Private Multi-Group Aggregation

Carolina Naim*, Rafael G. L. D'Oliveira†, and Salim El Rouayheb*
*ECE, Rutgers University, {carolina.naim, salim.elrouayheb}@rutgers.edu
†RLE, Massachusetts Institute of Technology, rafaeld@mit.edu

2021 IEEE International Symposium on Information Theory (ISIT) | 978-1-5386-8209-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISIT45174.2021.9518142

***Abstract*—We study the differentially private multi-group aggregation (PMGA) problem. This setting involves a single server and n users. Each user belongs to one of k distinct groups and holds a discrete value representing his data. The goal is to design schemes that allow the server to find the aggregate (sum) of the values in each group (with high accuracy), under communication and local differential privacy constraints. The privacy constraint guarantees that the user's group remains private. This is motivated by applications where a user's group can reveal sensitive information, such as his religious and political beliefs, health condition, or race.**

We propose a novel scheme, dubbed Query and Aggregate (Q&A) for PMGA. The novelty of Q&A is that it is an interactive aggregation scheme. In Q&A, each user is assigned a random query matrix, to which he sends the server an answer based on his group and value. We characterize the Q&A scheme's performance in terms of accuracy (MSE), privacy, and communication. Private aggregation schemes for related settings in the literature are predominantly non-interactive and based on randomized response. We compare Q&A to the Randomized Group (RG) scheme, which adapts existing schemes to the PMGA setting. We observe that typically Q&A outperforms RG, in terms of utility vs. privacy, in the high privacy regime. Moreover, an attractive property of Q&A is that its communication cost per user does not depend on the number of groups.

I. Introduction

We consider the problem of distributed aggregation in which a centralized server wishes to compute the aggregate (sum) of the data (values) held by several participants (users). The users can communicate with the server in both directions. Privacy is a significant concern since participants have to share their data, which can be personal and sensitive. This has motivated works on private and secure distributed aggregation in many applications such as medical studies [1] or more recently federated learning [2]–[5].

In this work, we focus on the setting depicted in Figure 1, in which users belong to different groups. The server wants to find the aggregate for each group separately, instead of the aggregate over the whole population. The users' groups can be based, for example, on their political views, immigration status, health condition, location, race, to name a few. Evidently, this raises additional privacy concerns since participating users may be rightfully wary of revealing their group.

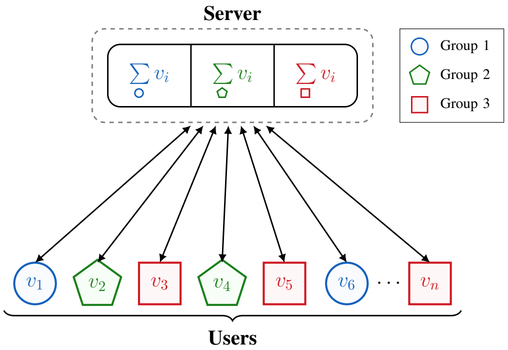


Fig. 1: An instance of the Private Multi-Group Aggregation problem with n users. Each user i, for $i \in [n]$, has a value, v_i, and belongs to one of the $k = 3$ distinct groups. The server's goal is to estimate the sum of the values in each of the groups.

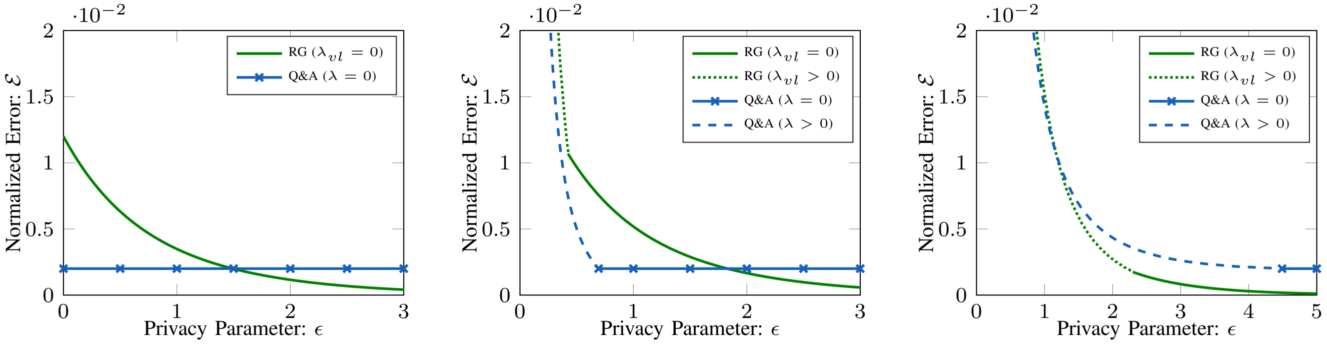
As an example of applications, consider medical research on the efficacy of a new vaccine on patients with or without a chronic illness, say diabetes. A person may want to contribute his experiment results, but does not want to reveal his medical condition (group). Another application is population polling. For instance, a political organization seeks to poll voters per group, such as race, gender, or income bracket. Participants want to partake in these polls without disclosing their groups.

We present the problem of Private Multi-Group Aggregation (PMGA), where local differential privacy guarantees are given over a user's group. We are interested in regimes with a large number of groups since this allows the server more refined statistics about the population. Our main objective is to design schemes with low communication cost, as users can have limited bandwidth. In particular, we focus on schemes that offer communication costs that are constant or at most logarithmic in the number of groups. Moreover, we study the trade-offs they offer between privacy, measured using local differential privacy, and accuracy, i.e., the aggregate estimation accuracy for each group.

A. Related Work

The classical setup for secure and private aggregation in the literature does not distinguish among groups, and the privacy guarantees are on the user's data (values). Differentially private schemes and bounds for private aggregation were studied in [6]–[10]. In [2] secure aggregation based on information-theoretic (secret sharing) and cryptographic techniques was developed for applications in federated learning (FL) [11]. Secure aggregation algorithms for FL with improved communication and computation overhead were proposed in [12], [13], and with robustness against adversarial users in [14]. The

The work of the first and last authors was supported by NSF Grant CCF 1817635. The work of the second author was supported by the FinTech@CSAIL Research Initiative.



(a) $p_1(1) = 0.5$ and $p_2(1) = 0.5$.

(b) $p_1(1) = 0.6$ and $p_2(1) = 0.3$.

(c) $p_1(1) = 0.9$ and $p_2(1) = 0.01$.

Fig. 2: Comparison of the Q&A and RG schemes for $k = 2$ groups, binary alphabet, *i.e.,* $v \in \{-1, 1\}$, and fixed total communication cost equal to 500 bits, *i.e.,* 500 bits communicated by all the users to the server. The Q&A scheme requires less communication cost per user compared to the RG scheme; therefore, for fixed total communication cost, the Q&A scheme can have more users. The subfigures (a), (b) and (c) illustrate accuracy vs. privacy of the Q&A and RG schemes for different user's value distributions, $p_1(1)$ and $p_2(1)$. The dashed (or dotted) curves represent the performance of the schemes with the additional layer of privacy that hides the user's values, *i.e.,* $\lambda > 0$ for the Q&A scheme, and $\lambda_{vl} > 0$ for the RG scheme.

schemes in [2], [12]–[14] have a communication cost per user that grows with the number of users.

A related problem is federated submodel learning [15]–[17]. In this setting, one or multiple servers hold various submodels (vectors) and each user wants to train (update) a private subset of these submodels. The notion of submodels here is similar to the notion of groups in our problem; however, a user's update depends on the submodels at the server in addition to his data. The proposed solutions in [16], [17] use information-theoretic private information retrieval (PIR) to privately download and update the submodels. Thus, they require multiple servers, and the communication cost per user is linear in the number of submodels (groups). Moreover, in [15] differentially private techniques were used to allow a user to download the required submodels, and update it using secure aggregation. The resulting scheme has a communication cost per user that grows with the total number of users.

B. Contributions

We propose a novel scheme for PMGA that we call the Query and Aggregate (Q&A) scheme that provides local differential privacy guarantees on the users' groups. The Q&A scheme is interactive in that the user is assigned a query matrix and sends the server an answer based on his group and value. This allows to shift the bulk of the total communication cost to the query stage (server-to-user) which could be done offline since it does not depend on a user's group and value. As a result, the online communication cost, user-to-server, does not depend on the number of groups and users. We characterize, in Theorem 1, the performance of the Q&A scheme in terms of privacy, communication cost, and accuracy.

We compare Q&A to a non-interactive scheme which we call the Randomized Group (RG) scheme. RG is an adaptation of standard randomized response [18] schemes from the literature and consists of each user reporting a noisy version of his group and value to the server. For a fixed total communication cost, we observe that in general Q&A offers better accuracy in high privacy regimes (small ϵ), as illustrated in Figure 2.

C. Paper Organization

The rest of the paper is organized as follows. In Section II, we describe the formulation of the Private Multi-Group Aggregation problem. We present the details of the Q&A scheme in Section III, and our main result in Theorem 1. In Section IV, we describe the details of the RG scheme, compare the two schemes and elaborate on the setup of Figure 2. A more detailed discussion of the results, including proofs, can be found in the full version [19].

II. Problem Formulation

We consider the setting depicted in Figure 1 in which there are n users, indexed from 1 to n, and a single server. The users can communicate with the server but not among each other. Each user $i \in [n] := \{1, \ldots, n\}$ belongs to one of k groups, indexed from 1 to k. Moreover, user $i \in [n]$ holds a value $v_i \in \mathcal{V} := \{\pm 1, \ldots, \pm m\}$. We assume that the server knows each user's index but does not know his value or group. We assume that the users are not adversarial and will faithfully participate in the scheme.

We denote by G_i the random variable representing the group that user i belongs to. We assume that G_i, for all $i \in [n]$, are identical and independent random variables from the alphabet $\mathcal{G} = \{1, \ldots, k\}$. The probability that any user $i \in [n]$ belongs to group $g \in \mathcal{G}$ is denoted by $\theta_g := \Pr(G_i = g)$. We assume that $\theta := (\theta_1, \ldots, \theta_k)$ is a realization of the random vector Θ. Each user i in group g holds an independent random scalar value V_i drawn from the alphabet $\mathcal{V} := \{\pm 1, \ldots, \pm m\}$ according to the distribution $p_g(v) := \Pr(V_i = v | G_i = g)$. The values of the users in the same group are i.i.d. We summarize the users' value distributions, $p_g(v)$, by a $k \times 2m$ matrix p. The matrix p is unknown and is assumed to be the realization of a random variable P. Given their group $g \in \mathcal{G}$, for all $v \in \mathcal{V}$, the users behave identically, *i.e.,* $p_g(v) = \Pr(V_i = v | G_i = g)$ for any $i \in [n]$.

User i knows the realization of the random variables G_i and V_i representing his group and value. However, the distribution of the random variables P and Θ, and their realizations, are not necessarily known neither to the server nor to the user.

The goal is to design a scheme that allows the server to compute an estimate of the sum of values per group, *i.e.,* to estimate the aggregate vector $\mathbf{S} \in \mathbb{Z}^k$ with

$$\mathbf{S}(g) = \sum_{i \in [n]: g_i = g} v_i, \quad \text{for all } g \in \mathcal{G}.$$

We consider schemes where each user i can be assigned a query $q_i \in \mathcal{Q}$, which is also known to the server. In response to the query, the user sends the server an answer $a_i \in \mathcal{A}$. Upon receiving the answers from all users, the server finds an estimate $\hat{\mathbf{S}}$ of $\mathbf{S}$. We characterize the efficiency of a scheme according to (i) communication, (ii) accuracy, and (iii) privacy.

(i) ***Communication:*** We characterize the communication cost by the number of bits communicated between the server and the user. We look at the communication cost from two vantage points: (i) user-centric, that measures the communication per user, *i.e.,* the number of bits communicated between a user and the server; (ii) server-centric, that measures the total communication the server receives from all the users.

(ii) ***Accuracy:*** We use a normalized mean square error to measure the accuracy of a scheme π, *i.e.,* the risk of the estimator $\hat{\mathbf{S}}_\pi$ is

$$\mathcal{E}_\pi := \frac{1}{n^2}\text{MSE}(\hat{\mathbf{S}}_\pi),$$

where $\text{MSE}(\hat{\mathbf{S}}_\pi) = \mathbb{E}\left[||\hat{\mathbf{S}}_\pi - \mathbf{S}||_2^2 \big| P = p, \Theta = \theta\right]$. When it is clear from context, the conditioning on P and Θ will be implicit. The normalization factor accounts for the true aggregate, $\mathbf{S}$, being proportional to the number of users, n.

(iii) ***Privacy:*** We focus on keeping a user's group private but not necessarily his value. We use local differential privacy [20], [21] as our measure for the desired privacy level of a user's group. Since a user's value and group can be correlated, it will sometimes be necessary (depending on the required privacy level) to also hide a user's value in addition to his group. To that end, a user's answer to the server will be the output of a randomized mechanism $\mathcal{M} : \mathcal{G} \times \mathcal{V} \times \mathcal{Q} \to \mathcal{A}$ that outputs a user's answer a belonging to an alphabet $\mathcal{A}$ based on his value, group, query and local randomness.

Definition 1. Let ϵ be a positive real number, and $\mathcal{M}$ be a randomized mechanism. We say $\mathcal{M}$ is ϵ-locally differentially private with respect to the group if for any $g, g' \in \mathcal{G}$, $q \in \mathcal{Q}$, and $a \in \mathcal{A}$,

$$\begin{aligned}&\Pr(\mathcal{M}(G, V, Q) = a | G = g, Q = q, P = p, \Theta = \theta) \leq \\ &e^\epsilon \Pr(\mathcal{M}(G, V, Q) = a | G = g', Q = q, P = p, \Theta = \theta), \quad (1)\end{aligned}$$

where the probability is taken over the randomness of the mechanism and the random variable V.

The probabilities in the local differential privacy definition are taken given the realizations of the random variables P and Θ. Even though the server does not necessarily know these realizations, the privacy definition above assumes this knowledge. This is needed because, with enough answers collected from users, the server can infer information about the distributions of P and Θ.

III. The Query and Aggregate (Q&A) Scheme

In this section, we describe the Q&A scheme. In Section III-A, an example illustrates the key ideas of this scheme and gives intuition about the proof of Theorem 1. In Section III-B, we give the description of the general (Q&A) scheme.

A. A 1-bit Example: Two groups and a binary alphabet

We focus on the special case of two groups, $k = 2$, and a binary alphabet, $\mathcal{V} = \{-1, 1\}$. In this case, the Q&A scheme needs only a single bit of communication per user.

Scheme Description: It is composed of the following steps.

1) *Queries:* Each user i responds to a random query q_i which is a 2 by 2 matrix. More specifically, the query q_i is chosen uniformly at random from the set

$$\mathcal{Q} = \left\{ \left[\begin{smallmatrix} -1 & +1 \\ +1 & -1 \end{smallmatrix}\right], \left[\begin{smallmatrix} -1 & +1 \\ -1 & +1 \end{smallmatrix}\right], \left[\begin{smallmatrix} +1 & -1 \\ -1 & +1 \end{smallmatrix}\right], \left[\begin{smallmatrix} +1 & -1 \\ +1 & -1 \end{smallmatrix}\right] \right\}.$$

   The user's assigned query is independent of his group and value. Moreover, it is assumed that the server knows the queries assigned to each user.

2) *User's answer*: Each user sends the server a 1-bit answer, a_i, depending on the query he received. The user will only look at the *row* of the query matrix that corresponds to his group, *i.e.,* row 1 if he is in group 1 and row 2 if he is in group 2. He answers with the index of the *column* that contains his value, *i.e.,* $a_i = 1$ or $a_i = 2$.

3) *Server's estimation:* The server receives the 1-bit answer a_i from each user i. He maps the answer into the vector $q_i(:, a_i) := (q_i(1, a_i), q_i(2, a_i))$, *i.e.,* the a_i^{th} column of the query matrix q_i. This is possible because he knows the user's assigned query. Then, the server forms the estimates of the aggregate for each group, such that

$$\hat{\mathbf{S}}_{\text{QA}} = \sum_{i=1}^{n} q_i(:, a_i).$$

Next we give a brief analysis of the scheme in terms of accuracy (MSE), privacy, and communication cost.

Accuracy: We show that the normalized mean square error goes to zero as the number of users increases, allowing the server a better estimate of the true aggregate $\mathbf{S}$.

Without loss of generality, let us consider, $\mathbf{S}(1)$, the aggregate corresponding to group 1. Then, we have its estimate

$$\begin{aligned}\hat{\mathbf{S}}_{\text{QA}}(1) &= \sum_{i \in [n]: g_i = 1} q_i(1, a_i) + \sum_{i \in [n]: g_i = 2} q_i(1, a_i) \\ &= \underbrace{\mathbf{S}(1)}_{\substack{\text{True Aggregate} \\ \text{for Group 1}}} + \underbrace{\sum_{i \in [n]: g_i = 2} q_i(1, a_i)}_{\text{Noise}}. \qquad (2)\end{aligned}$$

Therefore, the estimate $\hat{\mathbf{S}}_{\mathrm{QA}}(1)$ can be interpreted as the true aggregate with an added noise term. The noise corresponds to the contribution of the users who do not belong to group 1. Since the queries were assigned uniformly at random, the distribution of the answers corresponding to the noise is uniform and independent of the true aggregate $\mathbf{S}(1)$. It follows from our choice of query matrices that the contribution to the estimate, of each user i in group 2, $q_i(1, a_i)$, is a realization of the random variable,

$$Q_i(1, A_i) = \begin{cases} -1 & \text{with probability } \frac{1}{2}, \\ +1 & \text{with probability } \frac{1}{2}. \end{cases} \tag{3}$$

Therefore, the noise is the sum of i.i.d. random variables with bounded variance that converges to a zero mean additive Gaussian noise. This implies that the expectation of the norm of the noise grows as $\mathcal{O}(\sqrt{n})$. And indicates that the normalized mean square error, $\mathcal{E}_{\mathrm{QA}}$, goes to zero as $\mathcal{O}(n^{-1})$.

Privacy: We show that the Q&A scheme is ϵ_{QA} locally differentially private. From Definition 1,

$$e^{\epsilon_{\mathrm{QA}}} = \max_{\substack{g,g' \in \{1,2\}, \\ a \in \{1,2\}, q \in \mathcal{Q}}} \frac{\Pr(A_i = a | G_i = g, Q_i = q)}{\Pr(A_i = a | G_i = g', Q_i = q)}. \tag{4}$$

The first thing we notice is that the ratio in (4) is equal to 1 for $g = g'$, and the maximum is always greater than or equal to 1 when $g \neq g'$. Therefore, we can limit the maximization in (4) to $g \neq g'$. Moreover, a user's value ($+1$ or -1) is a deterministic function of the answer, the query, and the group. Therefore, we can simplify (4) to

$$\begin{aligned} e^{\epsilon_{\mathrm{QA}}} &= \max_{\substack{g,g' \in \{1,2\}, g \neq g' \\ v,v' \in \{-1,1\}, q \in \mathcal{Q}}} \frac{\Pr(V_i = v | G_i = g, Q_i = q)}{\Pr(V_i = v' | G_i = g', Q_i = q)} \\ &= \max_{\substack{g,g' \in \{1,2\}, g \neq g' \\ v,v' \in \{-1,1\}}} \frac{p_g(v)}{p_{g'}(v')}, \end{aligned} \tag{5}$$

which follows from the independence of the random variables representing the user's value, V_i, and his assigned query, Q_i, and from the definition of $p_g(v) = \Pr(V_i = v | G_i = g)$. Thus, we obtain an expression of the privacy which only depends on the users' value distributions.

Notice that not all privacy levels can be guaranteed for fixed user value distributions $p_1(\cdot)$ and $p_2(\cdot)$. The reason is that, in its basic form, the Q&A scheme described above, does not guarantee privacy over the user's value. And since the user's value and group can be correlated, the user's value can leak information about his group. To mitigate this, in Section III-B, we add a second layer of privacy in order to protect a user's value. This gives flexible privacy guarantees that do not depend only on the user's value distributions.

Communication: Since the user's answer $a_i \in \{1, 2\}$, the scheme's communication cost is one bit per user. Moreover, we show in Theorem 1 that the general scheme's communication cost is always 1 bit per user when the alphabet is binary, irrespective of the number of groups. This is the fundamental limit on the zero-error communication cost if there were no groups and no privacy requirements.

Note that the query assignment must be known to both the server and the user. This can be accomplished without incurring communication cost. For instance, it can be implemented as the output of a public hash function that takes as input the user's index $i \in [n]$, or simply considered part of the scheme agreement that does not depend on a user's group and value.

B. The General Q&A Scheme

Here we describe the Q&A scheme, for any number of groups $k \geq 2$, and alphabet parameter $m \in \mathbb{N}$. It is obtained by generalizing the query matrices of the previous example.

1) **Queries:** Each user is assigned a random query matrix of dimension $k \times 2m$ and elements in $\mathcal{V} = \{\pm 1, \ldots, \pm m\}$. The query matrices assigned to each user are chosen independently and uniformly at random from the set $\mathcal{Q}$ defined as

$$\mathcal{Q} := \left\{ q \in \mathcal{V}^{k \times 2m} \middle| q(g, :) \in \textbf{Sym}(\mathcal{V}) \text{ for all } g \in [k] \right\}, \tag{6}$$

where $q(g, :) = (q(g, 1), \ldots, q(g, k))$, and $[k] := \{1, \ldots, k\}$. And $\textbf{Sym}(\mathcal{V})$ is the set of all row vectors which are an ordered permutation of the finite set $\mathcal{V}$.[1] Each row of a matrix $q \in \mathcal{Q}$ is a permutation of all the possible $2m$ values. Notice that the values cannot be repeated within a row but rows can be repeated. We designate by q_i the query assigned to user i.

The assumption is that the server also knows the query assigned to the user. As previously mentioned, since the query does not depend on the user's group or value, it could be assigned offline as part of the scheme agreement, or implemented as the output of a public hash function.

2) **User's Answer:** Given his assigned query, user i first privatizes his value using a randomized response [18], [22] step parameterized by $\lambda \in \left[0, 1 - \frac{1}{2m}\right)$.That is, given his true value v_i, the user chooses a randomized value $\mathring{v}_i$ such that

$$\Pr(\mathring{V}_i = \mathring{v}_i | V_i = v_i) = \begin{cases} 1 - \lambda & \text{for } \mathring{v}_i = v_i \\ \frac{\lambda}{2m-1} & \text{for } \mathring{v}_i \in \mathcal{V} - \{v_i\}. \end{cases} \tag{7}$$

Then, user i looks at the g_i^{th} row (g_i is the user's group) of the query matrix q_i, and sends to the server the answer a_i, which is the index of the column that has his randomized value $\mathring{v}_i$. More precisely, a_i is such that $q_i(g_i, a_i) = \mathring{v}_i$.

3) **Server's Estimation:** Upon receiving user i's answer, the server maps it into the a_i^{th} column of query q_i, *i.e.,*

$$q_i(:, a_i) = (q_i(1, a_i), q_i(2, a_i), \ldots, q_i(k, a_i))^\top.$$

The server sums the decoded answers from all the users, and multiplies by an unbiasing term, to find the estimate of the true aggregate, $\mathbf{S}$, *i.e.,*

$$\hat{\mathbf{S}}_{\mathrm{QA}} = \frac{2m - 1}{2m - 2m\lambda - 1} \sum_{i \in [n]} q_i(:, a_i). \tag{8}$$

Below we give an example of a possible query and its answer.

Example. Consider the setting where there are $k = 3$ groups and the alphabet of values is $\mathcal{V} = \{\pm 1, \pm 2\}$. Let $\lambda = 0$, *i.e.,*

[1] An ordered permutation of a set $\mathcal{V}$ is a vector where each element is a distinct element of $\mathcal{V}$, *e.g.,* $\textbf{Sym}(\{\pm 1, \pm 2\})$ has $4! = 24$ elements including $(-2, -1, 1, 2)$ and $(1, 2, -2, -1)$.

$V_i = \mathring{V}_i$. Suppose user 1 has value $v_1 = -1$ and belongs to group $g_1 = 2$. For instance, if he is assigned the query

$$q_1 = \begin{bmatrix} -2 & -1 & +1 & +2 \\ -2 & +1 & -1 & +2 \\ +2 & -1 & -2 & +1 \end{bmatrix},$$

then his answer is $a_1 = 3$, because his value $v_1 = -1$ is the third element of the second row (corresponding to his group $g_1 = 2$) of q_1. Upon receiving this answer, the server decodes it into the third column of q_1, *i.e.,* $q_1(:, a_1) = (+1, -1, -2)^\top$.

We note a few characteristics of the design of the queries and answers. The user's value is always one of the elements of the row corresponding to his group. And looking at the user's mapped answer $q_i(:, a_i)$, *i.e.,* a column vector of his assigned query q_i, his value is in row g_i of q_i. As for all the other vector elements, $q_i(:, a_i)$, they are uniformly distributed over $\mathcal{V}$. This follows from the queries' design and mirrors (3).

Theorem 1 characterizes the communication cost, accuracy and privacy of the Q&A scheme, and is proved in [19].

Theorem 1 (Q&A Scheme)**.** *Given a setting with n users, k groups, alphabet $\mathcal{V} = \{\pm 1, \dots, \pm m\}$, and the users' value distribution $p_g(v)$ for all $g \in \mathcal{G}, v \in \mathcal{V}$, the Query and Aggregate scheme (Q&A) satisfies the following properties.*

1) *The Q&A scheme has a communication cost of* $\log(2m)$ *bits per user.*
2) *The Q&A scheme is ϵ_{QA}-LDP with*

$$e^{\epsilon_{\text{QA}}} = \max_{\substack{v,v' \in \mathcal{V}, \\ g,g' \in \mathcal{G}, g' \neq g}} \left\{ \frac{(2m(1-\lambda)-1)p_g(v)+\lambda}{(2m(1-\lambda)-1)p_{g'}(v')+\lambda} \right\}, \qquad (9)$$

where the randomization parameter $\lambda \in \left[0, \frac{2m-1}{2m}\right)$.

3) *The estimator of the Q&A scheme is unbiased and has normalized mean square error*

$$\mathcal{E}_{\text{QA}} = \alpha n^{-1},$$

where $\alpha = \frac{2m\lambda\mathbb{E}[V_i^2]}{2m-2m\lambda-1} + \frac{(4m^2-1)(m+1)[(2m-1)(k-1)+2m\lambda]}{6(2m-2m\lambda-1)^2}$.
The normalized mean square error is $\mathcal{O}\left(\frac{km^4}{n}\right)$.

Remark (The choice of λ)**.** Given a required privacy parameter ϵ, the parameter λ that can guarantee this given ϵ is determined using (9). However, this requires the knowledge of the value distributions, $p_g(\cdot)$ for all $g \in \mathcal{G}$. Nevertheless, one can still use (9) to find a bound on λ that is independent of the users' value distributions such that

$$\lambda \geq \frac{2m-1}{2m+e^\epsilon-1}.$$

This bound can be tightened if some side information is known about the users' value distributions. For instance, suppose that $c_{\min} < p_g(v) < c_{\max}$ for all $g \in \mathcal{G}$ and $v \in \mathcal{V}$, for some constants $c_{\max}, c_{\min} \in [0,1]$, $c_{\max} > c_{\min}$. In this case, the following tighter bound can be shown

$$\lambda \geq \frac{(2m-1)c_{\max} - c_{\min}e^\epsilon}{2m(c_{\max}-c_{\min}e^\epsilon)+e^\epsilon-1}.$$

Evidently, smaller values of λ are better for accuracy because the mean square error is increasing in λ.

IV. The Randomized Group Scheme

To better gauge the performance of the Q&A scheme we compare it to the Randomized Group (RG) scheme which directly privatizes a user's group by adding noise to it through a randomized response step. In RG, each user i sends the server the answer $a_i = (\mathring{g}_i, \mathring{v}_i)$ of his privatized group and value respectively. More precisely, $\mathring{g}_i$ is chosen randomly according to the distribution

$$\Pr(\mathring{G}_i = \mathring{g}_i | G_i = g_i) = \begin{cases} 1-\lambda_{gr} & \text{for } \mathring{g}_i = g_i \\ \frac{\lambda_{gr}}{k-1} & \text{for } \mathring{g}_i \in [k] - \{g_i\}, \end{cases}$$

where g_i is user i's group and the parameter $\lambda_{gr} \in (0,1)$. As for the value $\mathring{v}_i$, there are two cases:

1) $\mathring{g}_i \neq g_i$: In this case, the user chooses $\mathring{v}_i$, uniformly at random, *i.e.,* $\Pr(\mathring{V}_i = \mathring{v}_i | \mathring{g}_i \neq g_i) = \frac{1}{2m}$ for all $\mathring{v}_i \in \mathcal{V}$. This choice ensures that when users lie about their groups, the aggregate of their contribution has a zero mean.
2) $\mathring{g}_i = g_i$: In this case, the user lies about his true value with probability $\lambda_{vl} \in \left[0, 1-\frac{1}{2m}\right)$. That is, he randomly chooses a value, $\mathring{v}_i$, according to the distribution

$$\Pr(\mathring{V}_i = \mathring{v}_i | V_i = v_i, \mathring{g}_i = g_i) = \begin{cases} 1-\lambda_{vl} & \text{for } \mathring{v}_i = v_i, \\ \frac{\lambda_{vl}}{2m-1} & \text{for } \mathring{v}_i \in \mathcal{V} - \{v_i\}. \end{cases}$$

The server aggregates the received answers and re-scales the aggregate to unbias the estimator, such that, for all $g \in [k]$ the estimate of the true aggregate of group g, $\mathbf{S}(g)$, is

$$\hat{\mathbf{S}}_{\text{RG}}(g) := \frac{2m-1}{(1-\lambda_{gr})(2m(1-\lambda_{vl})-1)} \sum_{i: a_i(1)=g} a_i(2).$$

Note that there are no queries assigned to users in this scheme.

Comparison of the RG and Q&A Schemes

The Q&A scheme has a communication cost ($\log(2m)$ bits per user) that does not depend on the number of groups k and beats the RG scheme ($\log(2km)$ bits per user).

To compare the two schemes on all fronts, we fix the total communication cost, *i.e.,* the number of bits communicated by all the users to the server, and compare the privacy vs. accuracy trade-offs. Figure 2 illustrates this comparison and the typical observation in our experiments, of varying alphabet sizes and number of groups, is the following two privacy regimes

(i) High Privacy Regime: for small values of the privacy parameter, ϵ, the Q&A scheme outperforms the RG scheme.
(ii) Low Privacy Regime: for large enough privacy parameter, ϵ, the RG scheme outperforms the Q&A scheme. This is because, as ϵ goes to infinity, the error of the Q&A scheme converges to a constant strictly greater than zero as we cannot further tune the parameters of the scheme. On the other hand, the error of the RG scheme converges to zero.

The performance of the RG scheme, and missing proofs and details can be found in [19].

Acknowledgment

The authors would like to thank Peter Kairouz for helpful discussions.

References

[1] S. Kim, M. K. Sung, and Y. D. Chung, "A Framework to Preserve the Privacy of Electronic Health Data Streams," *Journal of Biomedical Informatics*, vol. 50, pp. 95–106, 2014.

[2] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, B. H. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical Secure Aggregation for Privacy-Preserving Machine Learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017.

[3] M. Abadi, A. Chu, I. Goodfellow, B. H. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep Learning with Differential Privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[4] S. Truex, L. Liu, K.-H. Chow, M. E. Gursoy, and W. Wei, "LDP-Fed: Federated Learning with Local Differential Privacy," in *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, 2020.

[5] M. Kim, O. Günlü, and R. F. Schaefer, "Federated Learning with Local Differential Privacy: Trade-offs between Privacy, Utility, and Communication," *arXiv:2102.04737*, 2021.

[6] T. H. Chan, E. Shi, and D. Song, "Optimal Lower Bound for Differentially Private Multi-party Aggregation," in *European Symposium on Algorithms*, 2012.

[7] B. Ghazi, N. Golowich, R. Kumar, P. Manurangsi, R. Pagh, and A. Velingker, "Pure Differentially Private Summation from Anonymous Messages," in *1st Conference on Information-Theoretic Cryptography (ITC 2020)*, 2020.

[8] S. Goryczka and L. Xiong, "A Comprehensive Comparison of Multiparty Secure Additions with Differential Privacy," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 5, pp. 463–477, 2017.

[9] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, and Y. Zhou, "A Hybrid Approach to Privacy-Preserving Federated Learning," in *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, 2019.

[10] E. Shi, T. H. Chan, E. Rieffel, R. Chow, and D. Song, "Privacy-Preserving Aggregation of Time-Series Data," in *Proceedings of the 18th Annual Network & Distributed System Security Symposium Proceedings (NDSS)*, 2011.

[11] H. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

[12] J. So, B. Guler, and A. S. Avestimehr, "Turbo-Aggregate: Breaking the Quadratic Aggregation Barrier in Secure Federated Learning," *arXiv:2002.04156*, 2020.

[13] S. Kadhe, N. Rajaraman, O. O. Koyluoglu, and K. Ramchandran, "FastSecAgg: Scalable Secure Aggregation for Privacy-Preserving Federated Learning," *arXiv:2009.11248*, 2020.

[14] V. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust Aggregation for Federated Learning," *arXiv:1912.13445*, 2019.

[15] C. Niu, F. Wu, S. Tang, L. Hua, R. Jia, C. Lv, Z. Wu, and G. Chen, "Secure Federated Submodel Learning," *arXiv:1911.02254*, 2019.

[16] M. Kim and J. Lee, "Information-Theoretic Privacy in Federated Submodel learning," *arXiv:2008.07656*, 2020.

[17] Z. Jia and S. A. Jafar, "X-Secure T-Private Federated Submodel Learning," *arXiv:2010.01059*, 2020.

[18] S. L. Warner, "Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias," *Journal of the American Statistical Association*, vol. 60, pp. 63–69, 1965.

[19] C. Naim, R. G. L. D'Oliveira, and S. El Rouayheb, "Private Multi-Group Aggregation," 2021. [Online]. Available: http://eceweb1.rutgers.edu/~csi/PMGA_full.pdf

[20] C. Dwork, "Differential Privacy," in *The 33rd International Colloquium on Automata, Languages and Programming (ICALP)*, 2006.

[21] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *IEEE 54th Annual Symposium on Foundations of Computer Science*, 2013.

[22] P. Kairouz, K. Bonawitz, and D. Ramage, "Discrete Distribution Estimation under Local Privacy," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning (ICML)*, 2016.