Online Context-Aware Streaming Data Release With Sequence Information Privacy

Bo Jiang, Ming Li, *Fellow, IEEE*, and Ravi Tandon, *Senior Member, IEEE*

***Abstract*— Publishing streaming data in a privacy-preserving manner has been a key research focus for many years. This issue presents considerable challenges, particularly due to the correlations prevalent within the data stream. Existing approaches either fall short in effectively leveraging these correlations, leading to a suboptimal utility-privacy tradeoff, or they involve complex mechanism designs that increase the computation complexity with respect to the sequence length. In this paper, we introduce Sequence Information Privacy (SIP), a new privacy notion designed to guarantee privacy for an entire data stream, taking into account the intrinsic data correlations. We show that SIP provides a similar level of privacy guarantee compared to local differential privacy (LDP), and it also enjoys a lightweight modular mechanism design. We further study two online data release models (instantaneous or batched) and propose corresponding privacy-preserving data perturbation mechanisms. We provide a numerical evaluation of how correlations influence noise addition in data streams. Lastly, we conduct experiments using real-world data to compare the utility-privacy tradeoff offered by our approaches with those from existing literature. The results reveal that our mechanisms achieve better utility-privacy tradeoff than the state-of-the-art LDP-based mechanisms. Notably, the improvements become more significant for small privacy budgets.**

***Index Terms*— Information privacy, time series data, continual release.**

I. Introduction

In THE era of big data, data sharing has become extensive and pervasive across various industries, transforming the way businesses and organizations operate. The data-sharing mechanisms play a critical role in enabling decision-making, analytics, and automation. The setting of these mechanisms can be broadly classified into two categories: offline and online. The offline setting considers static data/dataset, such as database queries, which involve accessing and utilizing stored data for various applications. The online setting, on the other hand, often involves real-time processing and dissemination of data generated by IoT devices or cloud-based systems [1]. These mechanisms encompass varied applications such as real-time heart rate monitoring via smartwatches [2], instant updates from cloud-based infrastructure, and smart grid management for efficient energy distribution, etc.

As the shared data may contain personally sensitive information, investigating data-sharing methods in a privacy-preserving manner becomes critical. Differential Privacy (DP) [3], [4], [5], which is *de facto* standard in the privacy research community, has achieved great success in the offline data-sharing setting and has led many real-world implementations, such as surveying demographics and commuting patterns [6], and the 2020 U.S. Census [7]. DP is well-suited for answering aggregated queries and requires a trusted server. Conversely, Local DP (LDP) mechanisms [5] allow for the publication of individual records without reliance on a trusted server. They can be used to answer both statistical and individual queries. LDP-based mechanisms have been successfully adopted by Google's RAPPOR [8] for collecting web browsing behavior, and Apple's MacOS to identify popular emojis and media preferences in Safari [9], [10]. However, previous research has indicated that when independent ϵ-LDP mechanisms are applied to correlated data, the actual leakage for each mechanism is significantly greater than ϵ for highly correlated data [11], [12], [13] (the leakage upper bound is $k\epsilon$ when releasing k consecutive correlated data points). A strict way to upper bound the privacy leakage is to properly allocate the global privacy budget to each LDP-based mechanism by sequential composition. However, the privacy budget allocated to each mechanism may be too small to ensure an ideal utility, because LDP provides strong (worst-case) privacy guarantees and fails to *leverage correlations* in their definitions. This oversight can potentially lead to less-than-optimal *utility-privacy tradeoffs*.

Context-aware privacy notions, which incorporate context information (typically the data distribution) in privacy definition, offer a more relaxed and adaptive way to measure privacy leakage [14]. Mutual Information Privacy (MIP), for instance, gauges the mutual information between the raw data and its release [15]. MIP evaluates the Kullback-Leibler (KL) divergence, a statistical measure that quantifies the expected distance between two distributions, thereby naturally incorporating the data's prior distribution and correlations. However, MIP provides an average-case privacy protection, which may not be sufficient in practice [16]. Moreover, MIP is not sequentially composable, making it unsuitable for the online setting. In such settings, the requirement for *sequential composability* means that the decomposed privacy guarantee of each time step must remain independent of subsequent

Manuscript received 19 July 2023; revised 12 December 2023 and 14 February 2024; accepted 7 March 2024. Date of publication 19 March 2024; date of current version 6 May 2024. The work of Ravi Tandon was supported by NSF under Grant CCF 2100013, Grant CNS 2209951, Grant CCF 1651492, Grant CNS 2317192, and Grant CNS 1822071. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Meng Li. *(Corresponding author: Bo Jiang.)*

The authors are with the Department of Electrical and Computer Engineering, The University of Arizona, Tucson, AZ 85721 USA (e-mail: bjiang@email.arizona.edu; lim@email.arizona.edu; tandonr@email.arizona.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TIFS.2024.3378008, provided by the authors.

Digital Object Identifier 10.1109/TIFS.2024.3378008

steps. Lastly, MIP-based mechanisms, which require averaging over the input/output's support, typically introduce substantial *computational complexity* due to the exponential growth of the support with the data sequence length. Notably, as online data release demands the timely sharing of data, the privacy protection mechanisms need to be lightweight and have low computation complexity to avoid causing any delays in the data release.

Another context-aware privacy notion, Local Information Privacy (LIP) [17], [18]. LIP bounds the privacy leakage via the ratio between the posterior and the prior data distribution, which provides a worst-case privacy guarantee similar to LDP, while the utility of LIP-based mechanisms can be significantly enhanced compared to LDP (we provide a further comparison in section VI). However, LIP and LIP-based privacy mechanisms were originally proposed/designed for offline/one-time data release. In this paper, we extend it to handle the time series data in the online setting and propose Sequential Information Privacy (SIP). We show that SIP has sequential composition properties similar to LDP (while the privacy is bounded by a factor of 2 from LDP, meaning it also achieves strong privacy). At the same time, SIP enjoys modular mechanism design with low complexity. We develop novel mechanisms for two data release settings: instantaneous release setting, where data is released as it is generated; and batched release setting, where data is accumulated and released periodically. Both approaches have their own merits and use cases. Our proposed mechanisms ensure real-time privacy preservation while maintaining data utility.

Our main contributions in this paper are three-fold:

- To quantify the privacy leakage in the data stream, we introduce a novel privacy notion, termed Sequence Information Privacy (SIP), which measures the overall privacy leakage in the data sequence. We consider two common real-world online data release settings: instantaneous data release and batched data release. Subsequently, we define two metrics, Instantaneous Information Leakage (IIL) and Batched Information Leakage (BIL), corresponding to the aforementioned settings. We show that SIP enjoys similar composition theorems as LDP, which is upper bounded by the sum of IIL and BIL in each time step, in accordance with the release setting in a linear or sub-linear fashion (advanced composition).
- We propose privacy protection mechanisms corresponding to each data release setting. For the instantaneous release setting, we derive the optimal mechanism and its parameters in closed form; furthermore, we demonstrate correlation-dependent noise through an example. For the batched release setting, we first show the problem can be degraded to a sub-optimal problem by simplifying the mechanism parameters. Then, we propose a data release algorithm with simplified parameters based on gradient descent. We study the influence of batch size on data utility and computational complexity.
- We provide extensive experimental results, utilizing two real datasets with different application types (and correspondingly, different utility measures). We evaluate the utility-privacy tradeoff provided by both mechanisms and compare these results with existing solutions. Our analysis shows that, while the privacy guarantee offered by ϵ-SIP is strictly stronger than 2ϵ-LDP, the utility provided by ϵ-SIP based is higher than the LDP-based mechanisms under 2ϵ-LDP, especially under small privacy budgets.

II. System and Threat Model

A. System Setup

Let us consider the scenario of releasing time-series data in an online manner. Denote the raw data at each time stamp k as X_k that takes value from a finite support $\mathcal{X}$. Denote the data stream up to time T as $\mathbf{X}_1^T = \{X_1, \ldots, X_T\}$, and $\mathbf{x}_1^T$ as a realization of $\mathbf{X}_1^T$. We use the bold symbol to denote a vector. In the context-aware setting, the data stream is considered as a correlated random vector with $\Pr(X_1 = x) = P_1(x)$, and $\Pr(X_{k+1} = \mathbf{v} | \mathbf{X}_1^k = \mathbf{u}) = C_{\mathbf{u}}^{\mathbf{v}}$, for all u, v sequence. Further, we consider two types of data release scenarios: 1) instantaneously release and 2) batched release. Instantaneous release means each of the data in the sequence is released instantaneously. One example of the instantaneous release is the navigation app on the smartphone, users are sending location data to the server and accessing location-based services on the fly. Another example is online games, users' operation data is collected by the server continuously (usually less than every 20 ms). On the other hand, we also consider data to be released in a batched manner, for the applications where moderate delay is allowed to minimize the communication cost. Backend software in smartphones or PCs implements batched release by periodically sending collected logs to the server. This allows for efficient data management and analysis. Traffic monitoring systems utilize batched-release to collect and upload aggregated traffic flow data at regular intervals. This approach ensures accurate monitoring of traffic density and enables effective analysis of traffic patterns. Models of these two release settings are depicted in Fig. 1.

1) Instantaneous Release Setting: In the instantaneous release setting, to protect data privacy, data at each time step (for example, time k) is perturbed to Y_k before being released to the public. Assume that Y_k takes a value of y from the same domain as X, and the perturbation is done by a randomized mechanism $\mathcal{M}_k^I$, where the superscript I in the notation denotes instantaneous release setting. The mechanism outputs Y_k by considering the whole data sequence till k as well as all previous outputs, i.e., $Y_k = \mathcal{M}_k^I(\mathbf{X}_1^k, \mathbf{Y}_1^{k-1})$. Also, it is natural to assume that $X_{k+t} \perp\!\!\!\perp Y_k | \{\mathbf{X}_1^k, \mathbf{Y}_1^{k-1}\}$, for all $t \geq 1$ (we use $\perp\!\!\!\perp$ to denote independent between random variables), as the current release should not depend on data at future time steps. More specifically, the mechanism $\mathcal{M}_k^I$ is defined as follows

$$a_k^I(y_k | x_k, \mathbf{x}_1^{k-1}, \mathbf{y}_1^{k-1}) = \Pr\left(Y_k = y_k \mid \mathbf{X}_1^k = \mathbf{x}_1^k, \mathbf{Y}_1^{k-1} = \mathbf{y}_1^{k-1}\right). \tag{1}$$

2) Batched Release Setting: The raw data sequence is partitioned into different batches. Denote $\mathbf{O}_l = \mathbf{X}_{(l-1)w+1}^{lw}$ as one batch generated from the raw sequence. w here denotes the length of the batch, and $l \in [1, \tau]$ represents the batch index. In this paper, we assume w of each batch to be the

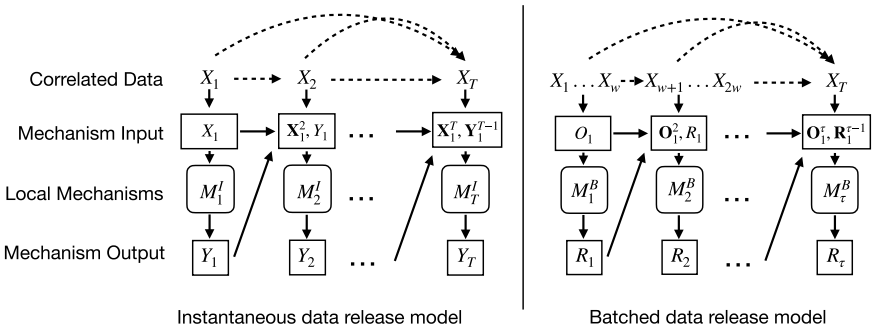


Fig. 1. Online privacy-preserving data release systems, a side-by-side comparison between the instantaneous data release model and the batched data release model.

same, however, it is straightforward to extend our analysis and results to the case where w for each batch is different from each other.

Similarly to the instantaneous setting, for data privacy consideration, at time l, a privacy protection mechanism considers all previous input/output and releases a perturbed version $\mathbf{R}_l$. Denote the mechanism for batched release as $\mathcal{M}_l^B(\mathbf{O}_1^l, \mathbf{R}_1^{l-1})$, then $\mathbf{R}_l = \mathcal{M}_l^B(\mathbf{O}_1^l, \mathbf{R}_1^{l-1})$, it is also natural to assume that $\mathbf{O}_{l+t} \perp\!\!\!\perp \mathbf{R}_l | \{\mathbf{O}_1^l, \mathbf{R}_1^{l-1}\}$, for all $t > 0$. As a result, $\mathcal{M}_l^B$ is defined as follows:

$$a_l^B(\mathbf{r}_l|\mathbf{o}_l, \mathbf{o}_1^{l-1}, \mathbf{r}_1^{l-1}) = \Pr\left(\mathbf{R}_l = \mathbf{r}_l \mid \mathbf{O}_1^l = \mathbf{o}_1^l, \mathbf{R}_1^{l-1} = \mathbf{r}_1^{l-1}\right). \tag{2}$$

B. Adversary Model

In this paper, the adversary can be anyone who has access to the released data. e.g. the server, or anyone in the public. We assume the adversary is honest but curious. He/She does not have access to the user's release system, and can only passively receive and observe the output sequence from the privacy-preserving mechanism. It is assumed that the adversary is interested in learning the raw data sequence. His/Her inference model stems from the Bayesian posterior probability distribution, given all historical observations. For the instantaneous release setting, for a set of observations $\mathbf{y}_1^{k-1}$, the adversary's belief of $\mathbf{X}_1^k$ is defined as:

$$\beta_k^I(\mathbf{x}_1^k|\mathbf{y}_1^{k-1}) = \Pr(\mathbf{X}_1^k = \mathbf{x}_1^k|\mathbf{Y}_1^{k-1} = \mathbf{y}_1^{k-1}). \tag{3}$$

We denote this posterior belief as the belief state of the adversary in the instantaneous release setting. For the batched release setting, similarly, the adversary's belief of $\mathbf{O}_1^l$ after receiving $\mathbf{r}_1^{l-1}$ is

$$\beta_l^B(\mathbf{o}_1^l|\mathbf{r}_1^{l-1}) = \Pr(\mathbf{O}_1^l = \mathbf{o}_1^l|\mathbf{R}_1^{l-1} = \mathbf{r}_1^{l-1}), \tag{4}$$

which is defined as the adversary's belief state in the batched release setting.

Besides, it is assumed that the adversary has the following abilities:

- The adversary knows the initial prior distribution of the first data $P_1(\cdot)$ and the data correlation in the data sequence. Hence, the adversary's belief state can be updated from time to time. This is a common assumption used in nearly all information-theoretic approaches. Note that this is a worst-case assumption. According to [18], if the adversary has a different knowledge from the true data prior distribution, the privacy leakage is decreased, and the reduced amount is proportional to the deviation from the true prior.
- The adversary knows the privacy-protection mechanism, including the release setting, current data index (time step), and the perturbation parameters.

III. Privacy Definition and Comparison With Existing Privacy Notions

A. Context-Aware Sequence Information Leakages

For data privacy, we start by defining the Sequence Information Leakage that occurs after a series of successive outputs. We then demonstrate how this leakage can be decomposed into each local leakage at various time steps.

Definition 1: The sequence information leakage (SIL) for releasing $\mathbf{Y}_1^T$ as privatized version of $\mathbf{X}_1^T$ is defined as:

$$\mathcal{L}(\mathbf{Y}_1^T \to \mathbf{X}_1^T) = \max_{\mathbf{x}_1^T, \mathbf{y}_1^T \in \mathcal{X}^T} \left| \log \frac{\Pr(\mathbf{X}_1^T = \mathbf{x}_1^T|\mathbf{Y}_1^T = \mathbf{y}_1^T)}{\Pr(\mathbf{X}_1^T = \mathbf{x}_1^T)} \right|. \tag{5}$$

This can be interpreted as the adversary's maximum information gain about $\mathbf{X}_1^T$ after observing the output sequence of $\mathbf{Y}_1^T$ compared to his prior knowledge. We say a privacy-preserving mechanism satisfies (ϵ, δ)-Sequence Information Privacy (SIP) if the following condition holds:

$$\Pr(\mathcal{L}(\mathbf{Y}_1^T \to \mathbf{X}_1^T) > \epsilon) < \delta. \tag{6}$$

We next define the instantaneous information leakage at each time step.

Definition 2: The instantaneous information leakage (IIL) at time k is defined as:

$$\mathcal{L}(Y_k \to \mathbf{X}_1^k) = \max_{\mathbf{x}_1^k \in \mathcal{X}^k, y_k \in \mathcal{X}} \left| \log \frac{\Pr(\mathbf{X}_1^k = \mathbf{x}_1^k|\mathbf{Y}_1^k = \mathbf{y}_1^k)}{\Pr(\mathbf{X}_1^k = \mathbf{x}_1^k|\mathbf{Y}_1^{k-1} = \mathbf{y}_1^{k-1})} \right|. \tag{7}$$

The operational meaning of the instantaneous information leakage can be interpreted as the adversary's additional belief on the data sequence $\mathbf{X}_1^k$ after observing Y_k at time k compared to the belief before taking this observation. Such a definition is presented based on an online data release manner, i.e., before observing Y_k, $k-1$ outputs have already been published.

Similarly, we define the information leakage in the setting of batched data release:

Definition 3: The batched information leakage (BIL) when releasing the l-th batched data sequence is defined as:

$$\mathcal{L}(\mathbf{R}_l \to \mathbf{O}_1^l) = \max_{\mathbf{o}_1^l \in \mathcal{X}^{lw}, \mathbf{r}_l \in \mathcal{X}^w} \left| \log \frac{\Pr(\mathbf{O}_1^l = \mathbf{o}_1^l | \mathbf{R}_1^l = \mathbf{r}_1^l)}{\Pr(\mathbf{O}_1^l = \mathbf{o}_1^l | \mathbf{R}_1^{l-1} = \mathbf{r}_1^{l-1})} \right|. \tag{8}$$

With similar operational meaning as the instantaneous information leakage: the adversary's additional knowledge on $\mathbf{O}_1^l$ after observing $\mathbf{R}_l$.

Regarding the relationship between the IIL/BIL and SIL, we have the following theorem:

Theorem 1: For a sequence of instantaneous-release privacy protection mechanisms $\mathcal{M}^I(1:T)$, such that $\mathcal{M}_k^I$ releases Y_k at time k, if the IIL $\mathcal{L}(Y_k \to \mathbf{X}_1^k) \leq \epsilon_k$, $\forall k \in 1, 2, .., T$, then $\mathcal{M}^I(1:T)$ satisfies $(\sum_{k=1}^T \epsilon_k, 0)$-SIP. Similarly, for a sequence of batched-release privacy protection mechanisms $\mathcal{M}^B(1:\tau)$, if each BIL satisfies $\mathcal{L}(\mathbf{R}_l \to \mathbf{O}_1^l) \leq \epsilon_l$, $\forall l \in 1, 2, .., \tau$, then $\mathcal{M}^B(1:\tau)$ satisfies $(\sum_{l=1}^\tau \epsilon_l, 0)$-SIP.

Theorem 1 posits that the privacy budget, as it pertains to a sequence of mechanisms, decomposes linearly in relation to the amount of data disclosed. Additionally, the aggregation of local leakages contributes to the global sequence leakage. Importantly, the introduction of a minor failure probability denoted as δ, allows for the achievement of sub-linear growth in the privacy budget. This property is summarized in the following Theorem.

Theorem 2: The sequence of mechanisms $\mathcal{M}^I(1:T)$ in Theorem 1 satisfies $(T\epsilon(e^\epsilon - 1) + \sqrt{T}\epsilon\sqrt{2\ln(1/\delta)}, \delta)$-SIP; similarly, the mechanisms $\mathcal{M}^B(1:\tau)$ satisfy $(\tau\epsilon(e^\epsilon - 1) + \tau\epsilon\sqrt{2\ln(1/\delta)}, \delta)$-SIP.

Remark: Theorem 2 is similar to the sequential composition of LDP, and the proof is done in a similar way. Theorem 1 and Theorem 2 effectively break down a global task into manageable local goals. Specifically, to design either the instantaneous $\mathcal{M}^I$ or batched $\mathcal{M}^B$ mechanism at each moment under a total SIP budget, it suffices to limit the IIL or BIL. Detailed proof of Theorem 1 and Theorem 2 are provided in the appendix.

B. Comparison With Existing Privacy Notions

1) Local Differential Privacy: The decentralized version of DP, Local Differential Privacy (LDP) [5], has gained much attention since its introduction. It adopts a similar structure as Differential Privacy but considers the input as each individual's data, so the privacy-utility tradeoff of each individual is customizable. The definition of LDP, when adapted to the sequential data release model, can be summarized as follows:

Definition 4: A privacy protection mechanism $\mathcal{M}$, is said to be ϵ-local differentially private for the sequence of $\mathbf{X}_1^T$, if for all $\mathbf{x}_1^T, \tilde{\mathbf{x}}_1^T \in \{\mathcal{X}\}_1^T$, and for all $\mathbf{y}_1^T \in Range(\mathcal{M}(\mathbf{X}_1^T))$,

$$\frac{\Pr(\mathbf{Y}_1^T = \mathbf{y}_1^T | \mathbf{X}_1^T = \mathbf{x}_1^T)}{\Pr(\mathbf{Y}_1^T = \mathbf{y}_1^T | \tilde{\mathbf{X}}_1^T = \tilde{\mathbf{x}}_1^T)} \in [e^{-\epsilon}, e^\epsilon]. \tag{9}$$

The relationship between ϵ-LDP and ϵ-SIP is shown in the following Theorem:

Theorem 3: If a privacy-preserving mechanism $\mathcal{M}$ satisfies ϵ-LDP, then it satisfies ϵ-SIP. Conversely, if $\mathcal{M}$ satisfies ϵ-SIP, then it satisfies 2ϵ-LDP.

In essence, ϵ-SIP offers a more relaxed privacy guarantee compared to ϵ-LDP. This is because, while LDP is designed to defend against adversaries possessing arbitrary prior knowledge, ϵ-SIP operates under the assumption that both prior knowledge and data correlations are known, fixed, or at least inferable from existing samples. In Theorem 3 we present a general conversion bound between SIP and LDP. Furthermore, a more tightened bound, dependent on data prior, can be formulated following a methodology similar to that in [18]. It is important to note, however, that this conversion of the privacy budget does not directly translate into a weakened privacy guarantee by the respective mechanisms. In Section VI, we demonstrate that SIP's relaxed definition and our mechanism do not compromise privacy under a uniform privacy measure (in our study, we use identification rate as the metric). Moreover, the 2ϵ bound associated with LDP is actually advantageous for the SIP framework, as it upper bounds the worst-case information leakage while taking the context into consideration, akin to the privacy guarantee of LDP. Nevertheless, due to the utilization of data prior/context, our mechanisms exhibit a more favorable utility-privacy tradeoff than that offered by LDP, even when we compare our mechanisms under the worst-case privacy leakage (with those satisfying 2ϵ-LDP).

LDP's worst-case privacy protection means the condition in (9) must hold for every possible $\mathbf{x}_1^T, \tilde{\mathbf{x}}_1^T$, and $\mathbf{y}_1^T$. LDP is also sequentially decomposable by applying independent mechanisms [22], and can be adapted in the online release setting. Formally, for a sequence of privacy-preserving mechanisms $\mathcal{M}(1:T)$, if each mechanism $\mathcal{M}$ satisfies ϵ_k-LDP for X_k. Then $\mathcal{M}(1:T)$ satisfies $\sum_{k=1}^T \epsilon_k$-LDP for $\mathbf{X}_1^T$ [23]. When implementing LDP with sequential composition, the total privacy budget could drastically inflate if each local mechanism is using a budget that is reasonable for ensuring utility. Conversely, the utility of each local data release could significantly diminish if a more restrained budget is applied to maintain robust privacy protection.

The field of sequential data release, employing LDP or its relaxed variants, has rapidly gained attention. For instance, the concept of w-event-level LDP was introduced in a study by [24]. This approach ensures that each segment of a data sequence within a window of w maintains an ϵ-LDP guarantee. The overall privacy leakage is composed of a cumulative function of leakages across different windows. This concept builds upon the central DP version initially proposed in [25]. However, event-level protection will degrade the privacy guarantee. In another study by Xue et al. [26], a Dynamic Difference Report Mechanism (DDRM) was proposed. DDRM is designed to safeguard changes in data during continuous release, with LDP ensuring the privacy of these change points. However, DDRM's applicability is limited in scenarios where time-series data exhibit significant fluctuations over time. In [27], another LDP-based mechanism is proposed for the real-time estimation of item counts in streaming data. This approach, combining Cuckoo and Bloom filters, enhances

utility (accuracy of count estimates) and querying efficiency, particularly for fuzzy counting scenarios. Furthermore, the work by Jain et al. [28] analyzes the data utility of continual release mechanisms. They present lower bounds on errors for DP-based mechanisms, particularly when dealing with "Max-Sum" and "Sumselect" queries. Despite their significance, adapting these results to local settings remains a challenging task, underscoring the need for further research in this area.

Recent studies have begun to incorporate data correlation into their model assumptions to enhance the utility-privacy tradeoff provided by the LDP mechanism. For example, Cao et al. proposed a method to quantify DP leakage for data with temporal correlations [29]. They developed DP mechanisms for location aggregations under temporal correlation [30], under the assumption that data correlation can be discerned by an adversary. Subsequently, they define a subset of the output's support to decrease the DP sensitivity. However, such an approach may result in an increased failure probability of DP, also the correlation itself is still not leveraged in the mechanism design. Further studies [31], [32], [33], [34], [35] also explored scenarios where adversaries possess knowledge of data correlations. These works primarily concentrate on creating novel mechanisms to protect a single user's privacy by extending DP. Nevertheless, unlike the setting of this paper, these studies consider sequential data release in an offline setting. In contrast, Wang et al. put forward an online data release mechanism in [19], satisfying either DP or LDP, depending on the specific circumstances. However, still, their proposed noise-adding mechanisms for each instantaneous data point operate independently. In [36], Zhang et al. propose to learn the correlation as the sequence is generated, using it to estimate future data. This guides the generation of noisy released data, allowing for real-time queries with higher utility. However, their work assumes a central setting, which is different from the setting considered in this paper. Also, [2] adopts DP as the privacy notion, which is not context-aware.

2) Pufferfish Privacy: Another privacy notion that provides privacy protection over a set of self-defined secrets is Pufferfish privacy. When adapted into the data release model, Pufferfish privacy is defined as:

Definition 5 (ϵ-Local Pufferfish Privacy [12]): Given set of potential secrets $\mathbb{S}$, a set of discriminative pairs $\mathbb{S}_{pairs}$, a set of data evolution scenarios $\mathscr{P}_{\{\mathcal{X}\}_1^T,\mathcal{S}}$, and a privacy parameter $\epsilon \in R^+$, an (potentially randomized) algorithm $\mathcal{M}$ satisfies ϵ-PufferFish ($\mathbb{S}$, $\mathbb{S}_{pairs}$, $\mathscr{P}_{\{\mathcal{X}\}_1^T,\mathcal{S}}$) privacy if

- *for all possible outputs $\mathbf{y}_1^T \in range(\mathcal{M})$,*
- *for all pairs $(s_i, s_j) \in \mathbb{S}_{pairs}$ of potential secrets,*
- *for all distributions $\boldsymbol{P}_{\{\mathcal{X}\}_1^T,\mathcal{S}} \in \mathscr{P}_{\{\mathcal{X}\}_1^T,\mathcal{S}}$ for which $\Pr(s_i|\boldsymbol{P}_{\{\mathcal{X}\}_1^T,\mathcal{S}}) \neq 0$ and $\Pr(s_j|\boldsymbol{P}_{\{\mathcal{X}\}_1^T,\mathcal{S}}) \neq 0$*

the following holds:

$$e^{-\epsilon} \leq \frac{\Pr(\mathcal{M}(\mathbf{X}_1^T) = \mathbf{y}_1^T|\boldsymbol{P}_{\{\mathcal{X}\}_1^T,\mathcal{S}}, s_i)}{\Pr(\mathcal{M}(\mathbf{X}_1^T) = \mathbf{y}_1^T|\boldsymbol{P}_{\{\mathcal{X}\}_1^T,\mathcal{S}}, s_j)} \leq e^{\epsilon}.$$

The relationship between Pufferfish Privacy and SIP isn't directly deducible as they operate under different assumptions. Pufferfish assumes the possibility of multiple data generation scenarios, captured by $\mathbf{P}_{\mathcal{X}_1^T,\mathcal{S}}$. Conversely, SIP presumes that the correlation among data is given. In the context of streaming data release, these correlations can naturally be learned from previous releases, thus SIP is more suitable for the online release setting.

Pufferfish privacy further distinguishes itself by protecting a latent variable S that correlates with the data stream, as opposed to the data itself. This model has been the subject of other studies, primarily from an information-theoretic perspective, such as those in [37], [38], and [39]. Contrarily, SIP postulates that each individual data value is privately sensitive. As a result of this assumption, Pufferfish isn't as intuitively decomposable like LDP and SIP, without adopting additional assumptions of Markovity in the data sequence.

Among the existing literature on Pufferfish Privacy, [20] proposed a privacy protection mechanism that also considers data correlation. The proposed mechanism operates under the assumption that the data distribution adheres to the Markov Quilt properties. This premise simplifies data correlation and reduces computational complexity, while also allowing the mechanism to be sequentially composable. However, the data release mechanism only functions when the secrets are sequentially obtained, whereas the data sequence is predetermined. This assumption renders the mechanism unsuitable for an online setting. Furthermore, the algorithm necessitates an exhaustive search of all possible combinations of proximate and distant nodes in the Markov Quilt, resulting in only a modest reduction in complexity.

3) Information Theoretical Approaches: We next compare SIP with privacy notions borrowed from information theory. The first definition to compare with is Mutual Information Privacy (MIP), which is measured by the mutual information between the input and the output of the privacy-protection mechanism:

Definition 6: [15] A mechanism $\mathcal{M}$ satisfies ϵ-MIP for some $\epsilon \in \mathbf{R}^+$, if the mutual information between $\mathbf{X}_1^T$ and $\mathbf{Y}_1^T$ satisfies $I(\mathbf{X}_1^T; \mathbf{Y}_1^T) \leq \epsilon$.

It is evident that a privacy-preserving mechanism $\mathcal{M}$ based on SIP, guaranteeing $\mathcal{L}(\mathbf{Y}_1^T \to \mathbf{X}_1^T) \leq \epsilon$, would also ensure that the mutual information $I(\mathbf{X}_1^T; \mathbf{Y}_1^T) \leq \epsilon$. This is due to the mutual information being a statistical average of the SIP. MIP provides a relatively weak average-case privacy guarantee. Also, MIP is not sequentially composable: after applying the chain rule, $I(\mathbf{X}_1^T; \mathbf{Y}_1^T) \leq \epsilon$ can only be decomposed into bounding each $I(\mathbf{X}_1^T; Y_t|\mathbf{Y}_1^{t-1})$, which depends on the whole input sequence. Nevertheless, as mutual information or conditional mutual information intuitively captures data correlation and can be easily decomposed using the chain rule, MIP has been extensively explored as a privacy metric for time-series data, as demonstrated in [40].

Several studies on privacy-preserving online data release, such as [21] and [41], have also capitalized on data correlation, releasing aggregated location data online while ensuring individual or group privacy measured by MIP is constrained. As mutual information functions are convex, they can conveniently be integrated into optimization problems. The obfuscation mechanism at different time stamps is selected by a reinforcement learning algorithm. However, the complexity is still high depending on the length of the sequence.

TABLE I
COMPARISON OF DIFFERENT PRIVACY NOTIONS IN THE ONLINE SETTING

	Definition			Mechanism	
Privacy Notion	**Sequential Composability**	**Privacy Guarantee**	**Leveraging Correlation**	**Computation Complexity**	**Online Release**
LDP	Yes	Worst-case	No	$\mathcal{O}(\lvert\mathcal{X}\rvert)$ [19]	Yes
Pufferfish	Yes, with Markovity	Worst-case	Yes	$\mathcal{O}(T^3\lvert\mathcal{X}\rvert^3)$ [20]	No
MIP	No	Average-case	Yes	$\mathcal{O}(\lvert\mathcal{X}^{2T}\rvert)$ [21]	Yes
SIP (this paper)	Yes	Worst-case	Yes	$\mathcal{O}(\lvert\mathcal{X}\rvert)$	Yes

SIP can be perceived as an online sequential version of local Information Privacy (LIP) [17], [18], [42], defined as the ratio between the prior and posterior probabilities following observation of the output from the mechanism. In [17], comprehensive results on mechanism design based on LIP are offered, showing superior utility-privacy trade-offs compared to LDP, MIP, and Pufferfish. However, none of these mechanisms take into account the context of online sequential data release.

Considering the four typical challenges for sequential data release in the online setting: utility-privacy tradeoff, sequential composability, leveraging correlation, and low computation complexity, we summarize different privacy notions described in this section and how these notions address the four challenges in Table I.

IV. UTILITY PRIVACY TRADEOFF FOR INSTANTANEOUS RELEASE

A. Problem Formulation

In this paper, we define data utility as the expected distance between the arbitrary Quality of Service (QoS) function of the input ($Q(X)$) and output ($Q(Y)$). In the instantaneous release model, the utility is gauged by the expected distance between $Q(X_k)$ and $Q(Y_k)$ at each time stamp k. This measurement is known as the Instantaneous Expected Distance (IED):

$$E\left[\mathsf{D}(Q(X_k), Q(Y_k))\middle|\mathbf{Y}_1^{k-1}=\mathbf{y}_1^{k-1}\right]. \tag{10}$$

In (10), Q signifies the query function of X and Y that depends on the specific application, while $\mathsf{D}(a,b) : (R,R) \to R^+$ denotes a distortion or distance measure between a, b. The expectation $E[\cdot]$ is taken over both the underlying distribution of the data $P_X(x)$, as well as over the randomness of the mechanism. The expected distance between $Q(X)$ and $Q(Y)$ represents a general type of utility measurement. For instance, in a Location-Based Service (LBS), $Q(X) = X$, and the Euclidean distance between X and Y is typically used to gauge performance: $\mathsf{Utility} = -E[(X-Y)^2]$. Another example is histogram estimation, where the aim is to ascertain how many people belong to each data category or classification according to users' data value. In this case, $Q(X)$ is an indicator function, and the absolute distance utility function can be written as $\mathsf{Utility} = -\sum_{i=1}^{K} E[|\mathbb{1}_{X\in S_i} - \mathbb{1}_{Y\in S_i}|]$. These examples demonstrate that for different applications, the utility function can be adapted by modifying the Q function and the distortion function $\mathsf{D}(\cdot,\cdot)$.

The online mechanism we explore in this paper concentrates on minimizing the IED defined in Eq. (10), subject to IIL constraints i.e., the problem is defined as:

$$\begin{aligned}&\min E\left[\mathsf{D}(Q(X_k), Q(Y_k))\middle|\mathbf{Y}_1^{k-1}=\mathbf{y}_1^{k-1}\right],\\ &\text{Such that } \mathcal{L}(Y_k \to \mathbf{X}_1^k) \le \epsilon_l,\ \forall k \in [1,T].\end{aligned} \tag{11}$$

1) Privacy Metric: By Bayes rule, the privacy metric in the IIL can be expressed as:

$$\begin{aligned}&\frac{\Pr(\mathbf{X}_1^k=\mathbf{x}_1^k|\mathbf{Y}_1^k=\mathbf{y}_1^k)}{\Pr(\mathbf{X}_1^k=\mathbf{x}_1^k|\mathbf{Y}_1^{k-1}=\mathbf{y}_1^{k-1})}\\ &\quad=\frac{\Pr(Y_k=y_k|\mathbf{X}_1^k=\mathbf{x}_1^k,\mathbf{Y}_1^{k-1}=\mathbf{y}_1^{k-1})}{\Pr(Y_k=y_k|\mathbf{Y}_1^{k-1}=\mathbf{y}_1^{k-1})}\\ &\quad=\frac{a_k^I(y_k|\mathbf{x}_1^k,\mathbf{y}_1^{k-1})}{\sum_{\tilde{\mathbf{x}}_1^k\in\mathcal{X}^k} a_k^I(y_k|\tilde{\mathbf{x}}_1^k,\mathbf{y}_1^{k-1})\beta_k^I(\tilde{\mathbf{x}}_1^k|\mathbf{y}_1^{k-1})}.\end{aligned} \tag{12}$$

Note that, the term $\beta_k(\tilde{\mathbf{x}}_1^k|\mathbf{y}_1^{k-1}) = \Pr(\mathbf{X}_1^k=\tilde{\mathbf{x}}_1^k|\mathbf{Y}_1^{k-1}=\mathbf{y}_1^{k-1})$ can be further expressed as:

$$\begin{aligned}&\Pr(\mathbf{X}_1^k=\tilde{\mathbf{x}}_1^k|\mathbf{Y}_1^{k-1}=\mathbf{y}_1^{k-1})\\ &=\Pr(X_k=\tilde{x}_k|\mathbf{X}_1^{k-1}=\tilde{\mathbf{x}}_1^{k-1})\Pr(\mathbf{X}_1^{k-1}=\tilde{\mathbf{x}}_1^{k-1}|\mathbf{Y}_1^{k-1}=\mathbf{y}_1^{k-1})\\ &=C^{\tilde{x}_k}_{\mathbf{x}_1^{k-1}}\frac{a_{k-1}^I(y_{k-1}|\tilde{\mathbf{x}}_1^{k-1},y_1^{k-1})\beta_{k-1}^I(\tilde{\mathbf{x}}_1^{k-1}|\mathbf{y}_1^{k-2})}{\sum_{\bar{\mathbf{x}}_1^{k-1}} a_{k-1}^I(y_{k-1}|\bar{\mathbf{x}}_1^{k-1},y_1^{k-2})\beta_{k-1}^I(\bar{\mathbf{x}}_1^{k-1}|\mathbf{y}_1^{k-2})},\end{aligned} \tag{13}$$

where the nominator is the perturbation probability, and the denominator is a linear combination of the perturbation parameters with coefficients of some calculable posteriors.

2) Utility Function: The utility function can be further expressed as:

$$\begin{aligned}&E\left[D(Q(X_k)-Q(Y_k))|\middle|\mathbf{Y}_1^{k-1}=\mathbf{y}_1^{k-1}\right]\\ &\quad=\sum_{x,y} D(Q(x)-Q(y))\\ &\qquad\cdot\sum_{\mathbf{x}_1^{k-1}} a_k^I(y_k|\mathbf{x}_1^{k-1},x,\mathbf{y}_1^{k-1})\beta_k^I(\mathbf{x}_1^{k-1},x|\mathbf{y}_1^{k-1}).\end{aligned} \tag{14}$$

As a result, the utility-privacy tradeoff can be expressed as:

$$\min\ (14),\ \text{Such that } (12) \in [e^{-\epsilon_k}, e^{\epsilon_k}]. \tag{15}$$

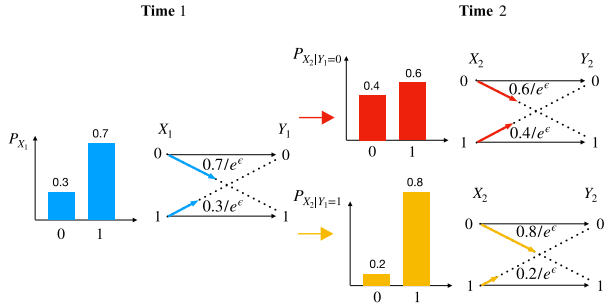


Fig. 2. Illustration of Conditional Randomize Response (CRR) mechanism, each mechanism parameters depend on data prior as well as previous release.

Hence, the local optimization problem can be represented as a function of the data correlation, previous data release policy, and belief state.

B. Conditional Randomize Response Mechanism

Next, we introduce the Conditional Randomized Response (CRR) perturbation mechanism, followed by our optimal mechanism based on CRR.

In a context-aware setting, the input to a privacy-preserving mechanism consists of the data values and their probabilistic distribution. For the online release setting, this distribution isn't fixed but instead depends on all previously observable releases. Therefore, the data distribution can be symbolized as the belief state defined earlier. This observation leads us to the CRR mechanism, which takes as input the current data value, as well as the belief state.

An illustrative example of the CRR mechanism is shown in Fig. 2. This mechanism perturbs X_1 according to probabilities designed based on the adversary's prior knowledge. After Y_1 is output, the adversary's prior knowledge about X_2 shifts to $\Pr(X_2|Y_1)$. Hence, the mechanism designs the perturbation parameters at time 2 in accordance with the distribution of $\Pr(X_2|Y_1)$.

We next derive the closed-form optimal solutions that minimize IED at each time step, which follows the next theorem.

Theorem 4: For the class of utility function with distance D satisfying the following properties: 1. non-negativity: $D(X,Y) \geq 0$; 2. Identity of Indiscernibles: $D(X,X) = 0$, 3. Symmetry: $D(X,Y) = D(Y,X)$, and 4. Triangle Inequality: $D(X,Y) \leq D(X,Z) + D(Z,Y)$, the following perturbation parameters at time k is the optimal solution of the problem defined in (15),

$$a_k^I(y_k|\mathbf{x}_1^k, \mathbf{y}_1^{k-1}) = \begin{cases} \dfrac{\beta^I(x_k|\mathbf{y}_1^{k-1})}{e^{\epsilon_k}}, & \text{if } y_k \neq x_k; \\ 1 - \dfrac{1-\beta^I(x_k|\mathbf{y}_1^{k-1})}{e^{\epsilon_k}}, & \text{if } y_k = x_k. \end{cases}$$

Detailed proof is provided in the appendix in the supplementary document. Key insights of the optimal solutions:

- The optimal perturbation parameters are found at the boundaries of the privacy constraints so that the least amount of noise is added while privacy can be protected.
- The conditional probability of Y_k is identical to that of X_k given $\mathbf{Y}_1^{k-1}$, i.e., $\Pr(X_k = x_k|\mathbf{Y}_1^{k-1} = \mathbf{y}_1^{k-1}) = \Pr(Y_k = x_k|\mathbf{Y}_1^{k-1} = \mathbf{y}_1^{k-1})$, which means the output at each time step always has the same marginal distribution of the input data.
- The mechanism does not depend on the historical raw data sequence. i.e. $\mathbf{x}_1^{T-1}$. This is due to the following two aspects: 1. the data utility at time k only depends on X_k. 2. A simplification from $\mathbf{x}_1^{T-1}$ won't violate the privacy constraints.

Algorithm 1 SIP Mechanism for Instantaneous Release

Input: current time k, initial prior $P_1(x)$, data correlation $C_{\mathbf{v}}^{\mathbf{u}}$, historical release sequence $\mathbf{Y}_1^{k-1}$, current X_k, previous $\beta^I(x_{k-1}|\mathbf{y}_1^{k-2})$.
Output: Instantaneous release Y_k
if $k = 1$:
    Release $Y_1 = y_1$ according to the following rule:
    $y_1 = x_1$ w.p. $1 - (1 - P_1(y_1))/e^\epsilon$;
    other y_1 w.p. $P_1(y_1)/e^\epsilon$.
else:
    for all $x_k \in \mathcal{X}$:
        update $\beta^I(x_k|\mathbf{y}_1^{k-1})$ according to (13):
    Release Y_k according to the following rule:
    $y_k = x_k$ w.p. $1 - (1 - \beta^I(y_k|\mathbf{y}_1^{k-1}))/e^\epsilon$,
    other y_k w.p. $\beta^I(y_k|\mathbf{y}_1^{k-1})/e^\epsilon$.

Given the optimal mechanism parameters, the algorithm for instantaneous release is shown in Alg. 1. As the calculating of the belief state takes constant time complexity, the computation complexity of Alg. 1 is $\mathcal{O}(|\mathcal{X}|)$.

1) Expected Error Upperbound: We proceed to assess the worst-case utility that can be attained by the SIP mechanism, as delineated in Alg. 1. In this analysis, the absolute error is utilized as a representative instance for the utility distance metric $D(\cdot,\cdot)$, though it is worth noting that the methodology is applicable to other forms of utility distance measures in a similar manner. Under the mechanism with parameters defined in Theorem 4, the utility function defined in (10) can be further expressed as follows:

$$\sum_{x \neq y} |x - y| \beta(x|\mathbf{y}_1^{k-1}) \beta(y|\mathbf{y}_1^{k-1}) / e^\epsilon. \tag{16}$$

The next proposition states the utility guarantee provided by the instantaneous SIP release mechanism.

Proposition 1: The error expression for SIP instantaneous release, as derived in (16), is upper bounded by

$$(x_{\max} - x_{\min})/2e^\epsilon,$$

where $x_{\max} \triangleq \max \mathcal{X}$ and $x_{\min} \triangleq \min \mathcal{X}$.

Proof: As $\sum_{x \in \mathcal{X}} \beta(y|\mathbf{y}_1^{k-1}) = 1$, the error upper bound is achieved when $\beta(x_{\max}|\mathbf{y}_1^{k-1}) = \beta(x_{\min}|\mathbf{y}_1^{k-1}) = 0.5$. □

We next theoretically compare the utility provided by instantaneous SIP and RR-LDP. Since the error upper bounds may not be exact or tight, directly comparing the upper bounds may not accurately reveal the superiority of one mechanism over another. In the following, we consider a binary example, with $\mathcal{X} = \{0, 1\}$, and the comparison between SIP and the exact expected error of the RR-LDP mechanism is provided in the following proposition.

Proposition 2: The error upper bound for the instantaneous release SIP mechanism, operating under a binary model, is $1/2e^{\epsilon}$. *In contrast, the* ϵ*-RR-LDP mechanism constantly incurs an expected error of* $1/(e^{\epsilon}+1)$. [1]

The expected error of RR-LDP is strictly larger than the upper bound of the instantaneous release SIP mechanism, which implies SIP constantly outperforms the RR-LDP mechanism.

C. Data Correlation Dependent Noise

We present an illustrative example next to elucidate how data correlation influences the amount of noise added at different time steps. When data is correlated, the necessary noise at each timestamp also relies on one another.

Example 1: Imagine releasing two correlated data points X_1 *and* X_2, *and* $X_1, X_2 \in \{0, 1\}$. *Denote the prior distribution of* X_1 *as* $P_1 = Pr(X_1 = 1)$ *and* $1 - P_1 = Pr(X_1 = 0)$. *We assume that the relationship between* X_1 *and* X_2 *is represented by a symmetric correlation channel, expressed as* $Pr(X_2 = 1|X_1 = 0) = Pr(X_2 = 0|X_1 = 1) = \phi$. *Prior to being published,* X_1 *and* X_2 *are perturbed to* Y_1 *and* Y_2 *respectively by the mechanisms* $\mathcal{M}_1$ *and* $\mathcal{M}_2$. *To ensure privacy, the local leakage of* $\mathcal{L}_1(Y_1 \to X_1) \le \epsilon_1$ *and* $\mathcal{L}_2(Y_2 \to X_2, X_1) \le \epsilon_2$ *must hold.*

1) Data Correlation and Noise Correlation: The correlation coefficient of X_1 and X_2 can be expressed as:

$$\rho_{X_1,X_2} = \frac{\sigma_{X_1X_2}}{\sigma_{X_1}\sigma_{X_2}} = \frac{E[X_1X_2] - E[X_1]E[X_2]}{\sqrt{\{E[X_1^2] - E^2[X_1]\}\{E[X_2^2] - E^2[X_2]\}}}.$$

Taking values in, we have:

$$\rho_{X_1,X_2} = \frac{(1-2\phi)\sqrt{P_1(1-P_1)}}{\sqrt{[P_1\phi + (1-P_1)(1-\phi)][(1-P_1)\phi + P_1(1-\phi)]}} \tag{17}$$

Observe that when $\phi = 0$, $\rho_{X_1,X_2} = 1$, when $\phi = 1/2$, $\rho_{X_1,X_2} = 0$; when $\phi = 1$, $\rho_{X_1,X_2} = -1$. Thus by varying the value of ϕ from 0 to 1, the correlation coefficient also changes from positive to negative.

The noise N at time 1 / time 2 in this example is defined as a binary random variable, such that $N = 1$ means the data value is flipped before release; $N = 0$ means the data is directly released. We refer to $\Pr(N = 1)$ as the "amount of noise". Our subsequent objective is to determine the correlation coefficient of ρ_{N_1,N_2} (detailed derivation of ρ_{N_1,N_2} as well as other parameters are shown in the Appendix).

2) Relationship Between ρ_{X_1,X_2} *and* ρ_{N_1,N_2}*:* We initially fix the value of P_1, then vary ϕ from 0 to 1 and observe changes in ρ_{N_1,N_2}. We know that ρ_{x_1,x_2} also ranges from -1 to 1. Besides charting how noise correlation varies, we are also interested in identifying some specific cases. For instance, when $\rho_{X_1,X_2} = 1$ or $\rho_{X_1,X_2} = -1$, we want to ascertain the value of ρ_{N_1,N_2} and how it is influenced by the prior of X_1. Nevertheless, even if $\rho_{N_1,N_2} = 0$, it doesn't conclusively prove independence. Thus, we consider calculating the mutual information between N_1 and N_2. If $I(N_1; N_2) = 0$, we can conclude that N_1 and N_2 are independent.

Furthermore, we aim to demonstrate how the prior of X_1 and the correlation between X_1 and X_2 affect the amount of noise required at the second timestamp. The hope is that under the correlated release mechanism when data are more strongly correlated, the required amount of noise at the subsequent timestamp decreases.

Finally, we set an upper bound for the noise at time 2, and identify the optimal parameters that minimize the total privacy leakage of X_1 and X_2. For comparison, we also derive the parameters that minimize the leakage of X_2, which is equivalent to treating X_1 and X_2 as independent (with the prior of X_2 changing according to the correlations). The goal of this comparison is to illustrate that by considering data correlation, under a fixed amount of noise, we can further minimize the total leakage of the data stream. In other words, we are adding noise where it matters the most.

The results of our analysis are depicted in Fig. 3 and 4, where Fig. 3 shows the outcomes when $P_1 = 0.5$ and Fig. 4 presents the results when $P_1 = 0.95$.

From both Fig. 3 and Fig. 4, we observe a decrease in ρ_{N_1,N_2} as ρ_{X_1,X_2} approaches 0. This suggests that when the data are more correlated, the noise added is more dependent on the previous data release mechanisms. Conversely, when $\rho_{X_1X_2} = 0$, $I(N_1; N_2) = 0$, which implies that when data X_1 and X_2 are independent, the noise added at two different timestamps is also independent. Another observation is that as X_1 and X_2 become more correlated, the required amount of noise at time 2 decreases correspondingly. This is because Y_1 already conveys a substantial amount of information about X_2, and the prior of X_2 is relatively more skewed than in scenarios where X_1 and X_2 are less correlated. When the amount of noise at time 2 is bounded, introducing correlated noise can further minimize the total privacy leakage. When $P_1 = 0.5$, the independent noise remains unchanged because the prior of X_2 does not fluctuate with ϕ. Upon comparing Fig. 3 and 4, we find that the intercept of Fig. 4 is greater than that of Fig. 3. This implies that when the prior is more uniformly distributed, the correlation of the noise decreases, while when the prior distribution of P_1 is more skewed, the correlation of the noise increases.

V. Utility-Privacy Tradeoff for Batch Setting

Note that we cannot directly extend the optimal solution in Theorem 4 to the release vector in the batched release setting, as the optimal parameters only maximize the probability of releasing one output that is identical to the input, which is not optimal for releasing vectors containing multiple data. Because they fail to enlarge the probabilities to release the sequence where most data are identical to the input but only a few are different. In this section, we first investigate the utility and privacy tradeoff in the batched release setting, followed by model simplification. Finally, we present our algorithm in the batched setting.

[1]The optimal RR-LDP mechanism under binary model is $p = e^{\epsilon}/(e^{\epsilon}+1)$, and $q = 1/(e^{\epsilon}+1)$, where p denotes the probability to direct release X, and q denotes the probability to release $1 - X$.

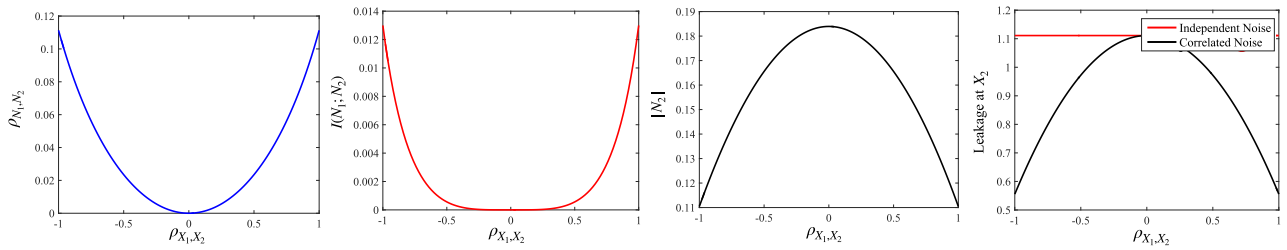


Fig. 3. Noise correlation, Mutual Information, Noise at time 2 and minimized leakage as a function of the correlation coefficient (when $P_1 = 0.5$).

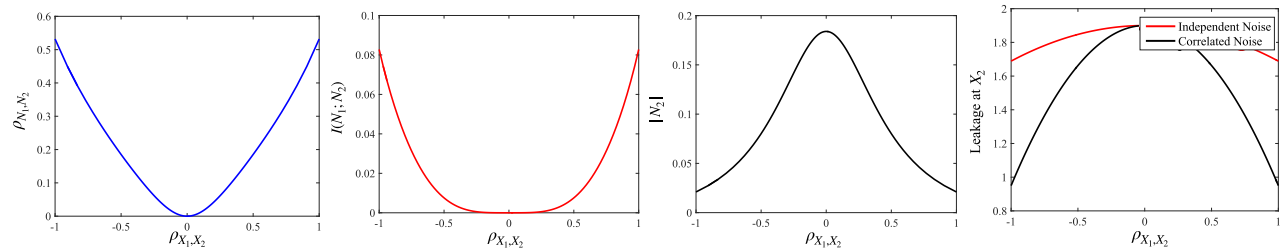


Fig. 4. Noise correlation, Mutual Information, Noise at time 2 and minimized leakage as a function of the correlation coefficient (when $P_1 = 0.95$).

A. Utility in the Batched Release Setting

In the batched release setting, the utility measurement is termed the Batched Expected Distance (BED):

$$E\left[\mathsf{D}(Q(\mathbf{O}_l), Q(\mathbf{R}_l))\middle|\mathbf{R}_1^{l-1} = \mathbf{r}_1^{l-1}\right]. \tag{18}$$

Similar to the instantaneous release setting, for the batched release setting, the optimization problem that yields the utility-privacy tradeoff is defined as:

$$\begin{aligned}&\min E\left[\mathsf{D}(Q(\mathbf{O}_l), Q(\mathbf{R}_l))\middle|\mathbf{R}_1^{l-1} = \mathbf{r}_1^{l-1}\right],\\ &\text{Such that } \mathcal{L}(\mathbf{R}_l \to \mathbf{O}_1^l) \le \epsilon_l,\ \ \forall l \in [1,\tau].\end{aligned} \tag{19}$$

The privacy constraint of the batch setting can be expressed as:

$$\frac{a_l^B(r_l|\mathbf{o}_1^l, \mathbf{r}_1^{l-1})}{\sum_{\tilde{\mathbf{o}}_1^l} a_l^B(r_l|\tilde{\mathbf{o}}_1^l, \mathbf{r}_1^{l-1})\beta_l^B(\tilde{\mathbf{o}}_1^l|\mathbf{r}_1^{l-1})}. \tag{20}$$

The belief state $\beta_l^B(\mathbf{o}_1^l|\mathbf{r}_1^{l-1})$ can be expressed as:

$$C_{\tilde{\mathbf{o}}_1^{l-1}}^{\tilde{o}_l} \frac{a_{l-1}^B(r_{l-1}|\tilde{\mathbf{o}}_1^{l-1}, \mathbf{r}_1^{l-1})\beta_{l-1}^B(\tilde{\mathbf{o}}_1^{l-1}|\mathbf{r}_1^{l-2})}{\sum_{\tilde{\mathbf{o}}_1^{l-1}} a_{l-1}^B(r_{l-1}|\tilde{\mathbf{o}}_1^{l-1}, \mathbf{r}_1^{l-2})\beta_{l-1}^B(\tilde{\mathbf{o}}_1^{l-1}|\mathbf{r}_1^{l-2})}, \tag{21}$$

where the correlation term can be derived as:

$$\begin{aligned}C_{\mathbf{o}_1^{l-1}}^{\mathbf{o}_l} &= \Pr(\mathbf{o}_l|\mathbf{o}_1^{l-1})\\ &= \prod_{i=(l-1)w+1}^{lw} \Pr\left(X_i = x_i\middle|\mathbf{X}_1^{i-1} = \mathbf{x}_1^{i-1}\right).\end{aligned} \tag{22}$$

The utility function of BED can be further expressed as:

$$\begin{aligned}&E\left[D(Q(\mathbf{O}_l) - Q(\mathbf{R}_l))\middle|\mathbf{R}_1^{l-1} = \mathbf{r}_1^{l-1}\right]\\ &= \sum_{\mathbf{o},\mathbf{r}} D(Q(\mathbf{o}) - Q(\mathbf{r}))\\ &\quad\times \sum_{\mathbf{o}_1^{l-1}} a_l^B(r_l|\mathbf{o}_1^{l-1}, \mathbf{o}, \mathbf{r}_1^{l-1})\beta_l^B(\mathbf{o}_1^{l-1}, \mathbf{o}|\mathbf{r}_1^{l-1}).\end{aligned} \tag{23}$$

Mechanism simplification Note that the mechanism parameter a_l^B contains the whole raw data sequence, which makes the computational cost grow exponentially. Next, we introduce a subset of policies that requires a memory of length L:

$$\begin{aligned}&a_l^s(\mathbf{r}_l|\mathbf{o}_{l-L+1}^l, \mathbf{r}_1^{l-1})\\ &= \Pr\left(\mathbf{R}_l = \mathbf{r}_l \mid \mathbf{O}_{l-L+1}^l = \mathbf{o}_{l-L+1}^l, \mathbf{R}_1^{l-1} = r_1^{l-1}\right).\end{aligned} \tag{24}$$

The next Theorem states that considering a subset of the policies (24) will not violate the privacy constraints

Theorem 5: For a batched release mechanism $\mathcal{M}_l^s$ that is parameterized by a_l^s, and satisfies the condition in (25),

$$\frac{\Pr(\mathbf{O}_{l-L+1}^l = \mathbf{o}_{l-L+1}^l|\mathbf{R}_1^l = \mathbf{r}_1^l)}{\Pr(\mathbf{O}_{l-L+1}^l = \mathbf{o}_{l-L+1}^l|\mathbf{R}_1^{l-1} = \mathbf{r}_1^{l-1})} \in [e^{-\epsilon}, e^{\epsilon}], \tag{25}$$

it is sufficient to show $\mathcal{M}_l^s$ makes the BIL defined in (8) upper bounded by ϵ.

Proof: For the privacy constraints (one term):

$$\begin{aligned}&\frac{\Pr(\mathbf{O}_1^l = \mathbf{o}_1^l|\mathbf{R}_1^l = \mathbf{r}_1^l)}{\Pr(\mathbf{O}_1^l = \mathbf{o}_1^l|\mathbf{R}_1^{l-1} = \mathbf{r}_1^{l-1})}\\ &= \mathsf{A}\cdot\frac{\Pr(\mathbf{O}_{l-L+1}^l = \mathbf{o}_{l-L+1}^l|\mathbf{R}_1^l = \mathbf{r}_1^l)}{\Pr(\mathbf{O}_{l-L+1}^l = \mathbf{o}_{l-L+1}^l|\mathbf{R}_1^{l-1} = \mathbf{r}_1^{l-1})},\end{aligned} \tag{26}$$

where

$$\begin{aligned}\mathsf{A} &= \frac{\Pr(\mathbf{O}_1^{l-L} = \mathbf{o}_1^{l-L}|\mathbf{R}_1^l = \mathbf{r}_1^l, \mathbf{O}_{l-L+1}^l = \mathbf{o}_{l-L+1}^l)}{\Pr(\mathbf{O}_1^{l-L} = \mathbf{o}_1^{l-L}|\mathbf{R}_1^{l-1} = \mathbf{r}_1^{l-1}, \mathbf{O}_{l-L+1}^l = \mathbf{o}_{l-L+1}^l)}\\ &= \frac{\Pr(\mathbf{O}_1^{l-L} = \mathbf{o}_1^{l-L}|\mathbf{R}_1^{l-1} = \mathbf{r}_1^{l-1}, \mathbf{O}_{l-L+1}^l = \mathbf{o}_{l-L+1}^l)}{\Pr(\mathbf{R}_l = \mathbf{r}_l|\mathbf{R}_1^{l-1} = \mathbf{r}_1^{l-1}, \mathbf{O}_{l-L+1}^l = \mathbf{o}_{l-L+1}^l)}\\ &\quad\cdot\frac{\Pr(\mathbf{R}_l = \mathbf{r}_l|\mathbf{R}_1^{l-1} = \mathbf{r}_1^{l-1}, \mathbf{O}_1^l = \mathbf{o}_1^l)}{\Pr(\mathbf{O}_1^{l-L} = \mathbf{o}_1^{l-L}|\mathbf{R}_1^{l-1} = \mathbf{r}_1^{l-1}, \mathbf{O}_{l-L+1}^l = \mathbf{o}_{l-L+1}^l)}\\ &= \frac{\Pr(\mathbf{R}_l = \mathbf{r}_l|\mathbf{R}_1^{l-1} = \mathbf{r}_1^{l-1}, \mathbf{O}_1^l = \mathbf{o}_1^l)}{\Pr(\mathbf{R}_l = \mathbf{r}_l|\mathbf{R}_1^{l-1} = \mathbf{r}_1^{l-1}, \mathbf{O}_{l-L+1}^l = \mathbf{o}_{l-L+1}^l)} = 1.\end{aligned} \tag{27}$$

The last equation holds because the new policy does not depend on the sequence of $\mathbf{O}_1^{l-L}$. Then, by considering the subset of policies, the BIL becomes:

$$\mathcal{L}(\mathbf{R}_l \to \mathbf{O}_{l-L+1}^l) = \max_{\mathbf{o}_{l-L+1}^l, \mathbf{r}_l} \left| \log \frac{\Pr(\mathbf{O}_{l-L+1}^l = \mathbf{o}_{l-L+1}^l | \mathbf{R}_1^l = \mathbf{r}_1^l)}{\Pr(\mathbf{O}_{l-L+1}^l = \mathbf{o}_{l-L+1}^l | \mathbf{R}_1^{l-1} = \mathbf{r}_1^{l-1})} \right|. \quad (28)$$

□

The utility function of BED, by adopting a policy in (2), can be expressed as:

$$\begin{aligned}
&E\left[D(Q(\mathbf{O}_l) - Q(\mathbf{R}_l)) | \mathbf{R}_1^{l-1} = \mathbf{r}_1^{l-1}\right] \\
&= \sum_{\mathbf{o},\mathbf{r}} D(Q(\mathbf{o}) - Q(\mathbf{r})) \\
&\quad \cdot \sum_{\mathbf{o}_{l-L+1}^{l-1}} \Pr(\mathbf{O}_l = o, \mathbf{O}_{l-L+1}^{l-1} = \mathbf{o}_{l-L+1}^{l-1}, R_l = r_l | \mathbf{R}_1^{l-1} = \mathbf{r}_1^{l-1}) \\
&= \sum_{\mathbf{o},\mathbf{r}} D(Q(\mathbf{o}) - Q(\mathbf{r})) \\
&\quad \cdot \sum_{\mathbf{o}_{l-L+1}^{l-1}} \{\Pr(R_l = r_l | \mathbf{R}_1^{l-1} = \mathbf{r}_1^{l-1}, \mathbf{O}_l = \mathbf{o}, \mathbf{O}_{l-L+1}^{l-1} = \mathbf{o}_{l-L+1}^{l-1}) \\
&\quad \cdot \Pr(\mathbf{O}_l = o, \mathbf{O}_{l-L+1}^{l-1} = \mathbf{o}_{l-L+1}^{l-1} | \mathbf{R}_1^{l-1} = \mathbf{r}_1^{l-1})\} \\
&= \sum_{\mathbf{o},\mathbf{r}} D(Q(\mathbf{o}) - Q(\mathbf{r})) \\
&\quad \cdot \sum_{\mathbf{o}_{l-L+1}^{l-1}} a_l^s(r_l | \mathbf{o}_{l-L+1}^{l-1}, o, \mathbf{r}_1^{l-1}) \beta_l^s(\mathbf{o}_{l-L+1}^{l-1}, o | \mathbf{r}_1^{l-1}). \quad (29)
\end{aligned}$$

On the other hand, the simplified belief state using (1) can be further expressed as:

$$\begin{aligned}
&\Pr(\mathbf{O}_{l-L+1}^l = \mathbf{o}_{l-L+1}^l | \mathbf{R}_1^{l-1} = \mathbf{r}_1^{l-1}) \\
&= C_{l-L+1}^l \Pr(\mathbf{O}_{l-L+1}^{l-1} = \mathbf{o}_{l-L+1}^{l-1} | \mathbf{R}_1^{l-1} = \mathbf{r}_1^{l-1}) \\
&= C_{l-L+1}^l \frac{\sum_{\mathbf{o}_{l-L}} a_{l-1}(\mathbf{r}_{l-1} | \mathbf{o}_{l-L}^{l-1}, \mathbf{r}_1^{l-2}) \beta_{l-1}^s(\mathbf{o}_{l-L}^{l-1} | \mathbf{R}_1^{l-2} = \mathbf{r}_1^{l-2})}{\sum_{\tilde{\mathbf{o}}_{l-L}^{l-1}} a_{l-1}(\mathbf{r}_{l-1} | \tilde{\mathbf{o}}_{l-L}^{l-1}, \mathbf{r}_1^{l-2}) \beta_{l-1}^s(\tilde{\mathbf{o}}_{l-L}^{l-1} | \mathbf{R}_1^{l-2} = \mathbf{r}_1^{l-2})} \\
&= C_{l-L+1}^l \frac{\sum_{\mathbf{o}_{l-L}} a_{l-1}^s(\mathbf{r}_{l-1} | \mathbf{o}_{l-L}^{l-1}, \mathbf{r}_1^{l-2}) \beta_{l-1}^s(\mathbf{o}_{l-L}^{l-1} | \mathbf{R}_1^{l-2} = \mathbf{r}_1^{l-2})}{\sum_{\tilde{\mathbf{o}}_{l-L}^{l-1}} a_{l-1}^s(\mathbf{r}_{l-1} | \tilde{\mathbf{o}}_{l-L}^{l-1}, \mathbf{r}_1^{l-2}) \beta_{l-1}^s(\tilde{\mathbf{o}}_{l-L}^{l-1} | \mathbf{R}_1^{l-2} = \mathbf{r}_1^{l-2})}. \quad (30)
\end{aligned}$$

where we use C in (30) to denote the correlation among batched sequences. The simplification allows us to reduce computation complexity without violating the privacy guarantee. In the following, we let $L = 1$. Note that by restricting the policy to only using a subset of historical input batches, the solution to (31) will be a sub-optimal solution of the original utility-privacy tradeoff formulation in (19). Hence, the simplified utility-privacy tradeoff of the batched model becomes:

$$\sum_{o,r} D(Q(\mathbf{o}) - Q(\mathbf{r})) a_l^s(\mathbf{r}_l | \mathbf{o}, \mathbf{r}_1^{l-1}) \beta_l^s(o | \mathbf{r}_1^{l-1}),$$

$$\text{Such that } \frac{a_l^s(\mathbf{r} | \mathbf{o}, \mathbf{r}_1^{l-1})}{\sum_{\tilde{\mathbf{o}}} a_l^s(\mathbf{r} | \tilde{\mathbf{o}}, \mathbf{r}_1^{l-1}) \beta_l^s(\tilde{\mathbf{o}} | \mathbf{r}_1^{l-1})} \in [e^{-\epsilon}, e^{\epsilon}]. \quad (31)$$

Notably, the objective function in (31) is a linear combination of $a_l^s(\mathbf{r} | \mathbf{o}, \mathbf{r}_1^{l-1})$ for all $\mathbf{r}, \mathbf{o}$, enabling the attainment of the global optimal solution through convergence-based algorithms such as the gradient descent. Herein, we utilize the gradient descent algorithm to numerically solve the optimization problem in (31), as outlined in Alg. 2. In essence, we initialize the perturbation parameters from a uniform distribution. In each iteration, we calculate a partial derivative of the utility function with respect to each parameter. Upon updating each a_l with a step length ϕ, we verify if the current parameters satisfy the privacy constraints. If they do not, we halt the update of the current parameters. Observe that the mechanism needs to update all parameters $\mathbf{o}, \mathbf{r} \in \mathcal{X}^w$. The computation complexity of Alg. 2. is $\mathcal{O}(|\mathcal{X}|^{2w})$.

Algorithm 2 SIP Mechanism for Batched Release

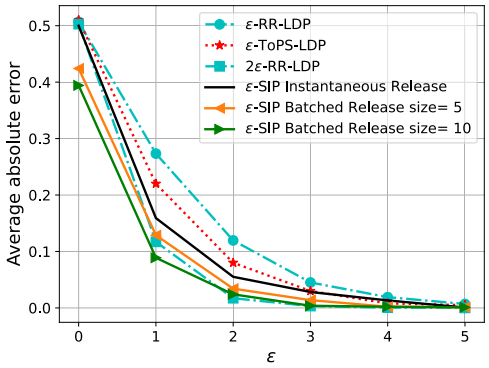
Input: current time l, initial prior $\mathrm{P}_{\mathbf{O}_1}(\mathbf{o})$, transitional matrix C_l^{l+1} for all $0 < l < B$, historical release sequence $\mathbf{R}_1^{l-1}$, current $\mathbf{o}_l$, Utility function U, step length ϕ.
Output: Batched release R_l
if $l \neq 1$:
    Update belief state for all o according to (30)
Initialize $a_l(\mathbf{r}_l | \mathbf{o}_l, \mathbf{r}_1^{l-1}) = 1/|\mathcal{X}|^w$ for all $\mathbf{o}_l$ and $\mathbf{r}_l$, $U^* = \infty$, $Act = \text{ones}(|\mathcal{X}|^w, |\mathcal{X}|^w])$.
While sum$(Act) \neq 0$:
  for $\mathbf{o} \in |\mathcal{X}|^w$:
    for $\mathbf{r} \in |\mathcal{X}|^w$:
    Calculate the derivative: $d(\mathbf{o}, \mathbf{r}) = \partial U / \partial a_l(\mathbf{r} | \mathbf{o}, \mathbf{r}_1^{l-1})$
      $a_l(\mathbf{r} | \mathbf{o}, \mathbf{r}_1^{l-1}) \leftarrow a_l(\mathbf{r} | \mathbf{o}, \mathbf{r}_1^{l-1}) + \phi \cdot d(\mathbf{o}, \mathbf{j})$
      Check privacy constraints, $s(\mathbf{o}, \mathbf{r}) = 0$
      for $\tilde{\mathbf{o}} \in |\mathcal{X}|^w$:
        $s(\mathbf{o}, \mathbf{r}) \leftarrow s(\mathbf{o}, \mathbf{r}) + a_l(\mathbf{r} | \tilde{\mathbf{o}}, \mathbf{r}_1^{l-1}) \beta_l(\tilde{\mathbf{o}} | \mathbf{r}_1^{l-1})$
      if $s(\mathbf{o}, \mathbf{r}) / [a_l(\mathbf{r} | \tilde{\mathbf{o}}, \mathbf{r}_1^{l-1}) \beta_l(\tilde{\mathbf{o}} | \mathbf{r}_1^{l-1})] \notin [e^{-\epsilon}, e^{\epsilon}]$:
        $Act[\mathbf{o}][\mathbf{r}] = 0$
Release $\mathbf{R}_l$ according to $a_l(\mathbf{r}_l | \mathbf{o}_l, \mathbf{r}_1^{l-1})$

VI. Experimental Results

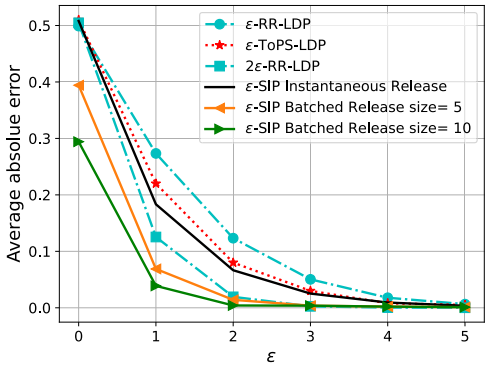
In this section, we evaluate the performance of our proposed mechanism and compare it with other solutions from related works.

A. Evaluations With Synthetic Data

In our first evaluation, we conduct a comparative analysis between the proposed instantaneous release mechanism, batched release mechanisms, and the following two mechanisms from the literature: 1) Randomized Response Mechanism Based on Local Differential Privacy (RR-LDP): This mechanism perturbs data values independently at each timestamp to satisfy ϵ_k-LDP. Thanks to the sequential composition theorem of LDP, the global leakage measured by LDP after time T is the sum of ϵ_k from $k = 1$ to T. 2) The LDP Mechanism from [19] (denoted as ToPS): This mechanism comprises three phases: threshold estimation, perturbation, and smoothing. Initially, the mechanism releases samples using the Laplacian mechanism and uses the first set of samples to estimate a threshold. A hierarchical method is then employed to handle the ranged values. Lastly, a post-processing (smoothing) phase ensures that the data sequence aligns with a



(a) Case 1. Utility-privacy tradeoff under weak data correlation



(b) Case2. Utility-privacy tradeoff under strong data correlation

Fig. 5. Privacy-utility tradeoff comparison for different data release settings using synthetic data.

specified distribution. Our comparison aims to measure the effectiveness of these techniques, highlighting their relative strengths and weaknesses in handling privacy-preserving data releases.

We first simulate by generating a binary time series that satisfies the first-order Markov chain with an initial probability of $P_1 = \Pr(X_1 = 1)$ and transitional matrix

$$\begin{bmatrix} q_{00} & q_{01} \\ q_{10} & q_{11} \end{bmatrix}$$

where $q_{ij} = \Pr(X_k = j | X_{k-1} = i)$. We then compare the mechanisms mentioned above under the following two settings: (Case 1) $P_1 = 0.5$, $q_{00} = q_{11} = 0.5$, which means the data in the sequence are more correlated with each other. (Case 2) $P_1 = 0.9$, $q_{00} = q_{11} = 0.9$, indicating the dependence in the sequence is strong. In each case, we sample $10,000$ ($T = 10,000$) data points using the transitional matrix to create a single time-series data sequence. We then assess the average absolute distance between the original data sequence and its perturbed counterpart. The perturbation process is repeated $1,000$ times to generate the perturbed sequence, from which we calculate the average distance for each mechanism.

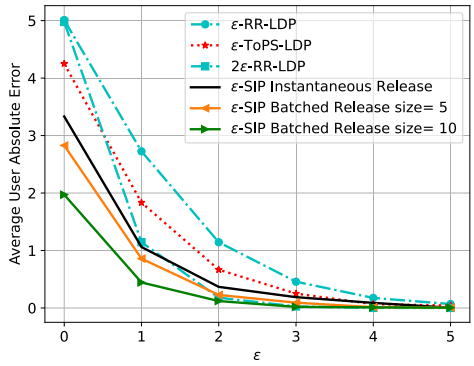
$$\sum_{i=1}^{T} |x_i - y_i| / T.$$

The comparison results are shown in Fig. 5. Observe that generally, the utility provided by the instantaneous release mechanism is sandwiched between ϵ and 2ϵ-RR-LDP mechanisms given any ϵ. However, as the data dependence increases, ϵ-SIP with instantaneous release even outperforms ϵ-LDP. On the other hand, batched release models always outperform instantaneous release models, and the advantage increases with the batch size. It is also worth noting that when the data correlation is strong. The advantages of context-aware models (SIP-based mechanisms) are even more obvious.

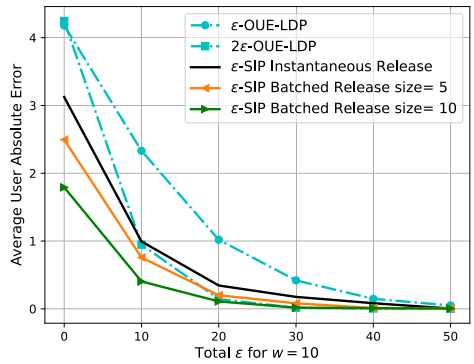
1) Click Streams Data (Kosarak): Investigating the clicking streams for a website can be very helpful in learning product popularity or guiding web page design. On the other hand, individuals' clicking data is privately sensitive as it may infer one's personal interest, working hours, daily behavior, etc.

a) Dataset: Kosarak is a dataset of click streams on a Hungarian website that contains around one million users and 41270 categories. The data is formatted so that each user clicks through multiple categories. After data cleaning and preprocessing, there are 990002 users in the dataset, each containing a data stream with lengths from 1 to 2498. We first extract each user's data stream and calculate the frequency of the occurrence of different patterns (every consecutive data value). For the instantaneous release model, we calculate the frequency of every pair of data values and estimate the conditional distribution from the frequency. Then for the batched release model, we first truncate the data streaming into several trucks according to the predefined batch size. Then we analyzed the frequency of every consecutive batch value. Then, we estimate the distribution from there. Finally, we summarized a frequency lookup table, which is then transferred to a correlation library. Given every possible current value, we are able to look up the following data within the correlation library with certain probabilities. Then we apply the mechanisms proposed in the previous section to the preprocessed model.

b) Utility and privacy: For our experiment with Kosarak, we randomly select 1000 users' data sequence as the input. Then we perturb these data sequences with the mechanisms proposed in the previous sections. The utility of this experiment is measured by the average user absolute distance between the input sequence and the output. Here we use "average user" to emphasize that the error is further aggregated among users. The privacy, on the other hand, is customized by varying the value of ϵ. We consider two budget allocation strategies, in case 1, we consider uniform budget allocation for each time step, with each privacy budget depicted on the x-axis. We compare with randomized response LDP mechanism (RR-LDP) with different privacy budgets. Further, we compare with the state-of-the-art LDP mechanism for continuous release, which is proposed in [19]. Then in case 2, we consider the advanced budget allocation algorithm proposed in [24], which adaptively assigns a privacy budget to each time step in a time window w, such that the released sequence in the time window satisfies w-event level LDP. The algorithm can be summarized as follows: Firstly, uniformly assign a privacy budget at the current time step according to the length of the time window and the total privacy budget left. Secondly, evenly split the assigned budget into two parts, ϵ_k^1 and ϵ_k^2. Then, privacy-preservingly estimates the dissimilarity between the previous release and current raw data using ϵ_k^1. Subsequently, compare with the error incurred at the current time step by adopting an ϵ_k^2-LDP mechanism. If the dissimilarity



(a) Comparison with uniform budget allocation and with Randomized Response LDP mechanism.



(b) Comparison with advanced budget allocation (Alg. 1 in [24]) and with Optimal Unary Encoding LDP mechanism [43].

Fig. 6. Utility privacy tradeoff comparison between different data release settings with Kosarak.

incurs less error, then directly release the previous output and set $\epsilon_k^2 = 0$, otherwise, release with the LDP mechanism with budget ϵ_k^2. Under the advanced budget allocation, we further compare with the Optimal Unary Encoding LDP (OUE-LDP) mechanism [43], which is shown to achieve better utility than RR-LDP for large input domains. The idea is to transfer the input value into a binary vector containing $|\mathcal{X}|$ bits, with the x bit to be 1 (x denotes the true value here), and all other values being 0. Then the mechanism perturbs each bit independently with optimized perturbation parameters. The comparison is shown in Fig.6.

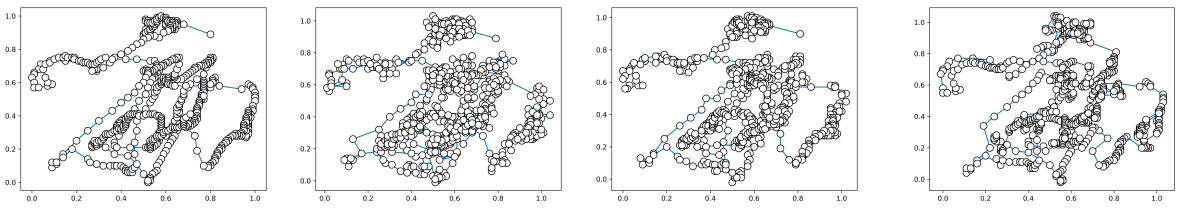
From the above figures, several key observations can be made: 1) Despite ϵ-SIP operating within the bounds of ϵ-LDP and 2ϵ-LDP, our proposed mechanisms can even outperform 2ϵ-LDP for some ϵs. 2) In the case of batched release models, we observe a trend where larger batch sizes contribute to improved utility. This concept also extends to the instantaneous release model, which can be considered a special case of batched release with a batch size of one. 3) While the Truncated Output Perturbation under Local Differential Privacy (ToPs-LDP) mechanism improves data utility compared to the Randomized Response under Local Differential Privacy (RR-LDP), primarily due to sensitivity reduction achieved through truncation, it still performs worse than our SIP-based mechanisms. This disparity stems from ToPs-LDP's inability to capture data correlation, a deficiency not present in our proposed mechanisms. 4) Employing the advanced budget allocation algorithm results in an enhancement of utility for each mechanism when compared to the uniform budget partitioning approach. Nevertheless, the advantages gained from integrating encoding-based optimization techniques within the LDP framework appear to be marginal. This phenomenon is primarily attributed to the characteristic structure of the output vector generated by the OUE-LDP mechanism, which tends to exhibit multiple 1s scattered across various positions. In the context of frequency estimation, this multiplicity of 1s does not present a significant concern. The reason is that the estimated frequency converges to the true mean as multiple users' data is aggregated. Moreover, any deviation from the actual frequency can be further reduced during the post-processing phase. Conversely, when dealing with utility metrics that are distance-based, these metrics necessitate the selection of a singular value from the output for representation. Consequently, the act of randomly choosing from multiple positions laden with 1s compromises the precision of the utility measure.

2) Eye-Tracking Data: Eye-tracking data is usually collected online by AR/VR devices. Usually, cameras are embedded in these devices to track users' eyeball moving, and these axis data are uploaded to the server in return for services, such as online video games, online social, etc. However, studies have figured out several privacy concerns in eye-tracking data consumption: In [44], Steil et al. pointed out that eye-tracking data can reveal one's private sensitive information, such as age, gender, health status, sexual orientation, personal trails, etc. Further, a group of researchers from Stanford University have shown they're able to reliably identify individuals Using a pool of 511 participants. Their system is said to be capable of identifying 95% of users correctly "when trained on less than 5 min of tracking data per person [45].

a) Dataset: In this experiment, we compare the performance of our mechanisms with independent LDP-based mechanisms with the dataset of "MOJO", which is collected by "Mojo Vision". The Mojo dataset contains ten users' eye-gazing data. Each user's data sequence can be viewed as a three-dimensional vector containing X, Y, and Z axes, where Z label measures image rotation, and we ignore this factor in the following experiment. The length of each individual's eye-gazing data sequence is different according to his/her recorded period. On average, each one of them has been collected for 5 hours, with a sample rate of 50 per second. We first normalize the coordinates to be within [0, 1]. Then we equally divided the area into a 100×100 space. Then we quantify each coordinate and cast the coordinates to the grid. We take a portion of one individual's normalized and quantified eye gazing sequence (1/1000) of the raw data sequence.

We assume the correlation within the data stream has Markov properties. Further, we assume a first-order and second-order Markov chain in the stream. Then the correlation is measured by the conditional probabilities, which can be estimated by frequency checking in the eye-gazing stream. We assume the initial probability is uniformly distributed.

We consider three types of data release mechanisms: (1) Temporal LDP mechanism proposed in [29]. The previous ToPS mechanism is generally utilized for releasing one-dimensional data. However, eye-tracking data encompasses



(a) Pre-processed data (b) Independent RR-LDP (c) Correlated Instantaneous release (d) Correlated Batched release

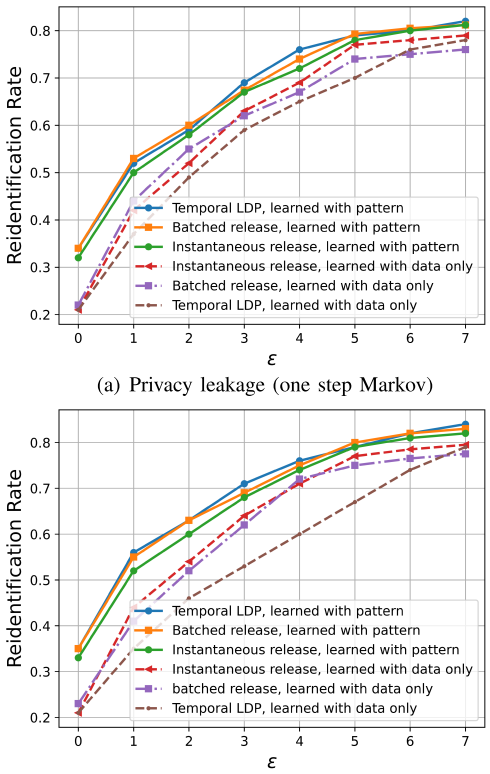
Fig. 7. Visualization of the eye tracking data: (a) quantized normalized eye-gazing data; (b) Release from independent randomized response LDP mechanisms (c) Release from correlated randomized response mechanism with the instantaneous release mechanism, (d) Release from correlated randomized response mechanism with the batched sequential release mechanism (batch size =10). For (b),(c),(d), total budget $\epsilon = 10$;.

both x and y axes, aligning more closely with location data. Therefore, it is more appropriate to compare with Temporal LDP, which is a mechanism originally proposed specifically for the release of location data. We first assign a total budget $\epsilon = 10$ to the data sequence every second. Then, the local privacy budget can be calculated by the sequential composition Theorem. Note that support of the randomized response mechanism is selected according to the realizations with non-zero conditional probabilities given the previous values (according to the Markov properties). (2) Conditional randomized response mechanism for the instantaneous release model. The parameters involved in this mechanism are summarized in this paper. (3) Batched data release mechanism, where we assume the batch sizes are 5 and 10, respectively. The perturbation parameters are calculated by numerically solving the optimization problem defined in (31). A visualization of the release of different mechanisms is shown in Fig. 7.

b) Privacy leakage: Since the privacy protection guarantees provided by different mechanisms are different, we want to see how they protect the re-identification rate of each individual under fixed epsilons. The re-identification rate refers to the probability that an attacker successfully re-identifies the user by observing his eye-tracking data. This technique is typically achieved through deep learning. We first design RNN models that leverage data correlations in predictions. Models are trained with half of the eye-gazing data streams from each user. The goal is to learn the eye-moving patterns of each individual. Then, we consider two scenarios regarding the RNN's prediction model: one considers the data correlation estimated from frequency (for the whole data stream) and leverages it in the data prediction; the other one makes predictions solely depending on the observed data.

The privacy leakage comparison of different models is shown in Fig. 8. Observe that when the reidentification model is trained with data only, the temporal LDP-based mechanism provides the most strict privacy protection. When the adversary trains the model with the pattern information, while the overall leakage of all mechanisms increases, the increase of SIP-based mechanisms is relatively smaller, and the leakage of SIP-based mechanisms is even smaller than LDP-based mechanisms. The reason is prior information has already been considered in the definition of SIP and noise is added more effectively to mitigate the privacy leakage against such adversaries.

c) Query utility: The utility of different mechanisms is measured by the average Euclidean distance between the raw



(a) Privacy leakage (one step Markov)

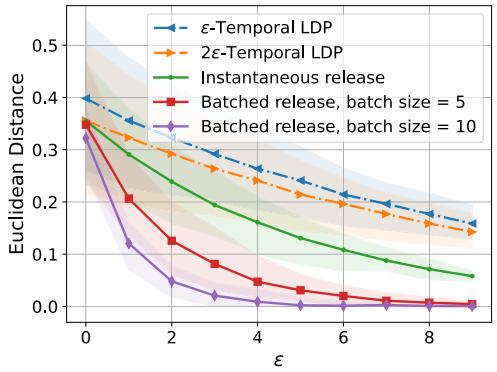
(b) Privacy leakage (two-step Markov)

Fig. 8. Privacy leakage comparisons, privacy evaluated by the identification rate of individuals in the dataset; (a) considers a one-step Markov Chain, and (b) considers a two-step Markov Chain.

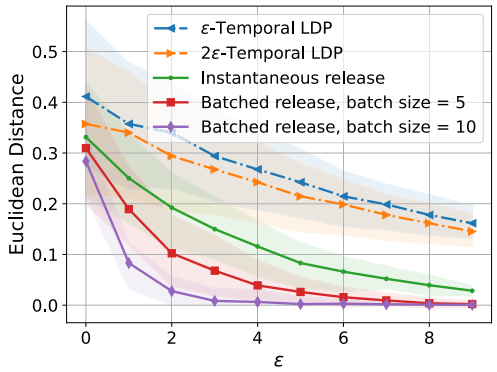
and released data at each time stamp. The utility comparisons of different models are presented in Fig. 9.

Observe that even though different mechanisms achieve similar privacy protections, their utilities are very different. SIP-based batched release models outperform other models significantly and the gap is even larger with bigger batch sizes. On the other hand, the instantaneous release model provides better utility even than the 2ϵ-temporal LDP. This is because even though temporal LDP leverages correlations to reduce sensitivity, the definition and the mechanism do not consider the correlation, while SIP is totally context-aware.

d) Run time comparison: We compare the computational efficiency of various mechanisms discussed. Given that the complexity does not depend on the parameter ϵ, we combine the cases of ϵ-Temporal LDP and 2ϵ-Temporal LDP into a



(a) Utility measure (one step Markov)



(b) Utility measure (two-step Markov)

Fig. 9. Utility comparisons, utility measured by the averaged Euclidean distance between the raw and the released data at each time step. (a) Considers a one-step Markov Chain, and (b) considers a two-step Markov Chain. The shaded area represents a 95% confidence interval.

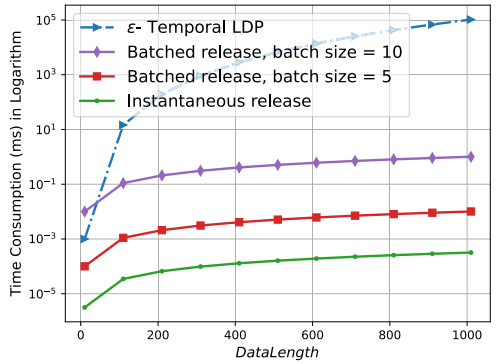


Fig. 10. Run time comparison of different mechanisms.

single category. Then, we vary the length of the eye-tracking data from 10 to 1000. The outcomes are shown in Figure 10. Both the batched release and instantaneous release mechanisms demonstrate a linear increase in run time to the data length, i.e. order of $\mathcal{O}(N)$. On the other hand, the run time for Temporal LDP exhibits a substantially steeper growth, which is in accordance with $\mathcal{O}(N^4)$ as presented in the computation complexity analysis in Table I.

VII. Discussions

Firstly, SIP offers a notable advancement in enhancing the utility-privacy tradeoff compared to LDP. This implies that to provide the same level of utility, SIP requires a much lower privacy budget compared to LDP. Hence, SIP requires less total budget compared with LDP for the same data sequence length. As demonstrated in experiments using eye-tracking data, the utility enhancement of SIP does not sacrifice its privacy protection.

Secondly, SIP's robustness merits discussion. In scenarios where data distribution or correlations are inaccurate, the estimated privacy protection might not be precise, potentially leading to conservative privacy leakage under SIP with a risk of higher actual data leakage. This discrepancy, however, can be bounded by a factor dependent on the difference between the actual and estimated data distributions as discussed in [18]. Additionally, adopting a $\epsilon/2$-SIP mechanism, which implies ϵ-LDP, ensures that the leakage under LDP is always bounded by ϵ. While $\epsilon/2$-SIP potentially provides better utility for high privacy regime (ϵ close to 0).

Lastly, we address the issue of privacy budget exhaustion. Due to the linear composition property in SIP, the total privacy budget might be exhausted rapidly, especially with a large data sequence length. To mitigate this, we suggest three strategies: 1. Tighter composition analysis, such as formulating approximate SIP's privacy loss distribution to derive a more accurate privacy profile (ϵ and δ tradeoff). 2. Releasing sampled results, with a random selection of sampled time steps to lower sensitivity. 3. Implementing time windows for data sequences, with SIP protecting data within each window. Then refresh the budget after each time window. This setting is valid especially when recent data are more sensitive than a long time ago, and the window size can be tailored based on the data's time sensitivity or utility requirements. These approaches can be potentially combined for practical applications.

VIII. Conclusion and Future Works

In this paper, we tackle the challenge of releasing streaming data while preserving privacy. Initially, we introduce Sequence Information Privacy (SIP), which is an extension of local information privacy to sequential data. SIP inherently acknowledges data correlations within its definition. Subsequently, we study its relationships with existing privacy concepts such as Local Differential Privacy and Pufferfish Privacy. We demonstrate that our SIP can be sequentially decomposed into individual local leakages, making the optimization of global utility-privacy tradeoff equivalent to independently solving each local instance. Based on two data release settings, instantaneous and batched release, we propose perturbation mechanisms that optimize this utility-privacy tradeoff. Our evaluation, using both synthetic and real data, compares the utility-privacy tradeoffs provided by our proposed mechanisms with those from existing works. Results indicate that our mechanisms can significantly enhance data utility without compromising data privacy.

In terms of future work, we are interested in the following directions. One direction is to remove the assumptions that the prior distribution and data correlation are known. We can make the mechanism release the first several data points in a context-free manner (for example, using LDP). As more observations are made, the data prior/correlation becomes more certain, and the mechanism leverages the context to achieve context-aware utility-privacy tradeoffs. Another direction to

consider is that the batched release mechanism still involves high computational complexity. We would like to investigate further ways to reduce the computational complexity without violating privacy constraints.

REFERENCES

[1] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the Internet of Things: Security and privacy issues," *IEEE Internet Comput.*, vol. 21, no. 2, pp. 34–42, Mar. 2017.

[2] B. Beaulieu-Jones et al., "Privacy-preserving generative deep neural networks support clinical data sharing," *Circulat., Cardiovascular Quality Outcomes*, vol. 12, no. 7, Dec. 2018, Art. no. e005122.

[3] C. Dwork, "Differential privacy," in *Proc. Int. Colloq. Automata, Lang. Program.*, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds. 2006, pp. 1–12.

[4] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory Cryptogr. Conf.*, 2006, pp. 265–284.

[5] C. Dwork, "Differential privacy: A survey of results," in *Proc. Int. Conf. Theory Appl. Models Comput. (TAMC)*, M. Agrawal, D. Du, and Z. Duan, Eds. 2008, pp. 1–19.

[6] A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber, "Privacy: Theory meets practice on the map," in *Proc. IEEE 24th Int. Conf. Data Eng.*, Apr. 2008, pp. 277–286.

[7] J. Abowd, "The US Census Bureau adopts differential privacy," in *Proc. 24th Int. Conf. Knowl. Discovery Data Mining*, London, U.K., Jul. 2018, p. 2867.

[8] Ú. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: Randomized aggregatable privacy-preserving ordinal response," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2014, pp. 1054–1067.

[9] G. Cormode, S. Jha, T. Kulkarni, N. Li, D. Srivastava, and T. Wang, "Privacy at scale: Local differential privacy in practice," in *Proc. Int. Conf. Manag. Data*, New York, NY, USA, May 2018, pp. 1655–1658.

[10] Apple DP Team. *Learning With Privacy at Scale*. Accessed: 2017. [Online]. Available: https://machinelearning.apple.com/2017/12/06/learning-with-privacy-at-scale.html

[11] D. Kifer and A. Machanavajjhala, "No free lunch in data privacy," in *Proc. SIGMOD PODS*, 2011, pp. 193–204.

[12] D. Kifer and A. Machanavajjhala, "A rigorous and customizable framework for privacy," in *Proc. 31st ACM SIGMOD-SIGACT-SIGAI Symp. Princ. Database Syst.*, May 2012, pp. 77–88.

[13] D. Kifer and A. Machanavajjhala, "Pufferfish: A framework for mathematical privacy definitions," *ACM Trans. Database Syst.*, vol. 39, pp. 1–36, Jan. 2014.

[14] C. Huang, P. Kairouz, X. Chen, L. Sankar, and R. Rajagopal, "Context-aware generative adversarial privacy," *Entropy*, vol. 19, no. 12, p. 656, 2017.

[15] W. Wang, L. Ying, and J. Zhang, "On the relation between identifiability, differential privacy, and mutual-information privacy," *IEEE Trans. Inf. Theory*, vol. 62, no. 9, pp. 5018–5029, Sep. 2016.

[16] P. Cuff and L. Yu, "Differential privacy as a mutual information constraint," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 43–54.

[17] B. Jiang, M. Li, and R. Tandon, "Local information privacy and its application to privacy-preserving data aggregation," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 3, pp. 1918–1935, May/Jun. 2022.

[18] B. Jiang, M. Seif, R. Tandon, and M. Li, "Context-aware local information privacy," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3694–3708, 2021.

[19] T. Wang et al., "Continuous release of data streams under both centralized and local differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, New York, NY, USA, 2021, pp. 1237–1253.

[20] S. Song, Y. Wang, and K. Chaudhuri, "Pufferfish privacy mechanisms for correlated data," in *Proc. ACM Int. Conf. Manag. Data*. New York, NY, USA: Association for Computing Machinery, May 2017, pp. 1291–1306.

[21] W. Zhang, B. Jiang, M. Li, and X. Lin, "Privacy-preserving aggregate mobility data release: An information-theoretic deep reinforcement learning approach," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 849–864, 2022.

[22] P. Kairouz, S. Oh, and P. Viswanath, "The composition theorem for differential privacy," *IEEE Trans. Inf. Theory*, vol. 63, no. 6, pp. 4037–4049, Jun. 2017.

[23] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential privacy under continual observation," in *Proc. 42nd ACM Symp. Theory Comput.*, New York, NY, USA, Jun. 2010, pp. 715–724.

[24] X. Ren, L. Shi, W. Yu, S. Yang, C. Zhao, and Z. Xu, "LDP-IDS: Local differential privacy for infinite data streams," in *Proc. Int. Conf. Manage. Data*. New York, NY, USA: Association for Computing Machinery, Jun. 2022, pp. 1064–1077.

[25] M. Joseph, A. Roth, J. Ullman, and B. Waggoner, "Local differential privacy for evolving data," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, 2018, pp. 1–10.

[26] Q. Xue, Q. Ye, H. Hu, Y. Zhu, and J. Wang, "DDRM: A continual frequency estimation mechanism with local differential privacy," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 7, pp. 6784–6797, Jul. 2023.

[27] D. Vatsalan, R. Bhaskar, and M. A. Kaafar, "Local differentially private fuzzy counting in stream data using probabilistic data structures," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 8, pp. 8185–8198, Aug. 2023.

[28] P. Jain, S. Raskhodnikova, S. Sivakumar, and A. D. Smith, "The price of differential privacy under continual observation," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 14654–14678.

[29] Y. Cao, M. Yoshikawa, Y. Xiao, and L. Xiong, "Quantifying differential privacy under temporal correlations," in *Proc. IEEE 33rd Int. Conf. Data Eng. (ICDE)*, Apr. 2017, pp. 821–832.

[30] Y. Xiao and L. Xiong, "Protecting locations with differential privacy under temporal correlations," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, New York, NY, USA, Oct. 2015, pp. 1298–1309.

[31] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux, "Quantifying location privacy," in *Proc. IEEE Symp. Secur. Privacy*, May 2011, pp. 247–262.

[32] G. Theodorakopoulos, R. Shokri, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec, "Prolonging the hide-and-seek game: Optimal trajectory privacy for location-based services," in *Proc. 13th Workshop Privacy Electron. Soc.*, New York, NY, USA, 2014, pp. 73–82.

[33] T. Zhu, P. Xiong, G. Li, and W. Zhou, "Correlated differential privacy: Hiding information in non-IID data set," *IEEE Trans. Inf. Forensics Security*, vol. 10, pp. 229–242, 2015.

[34] E. Bao, Y. Yang, X. Xiao, and B. Ding, "CGM: An enhanced mechanism for streaming data collection with local differential privacy," *Proc. VLDB Endowment*, vol. 14, no. 11, pp. 2258–2270, Jul. 2021.

[35] Q. Ye, H. Hu, N. Li, X. Meng, H. Zheng, and H. Yan, "Beyond value perturbation: Local differential privacy in the temporal setting," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2021, pp. 1–10.

[36] X. Zhang, M. M. Khalili, and M. Liu, "Differentially private real-time release of sequential data," *ACM Trans. Privacy Secur.*, vol. 26, no. 1, pp. 1–29, Nov. 2022.

[37] F. D. P. Calmon, A. Makhdoumi, M. Médard, M. Varia, M. Christiansen, and K. R. Duffy, "Principal inertia components and applications," *IEEE Trans. Inf. Theory*, vol. 63, no. 8, pp. 5011–5038, Aug. 2017.

[38] H. Wang and F. P. Calmon, "An estimation-theoretic view of privacy," in *Proc. 55th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Oct. 2017, pp. 886–893.

[39] S. Asoodeh, M. Diaz, F. Alajaji, and T. Linder, "Estimation efficiency under privacy constraints," *IEEE Trans. Inf. Theory*, vol. 65, no. 3, pp. 1512–1534, Mar. 2019.

[40] C. Y. T. Ma and D. K. Y. Yau, "On information-theoretic measures for quantifying privacy protection of time-series data," in *Proc. 10th ACM Symp. Inf., Comput. Commun. Secur.*, New York, NY, USA, Apr. 2015, pp. 427–438.

[41] E. Erdemir, P. L. Dragotti, and D. Gündüz, "Privacy-aware time-series data sharing with deep reinforcement learning," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 389–401, 2021.

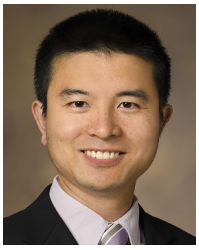
[42] F. D. P. Calmon and N. Fawaz, "Privacy against statistical inference," in *Proc. 50th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Oct. 2012, pp. 1401–1408.

[43] T. Wang, J. Blocki, N. Li, and S. Jha, "Locally differentially private protocols for frequency estimation," in *Proc. 26th USENIX Secur. Symp.*, Vancouver, BC, USA, Aug. 2017, pp. 729–745. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/wang-tianhao

[44] J. Steil, I. Hagestedt, M. X. Huang, and A. Bulling, "Privacy-aware eye tracking using differential privacy," in *Proc. 11th ACM Symp. Eye Tracking Res. Appl.*, New York, NY, USA, 2019, pp. 1–9.

[45] M. R. Miller, F. Herrera, H. Jun, J. A. Landay, and J. N. Bailenson, "Personal identifiability of user tracking data during observation of 360-degree VR video," *Sci. Rep.*, vol. 10, no. 1, p. 17404, Oct. 2020.

Bo Jiang received the bachelor's and master's degrees from Harbin Institute of Technology, China, in 2013 and 2015, respectively, the master's degree from Worcester Polytechnic Institute, MA, USA, in 2017, and the Ph.D. degree in ECE from The University of Arizona in 2023. He is currently a Research Scientist with the Privacy Innovation Laboratory, TikTok Inc. His current research interests include privacy-enhancing technologies, machine learning, radar image processing, and signal processing. This work was completed during his Ph.D. studies.



Ming Li (Fellow, IEEE) received the Ph.D. degree in ECE from Worcester Polytechnic Institute, MA, USA, in 2011. He was an Assistant Professor with the CS Department, Utah State University, from 2011 to 2015. He is currently an Associate Professor of ECE (and affiliated with CS) with The University of Arizona. He has published more than 135 journals and conference papers, with an H-index of 43. His research interests include wireless networks and security, privacy-enhancing technologies, and cyber-physical system security. He is a member of ACM. He received the NSF CAREER Award in 2014, the ONR YIP Award in 2016, and several paper awards, including the Best Paper Award from ACM WiSec in 2020. He was the TPC Co-Chair of IEEE CNS 2022. He served on the editorial boards for IEEE TRANSACTIONS ON MOBILE COMPUTING and IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING. He is currently an Associate Editor of IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY.

Ravi Tandon (Senior Member, IEEE) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology Kanpur (IIT Kanpur) in 2004 and the Ph.D. degree in electrical and computer engineering from the University of Maryland, College Park (UMCP), in 2010. He is currently the Litton Industries John M. Leonis Distinguished Associate Professor with the Department of ECE, The University of Arizona. Prior to joining The University of Arizona in Fall 2015, he was a Research Assistant Professor at Virginia Tech with positions at the Bradley Department of ECE, Hume Center for National Security and Technology; and the Discovery Analytics Center, Department of Computer Science. From 2010 to 2012, he was a Post-Doctoral Research Associate with Princeton University. His current research interests include information theory and its applications to machine learning, wireless networks, signal processing, communications, security, and privacy. He was a recipient of the 2018 Keysight Early Career Professor Award, the NSF CAREER Award in 2017, and the Best Paper Award at IEEE Globecom in 2011. He has served as an Editor for IEEE TRANSACTIONS ON INFORMATION THEORY, IEEE TRANSACTIONS ON COMMUNICATIONS, and IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.