Chained-DP: Can We Recycle Privacy Budget?

Jingyi Li Guangjing Huang, Liekang Zeng, Lin Chen and Xu Chen

***Abstract*—Privacy-preserving vector mean estimation is a crucial primitive in federated analytics. Existing practices usually resort to Local Differentiated Privacy (LDP) mechanisms that inject random noise into users' vectors when communicating with users and the central server. Due to the privacy-utility trade-off, the privacy budget has been widely recognized as the bottleneck resource that requires well provisioning. In this paper, we explore the possibility of privacy budget recycling and propose a novel Chained-DP framework enabling users to carry out data aggregation sequentially to recycle the privacy budget. We establish a sequential game to model the user interactions in our framework. We theoretically show the mathematical nature of the sequential game, solve its Nash Equilibrium, and design an incentive mechanism with provable economic properties. To alleviate potential privacy collusion attacks, we further derive a differentially privacy-guaranteed protocol to avoid holistic exposure. Our numerical simulation validates the effectiveness of Chained-DP, showing that it can significantly save privacy budget as well as lower estimation error compared to the traditional LDP mechanism.**

***Index Terms*—privacy preservation, differential privacy, game theory**

I. Introduction

A. Background and motivation

Vector mean estimation is a key operation and basic building block in many applications, e.g., federated learning [1] and frequency estimation [2]. In federated learning, in each training round, each user train machine learning models locally, and then upload the trained parameters vector to the server. The server aggregates the received parameters by vector mean estimation. Frequency estimation can also be regarded as a special case of vector mean estimation where each user owns a binary vector indicating whether the user owns each of the items in some universe, and the server wants to estimate the frequency of each item. Many of these applications are being widely deployed by companies such as Apple, Google, and Microsoft [3].

However, in many scenarios, the users' vectors are privacy-sensitive, and directly uploading compromises users' privacy. For example, in the above-mentioned federated learning, even though the users do not send original raw data to the server, the server can yet recover sensitive information from the received parameters [4]. From the user side, one is typically willing to accept a human-centric service by sharing its general preference but is loath to exact favor on specific items (e.g., movie ratings). Indeed, many countries and regions have enacted privacy protection-related laws, e.g., General Data Protection Regulation (GDPR) in European Union, and California Consumer Privacy Act (CCPA) in California, US. Both of the above inclinations drive the service providers to seriously take users' privacy into their Quality of Service (QoS) account.

A line of recent work has focused on enabling privacy-preserving vector mean estimation. In this context, a standard widely accepted privacy notion is Differential Privacy (DP) [5]. The key idea of DP is to provide strong privacy guarantees by injecting tunable levels of noise into the data before release while maintaining a proper trade-off between privacy and statistical utility. DP is a very strong statement that does not assume any background knowledge of threat models. Based on the *trust boundary*, there are two kinds of DP, central DP (CDP) and local DP (LDP) [6]. In CDP, the users upload the non-perturbed data to a trusted server, and the server injects noise and releases it after aggregation. In the contrast, LDP allows each user to inject noise locally. It is oblivious that LDP is more piratical to vector mean estimation than CDP in modern user-server style applications. **However, the server is usually malicious or semi-trusted, rendering it impractical in many scenarios.** Further, a lot of excellent works developed a series of LDP mechanisms for vector mean estimation, i.e., [7], [8]. These works aim to minimize estimation error and/or communication complexity. The state-of-the-art achieves the optimal error lower bounds with succinct communication for k-sparse vector [8]. However, compared to noise-free estimation and CDP mechanisms, **LDP mechanisms still introduce undesired estimation error.**

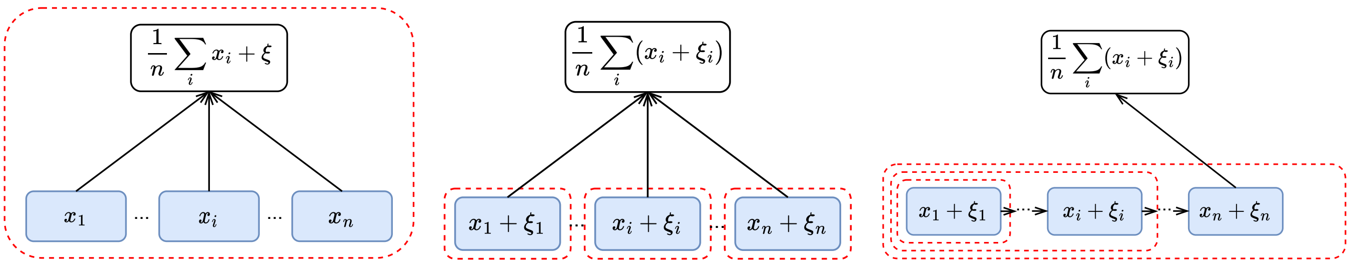
B. Fundamental Research Questions

Driven by the intrinsic limitation of LDP, we embark in this paper to investigate the following fundamental research questions:

Q1. *Is it possible to recycle privacy budget?*

Due to the privacy-utility trade-off, the privacy budget is a tight resource in differentially private mechanisms. If we can recycle the privacy budget, i.e., reuse the noise, the mechanisms will be more efficient. *Our intuition is the noise added by preceding users not only preserves their own privacy but also can be utilized by the later users in a sequence to fulfill privacy requirements such that the later users would like to inject less noise.* As a result, the overall noise added to the aggregated data will be lower, i.e., the accuracy will be higher. The difference between LDP and CDP is the trust boundary, which is shown in Fig.1(a)(b). It can be seen that the trust boundaries in CDP and LDP are extreme cases. CDP needs the users fully trust the interactive object (i.e., the server) but LDP assumes the users trust nobody else. In this paper, we propose a compromised diagram of Chained-DP which forms users into sequences and allows users to send their noisy data to other users, and the entire process is carried out user by user sequentially, as shown in Fig.1(c).

Jingyi Li, Guangjing Huang, Liekang Zeng, Lin Chen and Xu Chen are with School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China; Email: {lijy573, huanggj27, zenglk3}@mail2.sysu.edu.cn, chen@lri.fr, chenxu35@mail.sysu.edu.cn.



(a) CDP (b) LDP (c) Chained-DP

Fig. 1: Diagram of CDP, LDP, and the proposed Chained-DP. The red dotted line indicates the trust boundary. x_1, x_i, x_n are users' vectors and $\xi_1, \xi_i, , \xi_n, \xi$ are added noises.

Q2. *How do selfish users with different preferences behave in such a process?*

To answer this question, we provide a game-theoretical analysis of Chained-DP. We model the proposed noise-adding process as a sequential game where each user is a rational and selfish game player with different privacy-utility preferences. We solve the Nash Equilibrium (NE) of the sequential game in the complete information scenario. Further, we design an incentive mechanism, namely Broad And Shuffle (BAS) mechanism, which satisfies a multitude of desired properties without the complete information requirement. It turns out that the first user adds a very larger noise and the following users add nothing under NE. In addition, compared to LDP, Chained-DP introduces more communication delay. It is necessary to split a long sequence into sub-sequences to reduce communication delay. We give the guideline to split, i.e., users with similar utility-accuracy preferences are more suitable to be arranged in a sequence. The numerical results show that even being splitted, the estimation error of Chained-DP is always less than LDP.

Q3. *How to mitigate collusion attacks?*

Recall that one user in the sequentially noise-adding process may inject less noise than LDP or even no noise. If two users immediately before or after the user collude, it will lead to a high risk of privacy collusion attack. To tackle this challenge, we propose a Slice And Recombine (SAR) protocol that enforces different partitions of users' vectors to be arranged in different sequences randomly, which can significantly reduce the risk of overall vector exposure. Then, we derive the differential privacy guarantee provided by the protocol.

It is worth noting that Chained-DP is different from the classic sequential composition [5]. The sequential composition is to give the privacy loss upper bound of multiple times differentially releasing of one dataset (multiple releases). Alternatively, Chained-DP emphasizes injecting noise multiple times into one release (noise superposition on one release). In addition, it can be seen from Fig. 1 that in CDP and LDP, the devices need to communicate with the server several times, while Chained-DP only needs less. This means Chained-DP is more suitable for Device-to-Device Communications, Edge computing, and other distributed scenarios. It can exchange high-quality data locally between devices through LAN to achieve estimation accuracy and reduce expensive device-server communication overhead while meeting the requirements of privacy protection.

C. Key Contributions

The main contributions of this paper are as follows:

- To our best knowledge, this work is the first specific attempt to explore the possibility of recycling noises in a sequence mode. In such a diagram, the noise added by one user is also beneficial to the privacy of subsequent users. As a result, some users may add less noise, which lowers the overall noise level and shrinks the estimation error.
- We model the noise-adding process as a sequential game and analyze the process based on its Nash Equilibrium. More importantly, we design a simple but effective incentive mechanism, i.e, BAS mechanism, that satisfies four economic properties including truthfulness, fairness, voluntary participation, and efficiency. The proof of voluntary participation also reveals the superiority of Chained-DP compared to LDP.
- To mitigate the collusion attacks, we propose a practical privacy-preserving protocol, i.e., SAR protocol. Further, we derive the privacy guarantee provided by the protocol. It is worth noting that the protocol can be utilized in any sequence form wherein the first user has the priority to add maximal or vast majority of noise and users add zero or very little noise.

II. Related Work

Privacy-preserving federated learning. Privacy-preserving federated learning (PPFL) is an area of research that aims to address these privacy concerns by developing techniques that allow federated learning to be conducted while protecting the privacy of the data used for training. Except for the LDP, the other two kinds of popular approaches to PPFL are secure multi-party computation (MPC) [9] and homomorphic encryption (HE) [10]. However, these two kinds of methods usually introduce exponential overhead and may need extra special hardware to support them.

Privacy Budget Management. Privacy budget is an essential aspect of differential privacy, as it determines the trade-off between privacy and accuracy in privacy-preserving data analysis. Traditionally, it is regarded as a kind of non-renewable resource. Once it is used, the DP budget is forever consumed. Some works try to derive a tighter bound to reduce the budget based on statistical technologies, i.e. [11]. A line of recent works takes their effort to pack, allocate and manage privacy budget by modeling the privacy-utility trade-off as optimization problems in different settings, i.e., [12], [13]. Especially, [14] design a container-based real system scheduler to manage privacy budget. The closest work to Chained-DP is [15], which mentioned the concept of noise reusing in the privacy-preserving blockchain. However, it did not explore the feasibility specifically.

III. PRELIMINARIES

In this section, we will introduce some technical preliminaries to differential privacy first. Then, we give a corollary based on the definition of differential privacy to show the privacy budget of "multiple adding, single releasing".

Differential privacy is a mathematical framework defined for privacy-preserving data analysis, which aims at providing privacy guarantees for sensitive data and is regarded as a standard notion for rigorous privacy. The formal definition of (ϵ,δ)-local differential privacy (LDP) is as follows.

Definition 1. *((ϵ,δ)-LDP). A randomized mechanism $\mathcal{M} : \mathcal{X} \rightarrow \mathbb{R}^d$ is (ϵ,δ)-LDP if for any pair $x, x' \in \mathcal{X}$ and any measurable subset $S \subseteq Range(\mathcal{M})$, we have*

$$Pr[\mathcal{M}(x) \in S] \leq e^{\epsilon} \cdot Pr[\mathcal{M}(x') \in S] + \delta. \tag{1}$$

Roughly speaking, given a pair of neighborhood inputs, the definition requires the difference of the output of a randomized mechanism can be bounded by e^{ϵ} with high probability of $1 - \delta$. ϵ is the privacy budget which refers to the privacy level. When the value of ϵ becomes larger, privacy protection becomes weaker. δ is a relaxation factor and is generally very small. The case of $\delta = 0$ is called pure ϵ-LDP.

The well-known noise-adding privacy-persevering technique to achieve (ϵ,δ)-LDP is known as Gaussian mechanism, as defined next.

Definition 2. *(Gaussian Mechanism) Suppose a user wants to release a function $f(X)$ of an input X subject to (ϵ, δ)-LDP. The Gaussian release mechanism is defined as:*

$$\mathcal{M}(X) = f(X) + \mathcal{N}(0, \sigma^2\mathbf{I}), \tag{2}$$

where

$$\sigma = \frac{\Delta_f}{\epsilon}\sqrt{2\log\frac{1.25}{\delta}}. \tag{3}$$

If the sensitivity of the function is bounded by Δ_f, i.e., $\|f(x) - f(x')\|_2 \leq \Delta_f,\ \forall x, x'$, then for any $\delta \in (0, 1]$, Gaussian mechanism satisfies (ϵ, δ)-LDP.

Then, based on the Gaussian mechanism, we give the following corollary.

Corollary 1. *Given function $f(\cdot)$ bounded by Δ_f, i.e., $\|f(x) - f(x')\|_2 \leq \Delta_f,\ \forall x, x'$, suppose a user releases a perturbed function of an input X,*

$$\mathcal{M}(X) = f(X) + \sum_i \xi_i, \tag{4}$$

where $\xi_i \sim \mathcal{N}(0, \sigma_i^2\mathbf{I})$. If $\sigma_i = \epsilon_i^{-1}\Delta_f\sqrt{2\log(1.25/\delta)}$, $\sigma \in (0, 1]$, then (4) satisfies (ε, δ)-LDP,

$$\varepsilon = O\left(\sqrt{\frac{1}{\sum_i \sigma_i^2}}\right) \tag{5}$$

Proof Sketch. We know that the variance of noise summed over multiple Gaussian noises is equal to the sum of the variance of each Gaussian noise. Hence, we can substitute $\sum_i \sigma_i^2$ into (3), and solve the overall privacy budget. Then, we will get (5). □

Remark 1. *It is worth noting that (4) is different from the classic sequential composition [5] of DP mechanisms. The sequential composition is to give the privacy loss upper bound of multi-time differentially private releasing of one input/dataset. Instead, (4) means injecting noise multiple times into one releasing, and (5) gives the corresponding privacy budget of (4) but ignoring logarithmic factors.*

IV. SYSTEM MODEL

A. Model Description

In this subsection, we elaborate on the sequential noise-adding process. We consider a set of $\mathcal{K} \triangleq \{1, 2, .., K\}$ users, where each user $k \in \mathcal{K}$ has a privacy-sensitive vector $\boldsymbol{x}_i \in \mathbb{R}^d$. The users are lined up in a sequence $\theta_{\mathcal{K}} = (1, \cdots, K)$, and accumulate summation of the noisy vector one by one sequentially. Finally, the last user in the sequence sends the summation to the server. The server divides the summation by the total number of users K to obtain the estimated vector mean. To be specific, for $k = 1$, the first user applies the Gaussian mechanism and sends the noisy vector to the second user, i.e.,

$$\hat{S}_1 = \boldsymbol{x}_1 + \mathcal{N}(\sigma_1^2\mathbf{1}), \tag{6}$$

where $\mathcal{N}(\sigma^2\mathbf{1})$ refers to a random variable sampled from normal distribution with zero mean and σ^2 variance, and each dimension is independent identically distributed (i.i.d.), and $\mathbf{1} = (1, ..., 1)$. Then, the second user injects noise into the summation of the received data and its own data and sends the processed data to the third user. The rest of the users do the same operation. Formally, for any $k \in \mathcal{K}$ and $k > 1$, their releasing are given as follows:

$$\hat{S}_k = \hat{S}_{k-1} + \boldsymbol{x}_k + \mathcal{N}(\sigma_k^2\mathbf{1}), \tag{7}$$

The last user in the sequential sends

$$\hat{S}_K = \sum_{k\in\mathcal{K}} \boldsymbol{x}_k + \sum_{k\in\mathcal{K}} \mathcal{N}(\sigma_k^2\mathbf{1}), \tag{8}$$

where $\sigma_k^2 = 2\epsilon_k^{-2}\Delta_f^2\log(1.25/\delta)$, to the server. For simplicity, we assume that each dimension of the vector is bounded by Δ, thus the sensitivity is $\Delta_f = \sqrt{d}\Delta$. Further, we assume the users share the same failure probability δ. We give an interesting discussion of this assumption in Section VIII. Finally, the server obtains the vector mean estimation by $\boldsymbol{\mu} = \frac{1}{K}\hat{S}_K$.

B. Problem Formulation

In this subsection, we formulate users' cost minimization problem. Each user's cost are composed of two parts: accuracy cost and privacy cost. We model accuracy loss according to the upper bound of Mean Square Error (MSE), and model privacy cost in form of a DP budget. Note that when we model privacy cost, we do not take collusion attack into account first, and we tackle the collusion issues in Section VI.

1) Accuracy cost: When the noisy vectors are aggregated, the estimated vector mean is not accurate because of contained noises. We drive the upper bound of the MSE, i.e., L_2 error, between the real value and estimated value as the accuracy cost. MSE emphasizes the variance in the error, which is important for many applications [16], [17]. And some published works also used MSE to model accuracy cost [18], [19].

First of all, we denote the real vector mean as $\boldsymbol{\mu} = \frac{1}{K}\sum_{i\in\mathcal{K}}\boldsymbol{x}_i$. We calculate the expectation and variance of $\hat{\boldsymbol{\mu}}$, i.e., $\mathbb{E}(\hat{\boldsymbol{\mu}}) = \boldsymbol{\mu}$ and,

$$\begin{aligned}\mathbb{D}(\hat{\boldsymbol{\mu}}) &= \left[\mathbb{D}\left(\frac{1}{K}\sum_{i=1}^{K}x_i\right) + \mathbb{D}\left(\frac{1}{K}\sum_{i}^{K}\mathcal{N}(\sigma_i^2)\right)\right]_{1\times d} \\ &= \frac{1}{K^2}\sum_{i}^{n}\sigma_i^2\mathbf{1},\end{aligned} \tag{9}$$

then by Chebyshev's inequality, we get

$$\mathrm{MSE}(\boldsymbol{\mu},\hat{\boldsymbol{\mu}}) \leq \sum_{i}^{K}\sigma_i^2, \tag{10}$$

with high probability. Recall that we should define accuracy cost as the upper bound of the MSE. Hence, denote accuracy cost of user k as C_k^{acc}, we define

$$C_k^{acc} \triangleq \sum_{i}^{K}\sigma_i^2,\ \forall k\in\mathcal{K}. \tag{11}$$

2) Privacy cost: Without considering collusion, we model the privacy cost of user k as the privacy budget corresponding to the noise level injected into user k's data when it is sent out. We know that

$$\begin{aligned}\hat{S}_k &= \sum_{i<k}(\boldsymbol{x}_i + \mathcal{N}(\sigma_i^2)) + \boldsymbol{x}_k + \mathcal{N}(\sigma_k^2) \\ &= \sum_{i<k}\boldsymbol{x}_i + \left(\boldsymbol{x}_k + \sum_{i\leq k}\mathcal{N}(\sigma_i^2)\right),\ \forall i,k\in\mathcal{K},\end{aligned} \tag{12}$$

thus, based on Corollary 1, we define the privacy cost of user k as

$$C_k^{pri} \triangleq \sqrt{\frac{1}{\sum_{i\leq k}\sigma_i^2}},\ \forall i,k\in\mathcal{K}. \tag{13}$$

Finally, combining two costs, we formulate each user's cost minimization problem as follows.

$$\min_{\sigma_k^2} \quad C_k = C_k^{acc} + \alpha_k C_k^{pri}, \forall k\in\mathcal{K}, \tag{14}$$

where $\alpha_k \in [0,+\infty)$ and it is a balance parameter between privacy and accuracy.

The sum of the objective of (14) over k is convex and easy to solve the centralized optimum, but the users are selfish and their behavior is unclear. In the case of distributed optimization, each player is rational and selfish, thus the outcome is not necessarily consistent with the centralized optimum. Hence, we need to analyze the problem by game theory. Then, we formally formulate the sequentially noise-adding process as a multi-player sequential game. According to (13) and (11), we can see that each user's decision is influenced by other users in the sequence. Let $\sigma_{-k}^2 = (\sigma_1^2, ..., \sigma_{k-1}^2, \sigma_{k+1}^2, ..., \sigma_K^2)$ be the variance of the injected noise decisions by all other users except user k. Given other users' decisions σ_{-k}^2, user k would like to chose a proper decision σ_k^2 to minimize its cost function,i.e.,

$$\min_{\sigma_k^2\in[0,+\infty)} C_k(\sigma_k^2, \sigma_{-k}^2), \forall k\in\mathcal{K}. \tag{15}$$

Here $C_k(\sigma_k^2,\sigma_{-k}^2) \triangleq \beta_k C_k^{acc} + C_k^{pri}$. We then formulate the problem above as a strategic sequential game $\Gamma = \left(\mathcal{K}, \beta_{\mathcal{K}}, \{\sigma_k^2\}_{k\in\mathcal{K}}, \{C_k\}_{k\in\mathcal{K}}\right)$, where the set of users $\mathcal{K}$ is the set of players, $\theta_{\mathcal{K}}$ is the sequence of $\mathcal{K}$, $\sigma_k^2 \in [0,+\infty)$ is the strategy for player k, and C_k is the cost function to be minimized by player k. There are K stages in the game Γ. In each stage, player k chooses optimal σ_k^2 based on (15), and performs (7). In the sequel, we call the game Γ the multi-user privacy budget recycling game. We now introduce the important solution concept of Nash equilibrium (NE).

Definition 3. *A strategy profile $\boldsymbol{\sigma^{2*}} = (\sigma_1^{2*}, ..., \sigma_K^{2*})$ is a **Nash equilibrium** of the multi-user privacy budget recycling game of at the equilibrium $\boldsymbol{\sigma^{2*}}$, no user can further reduce its cost by unilaterally changing its strategy, i.e.,*

$$C_k(\sigma_k^{2*}, \sigma_{-k}^{2*}) \leq C_k(\sigma_k^2, \sigma_{-k}^{2*}),\ \forall\sigma_k^2\in[0,+\infty), k\in\mathcal{K}. \tag{16}$$

V. GAME-THEORETICAL ANALYSIS

In this section, we consider a game-theoretical analysis of the users' cost minimization problem for the sequentially noise-adding process. We first analyze the game in the complete information scenario. Then, we design a truthful mechanism where the optimal strategy of each player is to report their preference truthfully.

A. Complete Information Scenario

In this subsection, we find the NE in the complete information scenario. In other words, we assume that given a set of players, every player knows each other player's balance parameters and actions. Generally, the user at the top of the sequence has a certain dominance in sequential games. In the multi-stage discrete game, we could use back induction step by step to find the NE. However, our game is a continuous game. Hence, finding NE will be more challenging and tricky. Specifically, we try to derive the users' best responses to find inspiring insights. First of all, we derive the best response of user $k > 1$, given others' decisions, which is shown as fellows.

Lemma 1. *The best response strategy σ_k^2 of user $k \in \mathcal{K} \wedge k > 1$ in the multi-user privacy budget recycling game is*

$$\sigma_k^{2*} = \begin{cases} m\beta_k - \sum_{i<k} \sigma_i^{2*}, & \sum_{i<k} \sigma_i^{2*} < m\beta_k \\ 0, & otherwise, \end{cases} \tag{17}$$

where $m = (1/2)^{2/3}$ and $\beta_k = \alpha^{2/3}$.

Proof Sketch. It can be proved that the objective equation of (5) is convex with respect to $\sigma_k^2, \forall k \in \mathcal{K} \wedge k > 1$. Find the first order partial derivative of σ_k^2 for the objective equation of (5) and let it equal to 0. Then, we can solve the equation to get (17). □

Remark 2. *We call $\beta_k = \alpha_k^{2/3}$ as* ***preference*** *of user k and β_k uniquely identifies a user. Hence, we denote users' sequence based on each user's preference, i.e., $\boldsymbol{\beta}_{\mathcal{K}} = (\beta_1, ..., \beta_K)$.*

In the rest of the paper, we always denote $m = (1/2)^{2/3}$. Lemma 1 implies that each user's best response is based on its preference and added noises by the previous users in the sequence. If the variance of the existing noises is less than the threshold $m\beta_k$, the user k would like to inject noises up to $m\beta_k$.[1]

The best response can be understood as each user converts the utility-privacy trade-off to an appropriate threshold strategy. Furthermore, to sum up the variances over all k users, we have

$$\sum_{i \in \mathcal{K}} \sigma_i^{2*} = m \max \boldsymbol{\beta}_{\mathcal{K}}. \tag{18}$$

Hence, based on (18) and (11), we get a critical Corollary.

Corollary 2. *Given a set of users with different preferences $\{\beta_i\}_{i\in\mathcal{K}}$, the optimal privacy cost*

$$C_k^{acc*} \equiv 2^{-2/3} \max(\{\beta_i\}_{i\in\mathcal{K}}), \ \forall k \in \mathcal{K}. \tag{19}$$

In the complete information scenario, the first player knows that no matter how the other players are sorted, the optimal privacy cost only depends on the player with maximum preference in the sequence. Hence, the dominant strategy of the first player is to choose noise variance proportional to $\max \boldsymbol{\beta}_{\mathcal{K}}$ and subsequent players do not inject noises. Specifically, Theorem 1 gives the NE of the multi-user privacy budget recycling game.

Theorem 1. *Given any $\boldsymbol{\beta}_{\mathcal{K}}$, strategy profile*

$$\boldsymbol{\sigma^{2*}} = (m \max \boldsymbol{\beta}_{\mathcal{K}}, 0, ..., 0), \tag{20}$$

is the NE of the multi-user privacy budget recycling game.

Proof Sketch. Based on Lemma 1, we get the first player's marginal cost function given other players' best response strategies, i.e.,

$$C_1(\sigma_1^2, \sigma_{-1}^{2*}) = \begin{cases} \beta_1 \sqrt{\dfrac{1}{\sigma_1^2}} + C_1^{acc*}, & 0 \le \sigma_1^2 \le m \max \boldsymbol{\beta}_{\mathcal{K}} \\ \beta_1 \sqrt{\dfrac{1}{\sigma_1^2}} + K\sigma_1^2, & \sigma_1^2 > m \max \boldsymbol{\beta}_{\mathcal{K}}. \end{cases} \tag{21}$$

It is easy to find that $\beta_1 \sqrt{\frac{1}{\sigma_1^2}} + C_1^{acc*}$ is decreasing with respect to σ_1^2 (C_1^{acc*} is a constant). Further, $\beta_1 \sqrt{\frac{1}{\sigma_1^2}} + K\sigma_1^2$ is convex with respect to σ_1^2 in $[0, \infty)$ and the minimal value point $m\beta_1 < m \max \boldsymbol{\beta}_{\mathcal{K}}$. Hence, the (21) has a unique minimal value point $\sigma_1^{2*} = m \max \boldsymbol{\beta}_{\mathcal{K}}$. Lastly, substituting σ_1^{2*} into the subsequent players' best response, we get $\boldsymbol{\sigma^{2*}} = (m \max \boldsymbol{\beta}_{\mathcal{K}}, 0, ..., 0)$. □

Further, this NE is also a subgame perfect equilibrium [21], because the strategy chosen by each player at any point in the game is a best response to the strategies chosen by the other players in all possible subgames that could follow. So far, we have solved the multi-user privacy budget recycling game in the complete information scenario.

In this subsection, we analyze and solve the multi-user privacy budget recycling game in the complete information scenario. However, there are still two challenges. First, complete information is impractical. The preference of each user is private information. The user would not reveal the preference if it does not have enough incentives. Second, (20) indicates that except for the first user in the sequence, the other users do not add noise at all. This will lead to a huge risk of collusion attacks for non-first users. For any user $k > 1$, if the user $k-1$ and $k+1$ collude, the user k's data without any protection will be exposed. In the following two subsections, we will tackle these two challenges.

B. Mechanism Design

In this subsection, we aim at designing a mechanism that satisfies the following four desirable economic properties:

E1) **Truthfulness:** A user should truthfully reveal his preference.

E2) **Fairness:** All users' costs are equal under equilibrium.

E3) **Voluntary Participation:** Compared to the LDP, participating in the mechanism, a user can always achieve a smaller cost.

E4) **Efficiency**[2]**:** Following the definition of the price of anarchy (PoA) in game theory [23], the efficiency ratio PoA is upper-bounded.

In addition, we highlight 1) The comparison with LDP demonstrates voluntary participation. The property of (E3) reveals the priority of Chained-DP compared to LDP. 2) The theorem supporting (E4) also provides a guideline to split the sequence into sub-sequences to accelerate communication.

[1] We assume the users are able to estimate the variance of the noise contained in the data. A method to estimate the DP budget of the data publisher is proposed in [20], which supports this assumption. We discuss the assumption in detail in Sec. VIII.

[2] [22] proved that there does not exist any public goods mechanism that satisfies social welfare maximization and induces direct and truthful revelation at the same time. Our mechanism needs truthful revelation, so instead of pursuing optimal social welfare, we try to bound the PoA.

Next, we describe the Broadcast And Shuffle (BAS) mechanism. The proposed BAS mechanism is simple but effective. The mechanism includes three main stages:

- **Stage 1.** Each user $k \in \mathcal{K}$ broadcasts its preference biding b_k[3].
- **Stage 2**[4]**.** Arrange the users uniformly randomly in an ordered sequence, but if the user with the maximum preference among all users is in the first place, reshuffle the sequence until it is not.
- **Stage 3.** Perform the sequentially noise-adding process described in Sec. IV-A.

Mechanism 1 Broadcast And Shuffle (BAS)

Ensure: Noisy mean $\boldsymbol{\mu}$
Require: users' data $(\boldsymbol{x_1}, ..., \boldsymbol{x_K})$
// Stage 1
Each user $k \in \mathcal{K}$ broadcasts its preference biding b_k
$\boldsymbol{b} \leftarrow (b_1, ..., b_K)$
// Stage 2
$\boldsymbol{b} \leftarrow$`Shuffle`$(\boldsymbol{b})$
while $\max \boldsymbol{b} == \boldsymbol{b}[1]$ **do**
    $\boldsymbol{b} \leftarrow$`Shuffle` $(\boldsymbol{b})$
end while
// Stage 3
$\boldsymbol{\sigma^{2*}} \leftarrow (m \max \boldsymbol{b}, 0, ..., 0)$
for i: 1 to K **do**
    $S \leftarrow \boldsymbol{x}_{\boldsymbol{b}[i]} + \mathcal{N}(\boldsymbol{\sigma^{2*}} \boldsymbol{b}[i])$
end for
$\boldsymbol{\mu} \leftarrow \frac{1}{K} S$
return $\boldsymbol{\mu}$

A detailed description of the BAS mechanism is shown in Mechanism 1. Then, we show and prove the proposed BAS mechanism satisfies (E1)-(E4). First of all, the following Theorem implies (E1).

Theorem 2. *In the BAS mechanism, for a user $k \in \mathcal{K}$, the dominant strategy of the BAS mechanism is to report the preference β_k truthfully, i.e., $b_k = \beta_k, \forall k \in \mathcal{K}$.*

Proof. We prove this theorem by the counterfactual method, i.e., we show if a user $k \in \mathcal{K}$ broadcasts $b_k \neq \beta_k$, then the outcome cost $C_k^*(b_k, b_{-k}) \geq C_k^*(\beta_k, b_{-k})$, where C_k^* refers to equilibrium margin cost of user k and $b_{-k} = (b_1, ..., b_{k-1}, b_{k+1}, ..., b_K)$. There are two cases of user k's preference, i.e., maximum and non-maximum in the sequence.

Case 1 (maximum): In this case, according to (17), no matter where it is in the sequence, it always chooses to add noise up to the variance of noise up to $\beta_k = m \max \boldsymbol{\beta}_{\mathcal{K}}$ to minimizing its marginal cost. Hence, reporting any $b_k \in [0, +\infty)$ gains the same equilibrium cost, i.e., for any $b_k \in [0, +\infty)$, $C_k^*(b_k, b_{-k}) = C_k^*(\beta_k, b_{-k})$.

Case 2 (non-maximum): There are two sub-cases in this case, which will be discussed as fellows.

Case 2.1: First, it reports a preference bid $b_k > \beta_k$ or $b_k < \beta_k$, but the bid is still not the biggest bid among all players, i.e., $b_k < \max b_{-k}$. In this case, based on (20), we know that it will not cause any impact on its marginal cost, i.e., if $\beta_k \neq \max \boldsymbol{\beta}_{\mathcal{K}}$ and $b_k < \max b_{-k}$, then $C_k^*(b_k, b_{-k}) = C_k^*(\beta_k, b_{-k})$.

Case 2.2 Second, it reports a preference bid $b_k > b_{-k}$. In this case, the equilibrium marginal cost of user k is changed. Specifically,

$$C_k^*(\beta_k, b_{-k}) = \beta_k (m \max b_{-k})^{-1/2} + m \max b_{-k}, \quad (22)$$

and

$$C_k^*(b_k, b_{-k}) = \beta_k (m b_k)^{-1/2} + m b_k. \quad (23)$$

Consider $f(x) = \beta_k (mx)^{-1/2} + \frac{mx}{K^2}, x > 0$, the extreme point $x^* = \arg\min_x f(x) = \beta_k$. Hence, if $\beta_k \neq \max \boldsymbol{\beta}_{\mathcal{K}}$ and $b_k > b_{-k}$, then $C_k^*(b_k, b_{-k}) > C_k^*(\beta_k, b_{-k})$.

To summarize the all cases, we get $C_k^*(b_k, b_{-k}) \geq C_k^*(\beta_k, b_{-k})$. Hence, a user has no motivation to report $b_k \neq \beta_k$. □

Theorem 3. *Consider the multi-user privacy budget recycling game. For any two users $k, \hat{k} \in \mathcal{K}$,*

$$C_k(\sigma_k^{2*}, \sigma_{-k}^{2*}) = C_{\hat{k}}(\sigma_{\hat{k}}^{2*}, \sigma_{-\hat{k}}^{2*}). \quad (24)$$

Proof. Subscribing (20) into (14), we find that

$$C_k(\sigma_k^{2*}, \sigma_{-k}^{2*}) = 2^{1/3} \beta_k^{1/2} + 2^{-2/3} \beta_k, \forall k \in \mathcal{K}. \quad (25)$$

Hence, the theorem holds. □

Intuitively, Theorem 3 indicates the BAS mechanism achieves (E2). Further, we explore (E3). Generally, the property of voluntary participation is determined by whether a player's participation leads to positive utility in the utility maximization problem. Our problem is a cost minimization problem, so we can directly define the cost of not participating as infinite, but it is trivial. Hence, we investigate (E3) through comparison with LDP. We assume that if a user does not participate in the mechanism, it will add noise and send the data to the server individually, i.e., in LDP mode. And the noisy data will be used to compute the average value together with all other users no matter participate in the BAS mechanism or not. First, we write the optimization problem in LDP mode, which is a special case of (14), i.e,

$$\min_{\sigma_k^2} \; \beta_k (\sigma_k^2)^{-1/2} + \left(\sigma_k^2 + \sum_{i \in \mathcal{K} \setminus \{k\}} \sigma_i^{2*} \right), \; \forall k \in \mathcal{K}. \quad (26)$$

We denote the objective of (26) as C_k^L and its corresponding optimal cost as C_k^{L*}. Then, we have the following Theorem. Note that for the user k who chose to leave the sequence, $\sum_{i \in \mathcal{K} \setminus \{k\}} \sigma_i^{2*}$ can be regarded as a constant.

Theorem 4. *Consider the outcome of participating in the BAS mechanism. For any $k \in \mathcal{K}$,*

$$C_k^{L*} > C_k^*. \quad (27)$$

Proof Sketch. The objective function of (26) is convex. we solve the unique analytical solution i.e., $\sigma_k^{2L*} = m\beta_k$. Then, we do $C_k^{L*} - C_k^*$ and substitute σ_k^{2L*} into it. Then, we get

[3]We assume the preference is not very sensitive to the user and the users can also inject small noise before reporting.
[4]We assume that the probability of any two users having equal preference trends to 0. It makes sense that the preference is in continuous space $[0, +\infty)$.

two terms. For the first term, we know that $\max \boldsymbol{\beta}_{\mathcal{K}} \geq \beta_k$, $\sqrt{\frac{1}{m\beta_k}} - \sqrt{\frac{1}{\max \boldsymbol{\beta}_{\mathcal{K}}}} \geq 0$. For the second term, it is easy to find that $\beta_k + \max(\boldsymbol{\beta}_{\mathcal{K}} \backslash \{\beta_k\}) > \max \boldsymbol{\beta}_{\mathcal{K}}$. Hence, $C_k^{L*} - C_k^* > 0$, i.e., $C_k^{L*} > C_k^*$. □

Theorem 4 indicates leaving the sequence results in a bigger cost for any user $k \in \mathcal{K}$. Hence, the mechanism satisfies (E3).

Lastly, we verify (E4). First of all, we define the PoA of our problem in terms of the total cost (social welfare) as follows:

$$PoA = \frac{\sum_{k=1}^{K} C_k(\sigma_k^{2*}, \sigma_{-k}^{2*})}{\min_{\boldsymbol{\sigma^2}} \sum_{i=1}^{K} C_k(\sigma_k^2)}, \tag{28}$$

where the numerator is the summation of all users' optimal equilibrium cost and the denominator is the centralized optimum of (14).

Theorem 5. *For the BAS mechanism, given a $\boldsymbol{\beta}_{\mathcal{K}}$, the PoA in terms of total cost satisfies*

$$1 < PoA < \frac{\max \boldsymbol{\beta}_{\mathcal{K}}}{K \min \boldsymbol{\beta}_{\mathcal{K}}}. \tag{29}$$

Proof Sketch. It is oblivious that (14) is convex (non-negative weighted sums of basic convex functions). Hence, the centralized optimum of (14) is unique. Thus, we solve (14), i.e., $\boldsymbol{\sigma^{2\star}} = \left(m \left(\frac{\sum_{k=1}^{\mathcal{K}} \beta_k}{K} \right)^{2/3}, 0, ..., 0 \right)$. Then, we know that $\sigma_1^{2*} > \sigma_1^{2\star}$. We denote $C_T = \sum_{k=1}^{K} C_k$. It is easy to find that $\frac{C_T(\sigma_1^2, 0, ..., 0)}{\sigma_1^2}$ is decreasing with respect to σ_1^2 in $[0, +\infty)$. Hence, we have $\frac{C_1(\boldsymbol{\sigma^{2*}})}{\sigma_1^{2*}} < \frac{C_1(\boldsymbol{\sigma^{2\star}})}{\sigma_1^{2\star}}$. Then, using $\sum_{k=1}^{K} \beta_k > K \min \beta_k$ to deflate, we get $PoA < \frac{\max \boldsymbol{\beta}_{\mathcal{K}}}{\min \boldsymbol{\beta}_{\mathcal{K}}}$. □

Intuitively, Theorem 5 indicates that the smaller the gap between the maximal and minimal preference β_k in a sequence, the more efficient the BAS mechanism is. In other words, we can make a conclusion that users with closer preferences are willing to be in a sequence.

Moreover, Chained-DP is completed sequentially, which leads to a large communication delay. Fortunately, Chained-DP allows splitting the sequence into sub-sequences to alleviate this problem. Specifically, we can split a sequence into multiple sub-sequences. In each sub-sequence, we run the SAP protocol and finally we average all the mean vectors generated by each sub-sequence. Theorem 5 can also be used as a guideline for the splitting. Theorem 5 suggests it is more efficient to arrange users with similar preferences in the same sub-sequence.

Corollary 3. *For arbitrary split policy, the outcome estimation error of Chained-DP is always less than LDP.*

Corollary 4. *Given a $\boldsymbol{\beta}_{\mathcal{K}}$, we split the sequence into Q equal-length sub-sequences $\{\boldsymbol{\beta}_{\mathcal{K}}^i\}_{i=1}^{Q}$. The split policy achieves optimal error upper bound if and only if, for any two users with $\beta_j \in \boldsymbol{\beta}_{\mathcal{K}}^i$ and $\beta_k \in \boldsymbol{\beta}_{\mathcal{K}}^{-i}$, if $\beta_j < \beta_k$, then $\max \boldsymbol{\beta}_{\mathcal{K}}^i < \min \boldsymbol{\beta}_{\mathcal{K}}^{-i}$.*

The proof of Corollary 3 and 4 are straightforward and we omit them here. The policy in Corollary 4 can be easily implemented, i.e., sort the users according to their preference and equally split. We verify these two corollaries in Section VII.

To summarize, we design a simple but effective BAS mechanism. Then, we show and prove four economic properties of the BAS mechanism. However, the equilibrium of the mechanism is similar to the multi-user privacy budget recycling game. Except for the first user in the sequence, the others do not add noise, and the risk of collusion attacks is still reserved. In the following subsection, we will introduce a practical protocol to alleviate the risk.

VI. ANTI-COLLUSION DESIGN

In this section, we specify the definition of collusion privacy attacks in the sequentially noise-adding process first. Then, we introduce a practical anti-collusion attacks protocol, i.e., Slice And Recombine (SAR) protocol. Further, we analyze the performance of the protocol. We highlight that the privacy guarantee provided by the SAR protocol perfectly fits with the original definition of (ϵ, δ)-differential privacy and parallel composition. The SAR protocol can be generalized to any sequentially noise-adding process form where the first user adds all noise or the vast majority of noise and the others do not add noise or very small noise, even if the behavior analysis method is different from the modeling in Sec. IV.

Protocol 2 Slice And Recombine (SAR)

Ensure: Noisy mean μ
Require: users' data $(\boldsymbol{x_1}, ..., \boldsymbol{x_K})$, Number of slices N

```
// Slice
l = ⌈d/N⌉
for i: 1 to N do
    inx ← 1
    for k: 1 to K do
        if inx + l − 1 ≤ d then
            s_k^i ← x_k[inx, inx + l − 1]
        else
            s_k^i ← x_k[inx, inx + d mod l − 1]
        end if
    end for
    inx ← inx + 1
end for
μ^i ←Mechanism 1((s_1^i,...,s_K^i)){Parallel running for μ^i, ∀i ∈ [1, N]}
// Recombine
μ ←concat(μ^1, ..., μ^N)
return μ
```

A. Definition of Collusion Privacy Attacks

Before introducing the protocol, let us revisit the risk of collusion privacy attacks. Recall that the BAS mechanism faces severe collusion privacy violation risk. The first user in a sequence has the priority of adding the maximal noise, and the other users add their transparent raw data directly. Based on the user's behavior, the risk is two-fold.

First, for a user, if the closest two other users (previous and next) collude, there is no privacy protection anymore. To be specifically, according to (7) and (20), we know the user $k > 1$ sends $\hat{S}_k = \hat{S}_{k-1} + \boldsymbol{x}_k$ to user $k+1$. user $k-1$ and $k+1$ colludes means that $\boldsymbol{x}_k$ can be inferred by $\boldsymbol{x_k} = \hat{S}_k -$

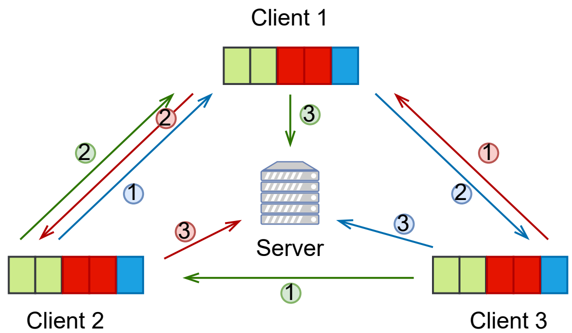


Fig. 2: Schematic Diagram of SAR protocol. Three colors represent three sliced sub-vectors (scalar) and the corresponding colored arrows represent uniformly randomly generated data flow, respectively.

$\hat{S}_{k-1}$ immediately. Therefore, we give the formal definition of collusion privacy attack.

Definition 4. *(Weak Attacker) Consider the BAS mechanism. For a victim $k \in \mathcal{K}$ and a set of attackers $\mathcal{A} \in \mathcal{K}$. Given a sequence $\boldsymbol{\beta}_{\mathcal{K}}$, we say the attackers are possible to launch a collusion privacy attack if and only if $\arg_{\boldsymbol{\beta}_{\mathcal{K}}} i+1 = \arg_{\boldsymbol{\beta}_{\mathcal{K}}} k = \arg_{\boldsymbol{\beta}_{\mathcal{K}}} j-1$, or $\arg_{\boldsymbol{\beta}_{\mathcal{K}}} j+1 = \arg_{\boldsymbol{\beta}_{\mathcal{K}}} k = \arg_{\boldsymbol{\beta}_{\mathcal{K}}} i-1$, where i, j can be arbitrary elements of $\mathcal{A}$.*

Second, we consider a stronger attacker model. If we allow data poisoning attacks, i.e., an attacker is able to replace others' true data with fake data. This means it is possible that even if attackers are not in the closest positions in the sequence, they can still infer other users' raw data, as long as the victim is located between two attackers, possibly with others in between.

Definition 5. *(Strong Attacker) Consider the BAS mechanism. For a victim $k \in \mathcal{K}$ and two attacker $i, j \in \mathcal{K}$. Given a sequence $\boldsymbol{\beta}_{\mathcal{K}}$, we say the attackers are possible to launch a collusion privacy attack if and only if $\arg_{\boldsymbol{\beta}_{\mathcal{K}}} i < \arg_{\boldsymbol{\beta}_{\mathcal{K}}} k < \arg_{\boldsymbol{\beta}_{\mathcal{K}}} j$, or $\arg_{\boldsymbol{\beta}_{\mathcal{K}}} j < \arg_{\boldsymbol{\beta}_{\mathcal{K}}} k < \arg_{\boldsymbol{\beta}_{\mathcal{K}}} i$, where i, j can be arbitrary elements of $\mathcal{A}$.*

B. Protocol Design and Analysis

In this subsection, we first introduce the SAR protocol. Then, we analyze the protocol from the perspective of differential privacy.

Protocol 2 illustrates the procedure of the SAR protocol. The intuition of the SAR protocol is that place different parts of the vector of a user randomly at different positions of sequences to reduce the probability of overall exposure to the attacker. To explain the SAR protocol intuitively, we give a schematic diagram of three users in Fig. 2. It can be seen that different parts of vectors (i.e., sub-vectors) are in different ordered sequences. Then, we describe the Protocol 2 in detail:

- First of all, the users agree on an appropriate slicing number. We denote the number as $N \in [1, d]$. N is the input of Protocol 2.
- Then, each user $k \in \mathcal{K}$ slices its data vector into $l = \lceil \frac{d}{N} \rceil$ sub-vectors $\boldsymbol{s}_k^i$, where $i \in [1, l]$. This procedure is described in line 3-11 in Protocol 2. Note that the data vector's dimension may not be divided without the remainder. The SAR protocol arranges the remaining part as a shorter vector, shown in lines 8-9.
- Next, for all $i \in [1, l]$, run line 13 to generate sub-means. Note that this step can be executed in parallel. Finally, recombine the sub-means to obtain the entire estimated mean, which is depicted in line 16.

Then, we analyze the SAR protocol from the perspective of differential privacy. We consider a model assuming that for a user $k \in \mathcal{K}$, the average probability of another user in the sequence being a collusion attacker is $p_k \in [0, 1)$. *p_k can be regarded as the quantitative indicators of* ***soft trust boundary***. Note that if a group of users is collusion attackers for user k, any two of them satisfying the Definition 4 or 5 can be regarded as a successful attack. Before participating in the Mechanism 1, the user can obtain p_k by observing. Based on this assumption and Definition 4 and 5, each user is able to calculate the probability of being colluded by other users. Further, recall the definition (Definition 1) and mathematical sense of approximate DP, a mechanism satisfies (ϵ, δ)-DP if and only if:

- With probability $1 - \delta$, the mechanism is ϵ-differentially private.
- With probability δ, the mechanism can not hold the guarantee.

For a sub-vector $\boldsymbol{s_k^i}$, we can derive a standard (ϵ, δ)-DP announcement. What is more, all sub-vectors are **disjoint subsets** of the entire vector. Hence, according to the parallel composition of DP (i.e., Theorem 6), we can derive the privacy guarantee of an entire release of a user.

Theorem 6. *(Parallel Composition [5]) If there are n functions $M_1, M_2, ..., M_n$ computed on* ***disjoint subsets*** *of the private database whose privacy guarantees are (ϵ_1, δ_1),...,(ϵ_n, δ_n) differential privacy, respectively, then any function g of them: $g(M_1, \ldots, M_n)$ is $(\max_i \epsilon_i, \max_i \delta_i)$ -differentially private.*

According to the above analysis, we can give the following two theorems[5].

Theorem 7. *Consider the Definition 4, the SAR mechanism satisfies $(\hat{\varepsilon}, \max(\hat{\delta}_1, \delta))$-differential privacy, where*

$$\hat{\varepsilon} = \frac{2\Delta \log(1.30/\delta)}{m \max \boldsymbol{\beta}_{\mathcal{K}}}, \tag{30}$$

and

$$\hat{\delta}_1 = \frac{(K-2)p_k^2}{K}. \tag{31}$$

Proof. First of all, by substituting the (20) into (3), we get (30) immediately. Then, we consider the failure probability. In addition, consider a sub-vector $\boldsymbol{s}_k^i$, if it is arranged in the first and last position of a sequence, the probability of being colluded is 0. And if it is arranged in other positions, the probability is p_k^2. Further, the sequence is generated in a uniformly random manner. Hence, we have $\hat{\delta}_1 = \frac{(K-2)p_k^2}{K}$.

[5] In practice, the user with maximal preference can always not be in the first place in the sequence. But for simplicity, we directly treat it as uniform distribution in the analysis. The related properties will not change due to simplification except for some non-critical constants.

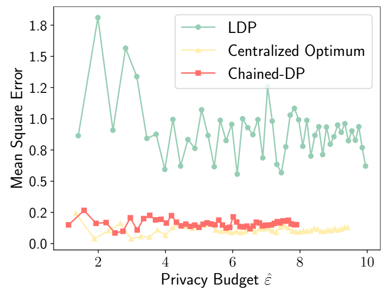


Fig. 3: Mean square error with different privacy budget by varying d.

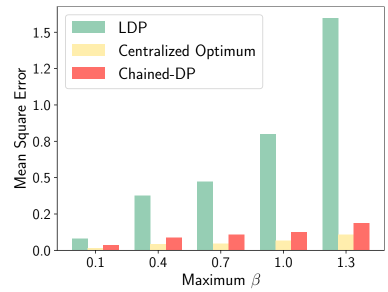


Fig. 4: Mean square error with different β ranges.

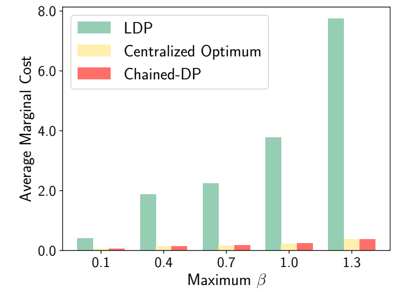


Fig. 5: Average marginal cost with different β ranges.

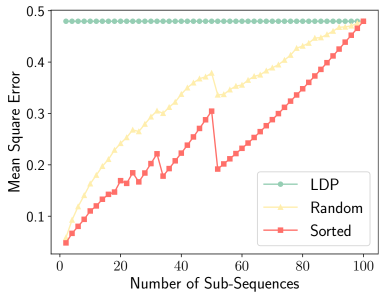


Fig. 6: Mean square error with different number of sub-sequences.

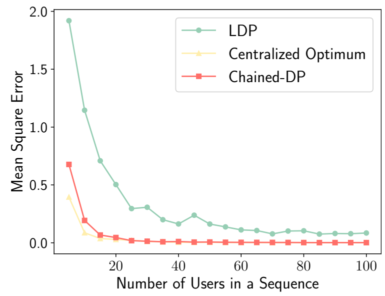


Fig. 7: Mean square error with different number of users.

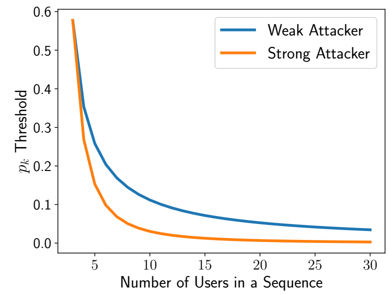


Fig. 8: p_k threshold with different number of users.

Then, based on Theorem 6, the SAR mechanism satisfies $(\hat{\varepsilon}, \max(\hat{\delta}_1, \delta))$-differential privacy. $\square$

Theorem 8. *Consider the Definition 5, the SAR mechanism satisfies* $(\hat{\varepsilon}, \max(\hat{\delta}_2, \delta))$*-differential privacy, where*

$$\hat{\delta_2} = 1 + (1 - p_k)^{K-1} - \frac{2(1 - (1 - p_k)^{K-1})}{K(1 - p_k)}. \tag{32}$$

Proof Sketch. The logic of this proof is similar to the proof of the Theorem 7. The privacy budget is the same as the Theorem 7. Further, We denote $\overline{p}_k = 1 - p_k$, then we have $\hat{\delta_2} = \frac{1}{K} \sum_{i=0}^{K-1} \left(1 - \overline{p}_k^i\right)\left(1 - \overline{p}_k^{K-1-i}\right)$. Then, we get (5) by consolidation. $\square$

Based on these two theorems, i.e., Theorem 7 and 8, we first analyze the privacy budget $\hat{\varepsilon}$. Note that we assume that each dimension of the vectors is bounded by Δ in Sec. IV. Thus, the optimal slicing strategy is to slice the vector into scalars. In reality, federal analytics usually bound the l_2-norm of gradient vectors by clipping [24]. In this case, each sub-vector needs to be re-clipped. The re-clipping may result in accuracy loss. To be specific, the sliced vector's new sensitivity $\Delta_s = O\left(\Delta_f \sqrt{K/N}\right)$. Intuitively, based on (30), the finer the vector slice, the smaller the privacy budget, i.e. the better the privacy protection, but with less accuracy.

Next, we analyze the failure probability $\hat{\delta_1}$ and $\hat{\delta_2}$. Different from the "gentle" Gaussian mechanism that can set δ a very small value, we have to carefully consider the failure probability. Empirically, the failure probability should be less than $1/K^2$. Thus, following the Definition 4, given a total number of users K,

$$p_k \leq \sqrt{\frac{1}{K(K-2)}}. \tag{33}$$

For the Definition 5, it is difficult to solve an analytical solution of p_k, but given a K, the numerical solution can be solved. Both Definition 4 and 5 require a small p_k. There are two possible methods to alleviate the problem. The first is that add a pre-mechanism where a user can choose whether participate in SAR protocol based on the p_k threshold. The second is that for each release, the user uploads the past-round sub-vectors or zero-vectors instead of the new sub-vectors with a probability p_p. Hence, it can further reduce failure probability by $\hat{\delta} \cdot p_p$. The nature of federated analytics supports this operation well [25]. Moreover, (33) also indicates that there cannot be too many users in a sequence, and the queue needs to be split if necessary.

To summarize this section, we first introduce the concept of the collusion attack. Then, we describe the SAR protocol. Next, we analyze the protocol from the perspective of differential privacy. Last but not least, based on the privacy budget and failure probability, we figure out the relationship between the number of slices and sensitivity. And, we give some advice to implement the SAR protocol in practice.

VII. Numerical Results

In this section, we evaluate Chained-DP by numerical studies. We build a simulation of Chained-DP and explore MSE, average marginal cost, and the p_k threshold by varying different parameter settings to show the effectiveness and superiority compared to LDP. We also chose the centralized optimum as another baseline. We show that Chained-DP is very closed to the centralized optimum.

We first consider the MSE with different privacy budgets by varying the dimension d. Expressly, we set the number of users in the sequence $K = 10$. Each user generates a vector wherein each dimension is in the range $[-1, 1]$ and a preference β_k

in the range $[0, 1.3]$ in a uniformly random manner. Then, we vary d from 2 to 100 to show the corresponding MSE and privacy budget $\hat{\varepsilon}$ of LDP, centralized optimum, and the proposed Chained-DP. The LDP fellows Eq. (26). The experiment result is shown in Fig. 3. It turns out that the MSE of Chained-DP is much lower than LDP and very close to the centralized optimum. In addition, with d increasing, the LDP and centralized optimum achieves a higher privacy budget than Chained-DP, which means our Chained-DP saves more budget.

Then, we investigate the influence of the preference β_k. We fix the other parameters in Fig. 3 and set $d = 50$. The preference is still generated uniformly randomly in a range, but we vary the maximum of the range from 0.1 to 1.3. Fig. 4 shows the MSE with different β ranges. It can be seen that the LDP introduces much higher errors than the other two. Further, Fig. 5 shows the average marginal cost of each user with different β ranges, and the results are similar. This phenomenon is supported by Theorem 5.

Furthermore, we know that to reduce the communication delay and lower the risk of being colluded, we need to split a long sequence into multiple sub-sequences. Hence, we show the MSE with different number of sub-sequences in Fig. 6. Specifically, we still fix $d = 50$, $\Delta = 1$ and set preference range $[-1, 1]$. Then, we split the generated sequence into from 2 to 100 sub-sequences. For each sub-sequence, we run a simulation of Chained-DP, and finally average the estimated vectors of all sub-sequence, and calculate the MSE. There are two kinds of split methods. The first is to split the sequence randomly. The second is to sort the users according to their preferences and then split them. It can be seen that the sorted and split method always outputs a lower MSE than the random split directly.

Lastly, we explore the influence of the number of users in a sequence. We set $\Delta = 1$, $D = 50$ and maximum β is 1.3. Then, we vary K from 3 to 100, which is shown in Fig. 7. We can see that when the number of users is small, Chained-DP is much better than LDP. But when the number raises, the two methods tend to be consistent. Additionally, Fig. 8 shows that when the number of users increases, the p_k threshold goes down. Viewing Fig. 6, Fig. 7 and Fig. 8 together, it is highly suggested that split the long sequences into small sub-sequences with 3-20 users when deployment Chained-DP in a real system.

VIII. DISCUSSION AND LIMITATIONS

First, we discuss the preference β_k. Recall that the preference β_k uniquely identifies a user in our model. The β_k is the coefficient of C_k^{pri} in (14). We can understand this coefficient as the balance parameter between privacy and accuracy, i.e., preference. But we can also understand this coefficient from another perspective. If all users value privacy and accuracy equally, then the β_k should be part of C_k^{pri}. Revisiting (3) and (5), we find that β_k can be defined as $\beta_k \triangleq \delta\Delta_f$. In this definition, we do not need the assumption that all users unify δ in the first place in Section IV.

Further, we assume the users are able to estimate the variance of the contained in data. This assumption has limitations. Even though [20] proposed a likelihood estimation method, it needs a successful data poisoning attack first. Take a closer look at this problem. We need to estimate the variance by $\sum_{i<k} \sigma_i^2 = \mathrm{mean}(\hat{S} - S)$. In reality, if a user can not use data poisoning, it only holds the upper bound of S. Hence, the estimation is biased. However, the structure of the best response would not be changed and the equilibrium may become a Bayesian Nash equilibrium. We intend to explore this problem carefully in future work.

IX. CONCLUSION

In this paper, we explore for the first time the possibility of recycling the privacy budget in a sequence specifically. Considering the users' selfish behavior, we model a sequential game to analyze the sequentially noise-adding process. We propose a simple yet effective BAS mechanism, which satisfies four desired properties including incentive compatibility, fairness, voluntary participation, and efficiency. In addition, we propose a practical protocol, i.e., SAR protocol, and analyze the protocol from the perspective of differential privacy. This is a pilot attempt to explore the sequentially noise-adding process and there are many interesting directions for extending this work. One possibility is to consider a multi-sequence model where the communication conditions are heterogeneous or competitive. Further, it is meaningful to study how to relax the assumption that the user can infer the variance accurately.

REFERENCES

[1] D. Wang, S. Shi, Y. Zhu, and Z. Han, "Federated analytics: Opportunities and challenges," *IEEE Network*, vol. 36, no. 1, pp. 151–158, 2021.

[2] T. Wang and J. Blocki, "Locally differentially private protocols for frequency estimation," in *Proc. USENIX Security*, 2017.

[3] Y. Wang and T. Li, "Privacy-preserving data analysis in cloud computing: A survey," *Journal of Cloud Computing*, vol. 8, no. 1, pp. 1–19, 2019.

[4] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Proc. NeurIPS*, vol. 32, 2019.

[5] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[6] M. Yang, L. Lyu, J. Zhao, T. Zhu, and K.-Y. Lam, "Local differential privacy and its applications: A comprehensive survey," *arXiv preprint arXiv:2008.03686*, 2020.

[7] P. Kairouz, Z. Liu, and T. Steinke, "The distributed discrete gaussian mechanism for federated learning with secure aggregation," in *Proc. ICML*, 2021, pp. 5201–5212.

[8] M. Zhou, T. Wang, T. H. Chan, G. Fanti, and E. Shi, "Locally differentially private sparse vector aggregation," in *Proc. IEEE S&P*, 2022, pp. 422–439.

[9] R. Xu, N. Baracaldo, Y. Zhou, A. Anwar, and H. Ludwig, "Hybridalpha: An efficient approach for privacy-preserving federated learning," in *Proc. AISEC*, 2019, pp. 13–23.

[10] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning," in *Proc. USENIX ATC*, 2020.

[11] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proc. ACM CCS*, 2016, pp. 308–318.

[12] Z. Liu, W. Wang, W. Chen, T. Liu, X. Du, and H. Guan, "Privacy budget management for multi-user differential privacy in cloud computing," *Information Sciences*, vol. 514, pp. 206–222, 2020.

[13] P. Tholoniat, K. Kostopoulou, M. Chowdhury, A. Cidon, R. Geambasu, M. Lécuyer, and J. Yang, "Packing privacy budget efficiently," *arXiv preprint arXiv:2212.13228*, 2022.

[14] T. Luo, M. Pan, P. Tholoniat, A. Cidon, R. Geambasu, and M. Lécuyer, "Privacy budget scheduling." in *Proc. OSDI*, 2021, pp. 55–74.

[15] W. Jiang, Z. Ma, S. Li, H. Xiao, and J. Yang, "Privacy budget management and noise reusing in multichain environment," *International Journal of Intelligent Systems*, vol. 37, no. 12, pp. 10 462–10 475, 2022.

[16] J.-J. Cao, S. Kwong, R. Wang, and K. Li, "A weighted voting method using minimum square error based on extreme learning machine," in *Proc. IEEE ICMLC*, vol. 1, 2012, pp. 411–414.

[17] R. Srikant and L. Ying, "Finite-time error bounds for linear stochastic approximation andtd learning," in *Proc. COLT*, 2019, pp. 2803–2830.

[18] A. Ghosh, T. Roughgarden, and M. Sundararajan, "Universally utility-maximizing privacy mechanisms," in *Proc. ACM STOC*, 2009, pp. 351–360.

[19] G. Liao, X. Chen, and J. Huang, "Prospect theoretic analysis of privacy-preserving mechanism," *IEEE/ACM Transactions on Networking*, vol. 28, no. 1, pp. 71–83, 2019.

[20] C. Dwork, N. Kohli, and D. Mulligan, "Differential privacy in practice: Expose your epsilons!" *Journal of Privacy and Confidentiality*, vol. 9, no. 2, 2019.

[21] R. B. Myerson, "Refinements of the nash equilibrium concept," *International Journal of Game Theory*, vol. 6, no. 2, pp. 73–80, 1977.

[22] M. Walker, "On the nonexistence of a dominant strategy mechanism for making optimal public decisions," *Econometrica: Journal of the Econometric Society*, pp. 1521–1540, 1980.

[23] T. Roughgarden, *Selfish routing and the price of anarchy*. MIT press, 2005.

[24] L. Qu, Y. Zhou, P. P. Liang, Y. Xia, F. Wang, E. Adeli, L. Fei-Fei, and D. Rubin, "Rethinking architecture design for tackling data heterogeneity in federated learning," in *Proc. IEEE/CVF CVPR*, 2022, pp. 10 061–10 071.

[25] W. Wu, L. He, W. Lin, R. Mao, C. Maple, and S. Jarvis, "Safa: A semi-asynchronous protocol for fast federated learning with low overhead," *IEEE Transactions on Computers*, vol. 70, no. 5, pp. 655–668, 2020.