^T,s^T)}{\operatorname{argmin}} \quad \mathbb{E}_{X^T,\hat{X}^T}[d(X^T, \hat{X}^T)]$$
$$\text{subject to:} \quad \mathcal{J}(\hat{X}^t; S^t) \leq \epsilon_t, \ \forall t \in [T] \tag{9}$$

Output: $p^*(\hat{x}^T|x^T, s^T)$

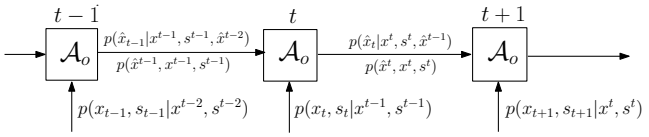

Fig. 1: Sequential structure of the online scheme.

Algorithm 2 Online scheme $\mathcal{A}_o$

for all $t \in [T]$ **do**

Input: $p(x_t, s_t|x^{t-1}, s^{t-1})$, $p'(\hat{x}^{t-1}, x^{t-1}, s^{t-1})$, ϵ_t

Update:

$$p(\hat{x}^{t-1}, x^t, s^t) = p(x_t, s_t|x^{t-1}, s^{t-1})p'(\hat{x}^{t-1}, x^{t-1}, s^{t-1}) \tag{10}$$

Solve optimization $\mathcal{A}_o(t)$**:**

$$p^*(\hat{x}_t|x^t, s^t, \hat{x}^{t-1}) = \underset{p(\hat{x}_t|x^t,s^t,\hat{x}^{t-1})}{\operatorname{argmin}} \quad \mathbb{E}_{X_t,\hat{X}_t}[d(X_t, \hat{X}_t)]$$
$$\text{subject to:} \quad \mathcal{J}(\hat{X}^t; S^t) \leq \epsilon_t \tag{11}$$

Update:

$$p'(\hat{x}^t, x^t, s^t) = p(\hat{x}^{t-1}, x^t, s^t)p^*(\hat{x}_t|x^t, s^t, \hat{x}^{t-1}) \tag{12}$$

Output: $p^*(\hat{x}_t|x^t, s^t, \hat{x}^{t-1})$, $p'(\hat{x}^t, x^t, s^t)$

C. Regret under a Hidden Markov Model

The schemes described in Eq.(9-11) do not make any model assumptions. Depending on the application and the corresponding model, the described method might be significantly simplified. Two simple and notable cases are the *Independent* and the HMM. In this section, we assume a generalized HMM, depicted in Fig. 2. In this model, there is a Markov relation between the hidden states, representing the private data $\{S_t\}_{t=1}^T$, and X_t is independent from any other state conditioned on S_t, S_{t-1}. This model allows a flexible dependency structure for the current data X_t with respect to the past private data.

Theorem III.4. *Let the information leakage metric $\mathcal{J}$, be either mutual or directed information. For a given batch problem with privacy levels $\{\epsilon_t\}_{t=1}^T$, if the sequence of random pairs $\{S_t, X_t\}_{t=1}^T$ satisfies the HMM model in Fig. 2, then there exists a choice of privacy levels $\{\epsilon'_t\}_{t=1}^T$ for the online problem resulting in no regret.*

Assuming further that the random triplets $\{x_t, s_t, s_{t-1}\}_{t=1}^T$ are identically distributed and the increments of privacy levels, $\{\delta_t = \epsilon_t - \epsilon_{t-1}\}_{t=1}^T$, are non-decreasing, then the online and the batch problems are equivalent for the same choice of privacy levels, resulting in no regret.

Theorem III.4 (proof in appendix) states that the online and the batch schemes are the same and that there is no regret under a general HMM assumption. If we further assume independence between time points, the online scheme reduces to a non-interactive one. Ongoing work includes deriving regret bounds when increments are not increasing.

IV. Experiments on smart-meter dataset

REDD dataset: The Reference Energy Disaggregation Data Set (REDD) [11], [12] consists of the power consumption of 6 houses. For each house, the power consumption of each appliance in the house is available every 3 seconds. For a given house, the *aggregate load* of that house at time t is defined as the total power consumption of all the appliances at that time. In our experiments, for each house, the aggregate load data is split into a training set (90% data) and a test set (10% data).

The collection of aggregate load data from a house at a fine-grained time-scale presents privacy risks to the household members. Research in the field of Non-Intrusive Load Monitoring (NILM) [12] has shown that aggregate loads can be disaggregated with high fidelity, and that the per-device consumption at every instant can be recovered. Consequently, the aggregate load can be used to make inferences on the household private information, including its occupancy, sleeping and eating patterns of its members, their health status.

Experimental setting: The scenario for the experiments involves a household, and a service provider, who may behave adversarially. The service provider offers some utility to the household, which requires inference of the state of washer-dryer in the house from the aggregate load. To provide utility to the user, for instance automated control of the washer/dryer, the service provider needs process the aggregate load data received the user. The utility U_t represents the outcome of the service provider's algorithm that runs on the released data. In this experiment, U_t will be the result of the inference on the state of *washer-dryer* from the load data.

The household is willing to give the aggregate load X_t to the service provider, but wishes to keep the information related to their eating patterns private, in particular the microwave usage which can also be inferred from the aggregate load. In this experiment, we choose the private information S_t to be the state of the *microwave* (ON or OFF). If the user released X_t as is, it can be used adversarially by the service provider to infer information regarding S_t, thus raising privacy concerns. The user will instead release a distorted version $\hat{X}_t$. Examples of adversarial providers include third-parties, such as apps, to whom the company operating the smart-meter may give access to the data it collects, or a malicious insider such as a curious employee. Our goal is to get the utility related to the washer-dryer, while keeping the sensitive information regarding microwave usage private. The dataset provides ground truth for both the microwave and washer-dryer state, which allows us to verify the performance of our approach. Note that neither the private data nor the utility are

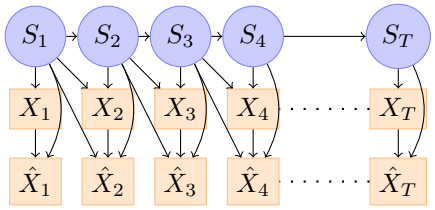

Fig. 2: HMM dependency graph for online or batch schemes

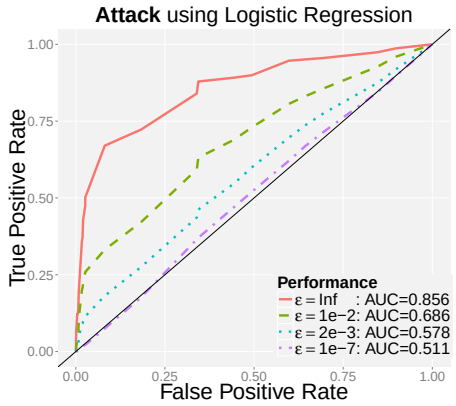

Fig. 3: ROC for inference attack on private data (microwave)

Fig. 4: ROC for utility inference (washer-dryer)

limited to components of the aggregate load, and could be any information correlated with the aggregate load.

Training phase: Each day is broken into 24 hourly periods. For each household, and for each hourly period, the empirical joint distributions for (S_t, X_t) and (U_t, X_t), corresponding respectively to the pairs of private data and aggregate load, and of utility data and aggregate load, are obtained using the training set. Thus 24 joint distributions are trained for each pair. The joint distributions for (S_t, X_t) are assumed to be available to both the adversarial provider, and to the privacy agent acting on behalf of the household, while the joint distribution for the pair (U_t, X_t) is assumed to be known only by the service provider. This worst-case setting assumes that the adversarial service provider has full knowledge of the joint statistics of the private data and the aggregate load. This might be the case in reality, as the adversarial provider might have data collected from a different but similar household, or have historical aggregate load data for this given household prior to the activation of privacy protection.

Using the training set, the adversarial provider trains models for inference of utility U_t from the aggregate load, as well as for inference attack on private S_t from the aggregate load. Our experiments evaluated several models for both utility and private data inference, including Maximum a Posteriori, logistic regression based on X_t, or regressions based on feature vectors. For the sake of conciseness, results are only presented for logistic regression, as other models led to similar results. Using the training set, the privacy agent trains a non-interactive privacy mapping for each hourly period, using the online scheme in Eq. (11), under the HMM setting in Fig. 2.

Test phase: The privacy agent uses the privacy mapping to generate the distorted loads $\{\hat{X}_t\}_{t=1}^T$. Privacy leakage levels are set to $\epsilon_t = t\epsilon$. In the non-private case, where the user sends the aggregate load itself, the leakage level is $\epsilon = \text{Inf}$.

For a leakage level ϵ, we input the distorted aggregate load to the previously trained models. Fig. 3 shows ROC curves of the inference attack on the microwave, while Fig. 4 shows the utility inference on the washer-dryer, both using logistic regression. The *Area Under the Curve* (AUC) illustrates intrusion level and prediction quality. As the leakage level ϵ decreases, inference attack quality degrades, whereas the quality of utility inference remains unchanged. For a small enough ϵ (purple curve), the inference attack curve becomes close to diagonal, implying that the inference algorithm does not outperform a random guess. Indeed, when ϵ goes to zero, the distorted output becomes independent from the private data.

V. Conclusion

We propose an online and a batch scheme for the design of utility-aware privacy mappings, in the setting where a user continuously releases a time-series that is correlated with his private information, to a service provider in the hope of deriving some utility from this release. These general schemes can be adapted to any model assumption suitable for a given application. We prove that the schemes can be cast as convex optimizations. Under an HMM assumption, we show that the outputs of the online and the batch schemes are the same, thus there is no regret. Experiments on a smart-meter dataset show that leakage can be bounded over time while maintaining utility of the distorted data. Other applications may include privacy for time-series data from health-monitoring devices, sensors in houses, offices, cars or handheld devices.

Acknowledgment

The authors would like to thank Andrea Montanari, for his insightful feedback and his very valuable suggestions.

References

[1] M. A. Lisovich, D. K. Mulligan, and S. B. Wicker, "Inferring personal information from demand-response systems," *IEEE Security and Privacy*, 2010.

[2] F. du Pin Calmon and N. Fawaz, "Privacy against statistical inference," in *Allerton*, 2012.

[3] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential privacy under continual observation," in *ACM STOC*, 2010.

[4] M. Jawurek, F. Kerschbaum, and G. Danezis, "Privacy technologies for smart grids - a survey of options," Tech. Rep. MSR-TR-2012-119, 2012.

[5] L. Sankar, S. R. Rajagopalan, S. Mohajer, and H. V. Poor, "Smart meter privacy: A theoretical framework," *IEEE Trans. Smart Grid*, 2013.

[6] A. Makhdoumi, S. Salamatian, N. Fawaz, and M. Médard, "From the information bottleneck to the privacy funnel," *ITW*, 2014.

[7] H. Yamamoto, "A source coding problem for sources with additional outputs to keep secret from the receiver of wiretappers," *IEEE Trans. Inf. Theory*, 1983.

[8] L. Sankar, S. R. Rajagopalan, and H. V. Poor, "Utility-privacy tradeoff in databases: An information-theoretic approach," *IEEE Trans. Inf. Forensics Security*, 2013.

[9] H. Marko, "The bidirectional communication theory–a generalization of information theory," *IEEE Trans. Commun.*, 1973.

[10] G. Kramer, "Directed information for channels with feedback," Ph.D. dissertation, University of Manitoba, Canada, 1998.

[11] J. Z. Kolter and M. J. Johnson, "Redd: A public data set for energy disaggregation research," 2011, http://redd.csail.mit.edu/.

[12] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation," in *AISTATS*, 2012.

APPENDIX A
PROOFS

We state all the proofs in the Appendix. We start with a simple decomposition property used several times in the proofs.

Lemma A.1 (Decomposition rules)**.**

$$\begin{aligned} I(S^t;\hat{X}^t) =& I(S^{t-1};\hat{X}^{t-1}) \\ &+ I(\hat{X}^{t-1};S_t|S^{t-1}) + I(\hat{X}_t;S^t|\hat{X}^{t-1}) \end{aligned} \tag{13}$$

$$I(\hat{X}^t \to S^t) = I(\hat{X}^{t-1} \to S^{t-1}) + I(\hat{X}^t;S_t|S^{t-1}). \tag{14}$$

The proof of LemmaA.1 is omitted as it follows straightforward from the definitions of the mutual and directed information.

A. Proof of Theorem III.3

Proof of Theorem III.3: We will prove the convexity of the online and the batch problems for the case where the information leakage metric is mutual information. The proof for directed information use similar arguments.

We start with the batch problem. Denote by $\hat{x}^{T-\{t\}}$, the sequence $\{\hat{x}_1,...,\hat{x}_{t-1},\hat{x}_{t+1},...,\hat{x}_T\}$. Clearly, the batch problem in Eq.(9) can be written as,

$$\begin{aligned} &\underset{\text{the probability simplex}}{\text{minimize over}} \quad \sum_{t=1}^{T}\sum_{\substack{\hat{x}_t\in\widehat{\mathcal{X}},\\ x_t\in\mathcal{X}}} p(\hat{x}_t|x_t)p(x_t)d(x_t,\hat{x}_t) \\ &\text{subject to:} \sum_{\substack{\hat{x}^t\in\widehat{\mathcal{X}}^t,\\ s^t\in\mathcal{S}^t}} p(\hat{x}^t|s^t)p(s^t)\log\left(\tfrac{p(\hat{x}^t|s^t)}{p(\hat{x}^t)}\right)\le\epsilon_t, \\ &\qquad\qquad \forall t\in[T]; \\ &\sum_{\substack{\hat{x}^{T-\{t\}}\in\widehat{\mathcal{X}}^{T-1},\\ s^T\in\mathcal{S}^T}} p(\hat{x}^T|x^T,s^T)p(x^{T-\{t\}},s^T|x_t) = p(\hat{x}_t|x_t), \\ &\qquad\qquad \forall t\in[T],\forall\hat{x}_t\in\widehat{\mathcal{X}},\forall x_t\in\mathcal{X}; \\ &\sum_{\substack{\hat{x}^T_{t+1}\in\widehat{\mathcal{X}}^{T-t},\\ s^T_{t+1}\in\mathcal{S}^{T-t},\ \forall x^T\in\mathcal{X}^T}} p(\hat{x}^T|x^T,s^T)p(x^T,s^T_{t+1}|s^t) = p(\hat{x}^t|s^t), \\ &\qquad\qquad \forall t\in[T],\forall\hat{x}^t\in\widehat{\mathcal{X}}^t,\forall s^t\in\mathcal{S}^t; \\ &\sum_{s^t\in\mathcal{S}^t} p(\hat{x}^t|s^t)p(s^t) = p(\hat{x}^t), \\ &\qquad\qquad \forall t\in[T],\forall\hat{x}^t\in\widehat{\mathcal{X}}^t. \end{aligned}$$

The optimization variables are the probabilities $p(\hat{x}_t|x_t)$, $p(\hat{x}^t|s^t)$, $p(\hat{x}^t)$, $\forall t\in[T]$ and $p(\hat{x}^T|x^T,s^T)$. Notice that the function $(x,y)\to x\log(x/y)$ is convex in the pair (x,y). Hence, the inequality constraints are just summations of convex functions. The other constraints are affine equality constrains between the optimization variables. Further, the objective is a linear function of variables. Therefore, we conclude that the batch problem is in fact convex.

Next, we show that the online algorithm can be written as a convex program. For this, we write the problem in Eq.(11) explicitly, for each t. Notice that, at time t, $p(\hat{x}^{t-1},x^t,s^t)$ is assumed to be known due to the sequential nature of the algorithm which we described in Section II. Therefore, for the online problem, we write,

$$\begin{aligned} &\underset{\text{the probability simplex}}{\text{minimize over}} \quad \sum_{\substack{\hat{x}_t\in\widehat{\mathcal{X}},\\ x_t\in\mathcal{X}}} p(\hat{x}_t|x_t)p(x_t)d(x_t,\hat{x}_t) \\ &\text{subject to:} \sum_{\substack{\hat{x}^t\in\widehat{\mathcal{X}}^t,\\ s_t\in\mathcal{S}^t}} p(\hat{x}^t|s^t)p(s^t)\log\left(\tfrac{p(\hat{x}^t|s^t)}{p(\hat{x}^t)}\right)\le\epsilon_t; \\ &\sum_{\substack{\hat{x}^{t-1}\in\widehat{\mathcal{X}}^{t-1},\\ x^{t-1}\in\mathcal{X}^{t-1},\\ s^t\in\mathcal{S}^t}} p(\hat{x}_t|x^t,s^t,\hat{x}^{t-1})p(\hat{x}^{t-1},x^{t-1},s^t|x_t) \\ &\qquad\qquad = p(\hat{x}_t|x_t), \\ &\qquad\qquad \forall\hat{x}_t\in\widehat{\mathcal{X}},\forall x_t\in\mathcal{X}; \\ &\sum_{\forall x^t\in\mathcal{X}^t} p(\hat{x}_t|x^t,s^t,\hat{x}^{t-1})p(\hat{x}^{t-1},x^t|s^t) = p(\hat{x}^t|s^t), \\ &\qquad\qquad \forall\hat{x}^t\in\widehat{\mathcal{X}}^t,\forall s^t\in\mathcal{S}^t; \\ &\sum_{s^t\in\mathcal{S}^t} p(\hat{x}^t|s^t)p(s^t) = p(\hat{x}^t), \qquad \forall\hat{x}^t\in\widehat{\mathcal{X}}^t. \end{aligned}$$

The optimization variables are $p(\hat{x}_t|x_t)$, $p(\hat{x}^t|s^t)$, $p(\hat{x}^t)$ and $p(\hat{x}_t|x^t,s^t,\hat{x}^{t-1})$. By the same argument in the batch case, we conclude that the above problem is convex.

The proofs for which the information leakage metric is set to directed information, follow from the similar steps. That is, by the Lemma A.1, we write the information constraint as

$$\begin{aligned} I(\hat{X}^t\to S^t) &= \sum_{\tau=1}^{t} I(\hat{X}^\tau;S_\tau|S^{\tau-1}), \\ &= \sum_{\tau=1}^{t}\sum_{\substack{\hat{x}^\tau\in\widehat{\mathcal{X}}^\tau,\\ s^\tau\in\mathcal{S}^\tau}} p(\hat{x}^\tau|s^\tau)p(s^\tau)\log\left(\frac{p(\hat{x}^\tau|s^\tau)}{p(\hat{x}^\tau|s^{\tau-1})}\right)\le\epsilon_t, \end{aligned}$$

and apply the same argument as before. ∎

B. Proof of Theorem III.4

Proof of Theorem III.4: We first prove the case where the information leakage metric is set to mutual information, that is, $\mathcal{J}(\hat{X}^t;S^t)=I(\hat{X}^t;S^t)$. Let $\mathcal{P}$ denote the set of joint probability measures $p(x,s_2,s_1)$ and define the function $h:\mathbb{R}_+\times\mathcal{P}\to\mathbb{R}$ as

$$\begin{aligned} h(\delta,p(x,s_2,s_1)) = \underset{p(\hat{x}|x,s_2,s_1)}{\text{minimize}}, \quad & \mathbb{E}[d(X,\hat{X})] \\ \text{subject to:} \quad & I(\hat{X};S_2|S_1)\le\delta. \end{aligned} \tag{15}$$

We have the following lemma that will be useful throughout the proof:

Lemma A.2. *For $p=p(x,s_2,s_1)$, the function $h(\delta,p)$ is convex and monotone decreasing in δ.*

Remark 1. *Please note the resemblance of $h(\delta, p)$ to the rate distortion theorem in which we would have X instead of S_2 and no conditioning on S_1 in the information constraint. Proof follows from similar steps.*

Proof of Lemma A.2: Consider the minimization problem stated in Eq.(15) for two different constraints, $\delta_1, \delta_2 \in \mathbb{R}_+$ and, assume that $p_1(\hat{x}|x, s_2, s_1)$ and $p_2(\hat{x}|x, s_2, s_1)$ are the corresponding minimizers, and that $I_{p_1}(\hat{X}, S_2|S_1)$ and $I_{p_2}(\hat{X}, S_2|S_1)$ be the corresponding mutual informations, respectively. For some $\lambda \in [0, 1]$, let $\delta_\lambda = \lambda\delta_1 + (1-\lambda)\delta_2$ and form a new probability measure by linearly combining the above distributions. That is,

$$\begin{aligned}
p_\lambda(\hat{x}|x, s_2, s_1) &= \lambda p_1(\hat{x}|x, s_2, s_1) + (1-\lambda)p_2(\hat{x}|x, s_2, s_1),\\
p_\lambda(\hat{x}|s_2, s_1) &= \sum_{x\in\mathcal{X}} p_\lambda(\hat{x}|x, s_2, s_1)p(x|s_2, s_1)\\
&= \lambda p_1(\hat{x}|s_2, s_1) + (1-\lambda)p_2(\hat{x}|s_2, s_1),\\
p_\lambda(\hat{x}, s_2|s_1) &= p_\lambda(\hat{x}|s_2, s_1)p(s_2|s_1)\\
&= \lambda p_1(\hat{x}, s_2|s_1) + (1-\lambda)p_2(\hat{x}, s_2|s_1),\\
p_\lambda(\hat{x}) &= \sum_{x\in\mathcal{X}, s\in\mathcal{S}} p_\lambda(\hat{x}|x, s_2, s_1)p(x, s_2, s_1)\\
&= \lambda p_1(\hat{x}) + (1-\lambda)p_2(\hat{x}).
\end{aligned}$$

Next, we write the mutual information over the distribution p_λ, using Kullback-Leibler(KL) divergence which is convex in both of its arguments.

$$\begin{aligned}
I_{p_\lambda}(\hat{X}; S_2|S_1) =& D_{\text{KL}}\left(p_\lambda(\hat{x}, s_2|s_1)\|p_\lambda(\hat{x})p(s_2|s_1)\right),\\
\leq& \lambda I_{p_1}(\hat{X}; S_2|S_1) + (1-\lambda)I_{p_2}(\hat{X}; S_2|S_1),\\
\leq& \lambda\delta_1 + (1-\lambda)\delta_2 = \delta_\lambda,
\end{aligned}$$

where we used the convexity of KL divergence. Therefore, p_λ is in the feasible set of $h(\delta_\lambda, p)$ and by definition,

$$h(\delta_\lambda, p) \leq \mathbb{E}_{p_\lambda}[d(\hat{X}, X)].$$

We also have, by linearity of the expectation,

$$\begin{aligned}
h(\delta_\lambda, p) \leq& \mathbb{E}_{p_\lambda}[d(\hat{X}, X)],\\
=& \lambda\mathbb{E}_{p_1}[d(\hat{X}, X)] + (1-\lambda)\mathbb{E}_{p_2}[d(\hat{X}, X)],\\
=& \lambda h(\delta_1, p) + (1-\lambda)h(\delta_2, p).
\end{aligned}$$

Hence we conclude $h(\delta, p)$ is convex in δ.

The monotone nature of $h(\delta, p)$ is due to the dependence of the feasible set on δ. As we increase δ, we also increase the size of the feasible set for $h(\delta, p)$ which implies that $h(\delta, p)$ is a decreasing function of δ. ∎

In the following, we will use the function $h(\delta, p)$ and its properties. Consider the modified version of the batch problem in Eq. (9):

$$\begin{aligned}
\underset{p(\hat{x}^T|x^T, s^T)}{\text{minimize}} \quad & \mathbb{E}[d(X^T, \hat{X}^T)] \qquad (16)\\
\text{subject to:} \quad & \sum_{\tau=1}^{t} I(\hat{X}_\tau; S_\tau|S_{\tau-1}) \leq \epsilon_t,\ \forall t \in [T].
\end{aligned}$$

Note that the above problem has the same objective function but a different feasible set compared to the problem stated in Eq. (9). Now, we write a simple inequality on the mutual information.

$$\begin{aligned}
I(\hat{X}^t; S^t) &= H(S^t) - H(S^t|\hat{X}^t),\\
&= \sum_{\tau=1}^{t} H(S_\tau|S^{\tau-1}) - H(S^t|\hat{X}^t),\\
&= \sum_{\tau=1}^{t} \left[H(S_\tau|S_{\tau-1}) - H(S_\tau|S^{\tau-1}, \hat{X}^t)\right],\\
&\geq \sum_{\tau=1}^{t} I(\hat{X}_\tau; S_\tau|S_{\tau-1}). \qquad (17)
\end{aligned}$$

The third equality follows from the model assumptions and the inequality follows from the fact that conditioning reduces the entropy. The above identity shows that the problem in Eq. (9) has a smaller feasible set compared to the problem in Eq. (16). Therefore, we conclude that the optimal value of Eq. (16) is smaller than or equal to the that of Eq. (9). Let the solution of the modified problem be attained at $p_*(\hat{x}^T|x^T, s^T)$. Then we can obtain the corresponding conditional distributions, $p_*(\hat{x}_t|x_t, s_t, s_{t-1})$, simply by using Bayes' rule and integration. Note that the constraints and the distortion only depends on these conditional distributions through their product measure hence the optimal solution of the modified problem is attainable by optimizing over the product measure

$$p(\hat{x}^T|x^T, s^T) = \prod_{t=1}^{T} p(\hat{x}_t|x_t, s_t, s_{t-1}).$$

By the same argument as before, since the above measure is attainable by the original batch problem, we conclude that the optimal value of the original problem will be attained by this product measure. The resulting dependence structure will be as shown in Figure 2.

For the optimal distribution p_* (or the set of conditional distributions), let $I_{p_*}(\hat{X}_t; S_t) = \delta_t$ for $t = 1, ..., T$. By the definition of $h(\delta, p)$, we know that

$$\sum_{t=1}^{T} h(\delta_t, p(x_t, s_t, s_{t-1})) \leq \sum_{t=1}^{T} \mathbb{E}[d(\hat{X}_t, X_t)].$$

This proves that the solution of the original batch problem will be the same as the product measure formed by the solutions of $h(\delta_t, p(x_t, s_t))$.

Similarly, for the online problem with the privacy levels $\{\sum_{\tau=1}^{t}\delta_\tau\}_{t=1}^{T}$, we can use the inequality in Eq.(17) and a similar argument as above to conclude that, at time t, minimizing over $p(\hat{x}_t|x_t, s_t, s_{t-1})$ will give the optimal result (obviously, we assume that the feasible set is not empty). Now, if we consider the solution of online problem sequentially, we can prove the first part by using an induction argument: Starting from $t = 1$, we observe that the solution is just the optimal solution of $h(\delta_1, p(x_1, s_1))$. Now assume that at time $t-1$, the solutions of the online problem and $h(\delta_{t-1}, p(x_{t-1}, s_{t-1}, s_{t-2}))$ match. Using the algorithm statement given in Eq. (11) and Lemma A.1, it is easy to see that the optimal value of the online problem at time t will be $h(\delta_t, p(x_t, s_t, s_{t-1}))$. Hence the proof of the first part is completed.

For the second part, we define a new sequence, $\{\delta'_\tau\}_{\tau=1}^T$, by

$$\delta'_1 = \epsilon_1,$$
$$\delta'_i = \epsilon_i - \epsilon_{i-1} \quad \text{for } i = 2, ..., T.$$

and we notice that the constructed sequence will be increasing by the given condition, i.e.

$$\delta'_1 \leq \delta'_2 \leq ... \leq \delta'_T.$$

We will show that the problem in Eq. (16) attains its optimum at the constraint boundaries.

First, we consider the case where $\forall t$, $\delta'_t < \delta_*$ where δ_* is defined as

$$\delta_* = \inf\{\delta > 0 : \ h(\delta, p) = 0\}.$$

In this part, since we have the identical distributions on the triples, we denote the joint distributions by $p = p(x_t, s_t, s_{t-1})$, $\forall t$. We showed in the previous part that the batch problem is in fact equivalent to

$$\underset{\delta_1, \delta_2, ..., \delta_T}{\text{minimize}} \quad \sum_{t=1}^{T} h(\delta_t, p) \tag{18}$$

$$\text{subject to} \quad \sum_{\tau=1}^{t} \delta_\tau \leq \sum_{\tau=1}^{t} \delta'_\tau, \ \forall t \in [T], \tag{19}$$
$$\delta_t \geq 0, \ \forall t \in [T].$$

By Lemma A.2, we have $\dot{h}(\delta, p) \leq 0$ and $-\dot{h}(\delta_i, p) \geq -\dot{h}(\delta_j, p) \geq 0$ for $i < j$ where $\dot{h}$ denotes the derivatives that are with respect to the first argument. Assume that for some sequence $\{\alpha_t\}_{t=1}^T$, the problem constraints are satisfied and we have

$$\sum_{\tau=1}^{t} \alpha_\tau \leq \sum_{\tau=1}^{t} \delta'_\tau, \ \forall t \in [T]. \tag{20}$$

For this choice of privacy levels to be the optimum solution, it should provide a smaller objective, i.e.,

$$\sum_{t=1}^{T} h(\delta'_t, p) \geq \sum_{t=1}^{T} h(\alpha_t, p).$$

But, using the convexity of the function h, we can write

$$\sum_{\tau=1}^{t} \{h(\alpha_\tau, p) - h(\delta'_\tau, p)\} \geq \sum_{\tau=1}^{t} \dot{h}(\delta'_\tau, p)(\delta'_\tau - \alpha_\tau).$$

Since $\delta_\tau < \delta_*$, we have $\forall \tau \ \dot{h}(\delta'_\tau, p) > 0$ and let the minimum of those be C. Then the right hand side of the above equation is larger than $C \sum_{\tau=1}^{t} (\delta'_\tau - \alpha_\tau) \geq 0$. Therefore the optimal value is attained at the sequence $\delta'_1, \delta'_2, ..., \delta'_T$.

Now, if for some t, $\delta_* \leq \delta'_t$, since the sequence of δ'_t's is increasing and also by the i.i.d. assumption, we have $\forall \tau \geq t$, $\delta_* < \delta_\tau$ and $h(\delta_\tau, p) = 0$. This means that the constraint has no effect on the minimization problem for $\tau \geq t$ and $\hat{X}_\tau$ will have the same distribution as X_τ resulting in 0 distortion. The optimal solution will be anywhere between δ_* and δ_τ. We can conclude that the optimal value will be attained at the boundary. Since the previous constraints will be attained by the above argument, we can conclude that the optimum value will be attained at the constraint boundaries.

Hence, we may conclude that the solutions of the online and the batch problems will be the same. ■