PRIVACY VS. CORRELATION IN INFORMATION RETRIEVAL AND AGGREGATION

By

CAROLINA NAIM

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Salim El Rouayheb

And approved by

_____

_____

_____

_____

_____

New Brunswick, New Jersey

January 2023

ABSTRACT OF THE DISSERTATION

Privacy vs. Correlation in Information Retrieval and Aggregation

by CAROLINA NAIM

Dissertation Director: Salim El Rouayheb

Privacy is now a major challenge encountered by users who can unknowingly reveal critical personal information through their online activities. Due to the correlation over time between the different behaviors of an online user or the correlation between his attributes, care should be taken when proposing privacy solutions. The main goal of this dissertation is to explore this interplay between privacy and correlation. To that end, we consider two problems that examine this tension, (i) ON-OFF privacy with correlated requests and (ii) private multi-group aggregation.

We start by considering the problem of ON-OFF privacy in which a user is interested in the latest message generated by one of n sources available at a server. The user has the choice to turn privacy ON or OFF depending on whether he wants to hide his interest at the time or not. The challenge is that the statistical correlation over time of a user's online behavior can lead to information leakage. As a consequence of correlation, the user cannot simply ignore privacy when privacy is OFF. We model the correlation between a user's requests by an n state Markov chain. Our goal is to design ON-OFF privacy schemes with optimal download rates that ensure privacy for past and past and future requests. We present inner and outer bounds on the achievable download rate for n sources. We also devise an efficient algorithm to construct an ON-OFF privacy scheme achieving the inner bound and prove its optimality for special families of Markov chains, such as in the case of $n = 2$ sources. In general, for $n > 2$, finding tighter outer bounds and efficient constructions of ON-OFF privacy schemes that would achieve them remains an open problem.

We then 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. 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. The challenge is that the user's group and value can be correlated. 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. We compare Q&A to the Randomized Group (RG) scheme, which is non-interactive and adapts existing randomized response schemes to the PMGA setting. We observe that typically Q&A outperforms RG, in terms of privacy vs. utility, in the high privacy regime.

ACKNOWLEDGMENTS

I want to express my deepest appreciation to my advisor and dissertation director Prof. Salim El Rouayheb. His unwavering commitment and patience helped me strive in the most challenging times, even during the chaos of the pandemic. Our meetings and conversations were vital in inspiring me to think outside the box, form comprehensive and objective scientific arguments, effectively present results, and ultimately become the researcher I am today.

I also extend my profound gratitude to my defense committee members: Prof. Emina Soljanin, Prof. Predrag Spasojevic, Prof. Anand Sarwate, and Prof. Mohammad Ali Maddah-Ali. Their help cannot be overestimated, as their thoughtful comments, recommendations, and constructive feedback steered me through my research.

I am grateful to all of those with whom I had the pleasure to work with during my Ph.D. I would like to extend my sincere thanks to Rafael D'Oliveira; his consistent support and guidance have been invaluable in my pursuit of this degree. He helped me develop my skills, answered my unending questions, and shaped my understanding of research. I also wish to thank Fangwei Ye, with whom I had the pleasure of collaborating with on my first paper, in addition to multiple other projects, and whose insight, extensive knowledge, and attention to detail inspired me.

Special thanks to my labmates and friends, Rawad Bitar, Serge Kas Hanna, Ghadir Ayache, and Zonghong Liu. Our insightful group meetings, discussions, lab dinners, and social activities helped me throughout my journey.

I would also like to acknowledge the assistance of the electrical and computer engineering department for their patience in answering all my inquiries and providing me with the resources to pursue my graduate studies. To my professors at Rutgers, thank you for your commitment to delivering a high-caliber education and for equipping me with the right tools to advance my knowledge.

I am also indebted to my undergraduate institution and professors. Particularly, I would like to thank Prof. Mustafa El-halabi for believing in my abilities and shaping the path leading to my Ph.D.

Many thanks to my good friends, both in New Jersey and Lebanon, for keeping my spirits high and motivating me every step of the way. Furthermore, as I fulfill this chapter of my life, I cannot begin to express my gratitude to my parents and sister, whose love and unconditional support are with me in whatever I pursue and are the foundation of all my achievements.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Motivation

Privacy is now at the forefront of the challenges encountered by online users when participating in online activities, such as watching videos, using location services, engaging in social networks, participating in learning algorithms, and more. Privacy vulnerabilities can have ethical, societal, and political implications as users can unknowingly reveal critical personal information (age, sex, diseases, political bias, etc.). Privacy research has gathered much attention in the recent years, attracting interest from academic researchers [1–3], tech companies [4, 5], and lawmakers [6, 7], where various privacy requirements and formulations are proposed.

One proposed solution to mitigate online privacy challenges is to grant individuals more control in deciding the level of privacy they require [6, 7]. However, due to the correlation in a user's online behavior, the proposed solutions can be misleading in terms of guaranteed privacy. For example, consider the setting of users in a social network. Each user can decide his privacy level on his own. However, the privacy choices that his friends in the social network make can severely jeopardize his privacy due to correlation. This tension between correlation and privacy also appears even when dealing with a single user's behavior. Imagine a user choosing in his privacy settings to allow an application on his smartphone to access his location only when the application is active. The user's location should not be directly revealed to the application as soon as it is activated since this will leak information on the user's previous locations due to correlation in time.

This dissertation explores this interplay between privacy and correlation in two main problems, which we refer to as (i) ON-OFF privacy for correlated requests and (ii) private multi-group aggregation. In ON-OFF privacy for correlated requests, we suppose that the user is given the option to set his privacy to be ON or OFF each time he wants to retrieve a message from a server. His choice for his privacy setting may depend on incentives from the service provider, his location, or the machine and network he is using. Due to the correlation in a user's online behavior over time, he should be careful when turning his privacy OFF so that he does not unintentionally leak information

about his requests when his privacy is ON. In the private multi-group aggregation setting, mainly motivated by learning and analytics applications, e.g., [8], users want to participate in an aggregation algorithm run by a server. Each user has a value and belongs to a private group. The server wants to find the aggregate (sum) of values in each group. Moreover, the user's group and value can be correlated, so suppose the user were to directly reveal his value and focus only on hiding his group. In that case, he might still leak private information about his group because of its correlation with his value – say, if a particular group tends to output a specific value more frequently.

We formulate and propose solutions to the problems summarized below.

(**1**) **ON-OFF Privacy for Past Correlated Requests:** We consider the setting in which a user is interested in retrieving the latest message generated by one of n sources stored at a server. The user's requests are correlated over time, and we model this correlation by an n state Markov chain. We assume the user can turn privacy ON or OFF as needed. The user wants to keep his past requests for which privacy was ON private in an information-theoretic sense, i.e., perfect privacy. We prove a fundamental upper bound on the download rate. Moreover, we give a polynomial time algorithm that preserves ON-OFF privacy and prove its optimality for $n = 2$ sources [9, 10].

(**2**) **ON-OFF Privacy for Past and Future Correlated Requests:** Here, we consider the more general privacy setting in which the user wants to hide both past and future requests for which privacy is ON. Moreover, we assume here that the user knows his requests in the future in a time window of size $\omega > 0$. For this setting, we provide similar results to (1). More precisely, we give an upper bound on the download rate and design a polynomial time algorithm that protects past and future requests for which privacy is ON [11, 12].

(**3**) **Private Multi-Group Aggregation:** In the third part of the dissertation, we consider a second problem that explores privacy vs. correlation. The setting here consists of n users, each belonging to one of k groups and holding a discrete value. The server's goal is to find the aggregate, i.e., the sum, of values for each group separately with high accuracy under communication and local differential privacy constraints. The privacy constraint is such that the user's group is the sensitive information that should be private. The challenge is that a user's group and value can be correlated. We design a communication-efficient query-based scheme that maintains the required privacy and study the trade-offs between privacy, accuracy and communication cost [13, 14].

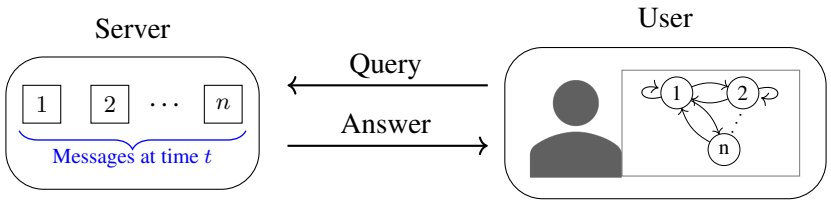


Figure 1.1: ON-OFF privacy setting. At time t the server generates n messages. The user sends a query, which may be a function of all previous requests $\{X_i : i \leq t\}$ generated by the n-state Markov chain, and privacy statuses (ON or OFF). The server replies with an answer, which is a function of its stored messages. The objective is to allow the user to obtain the message he wants with minimum communication cost while keeping his requests, for which privacy is ON, private.

In the rest of this chapter, we give an overview of these three topics and describe our main contributions. We provide further details and connect them to related work in the literature in each of their respective chapters.

1.2 ON-OFF Privacy for Past Correlated Requests

In Chapter 2, we start by considering ON-OFF privacy where the user's requests are correlated over time, focusing on protecting the privacy of the user's *past* requests for which privacy was turned ON (privacy is turned OFF in the future). Our results were published in [9, 10].

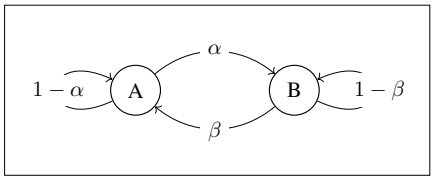
1.2.1 Setting

We consider the setup depicted in Figure 1.1 in which there are n information sources each generating a new message at each time $t \in \mathbb{N}$. At each time t, the user randomly chooses one of the sources and requests its latest generated message by sending the server a query Q_t. We focus on the simplest non-trivial correlation model given by an n-state Markov chain to study how correlation affects privacy. The privacy constraint we consider is information-theoretic: the user wants to leak zero information about the identity of the sources in which he is interested in at each time t, i.e., X_t, when his privacy is ON. More precisely,

$$I\left(X_{\mathcal{B}_t}; Q_t | Q_0, Q_1, \ldots, Q_{t-1}\right) = 0, \quad \text{for all } t \in \mathbb{N}, \tag{1.1}$$

where $\mathcal{B}_t := \{i : i \leq t, \text{privacy is ON}\}$, i.e., the past times privacy was ON.

When privacy is ON, we know from the literature on private information retrieval with a single



(a)

X_τ, X_t \ Q_t	A	B	AB
A, A	$\frac{\beta}{1-\alpha}$	0	$\frac{1-\alpha-\beta}{1-\alpha}$
A, B	0	1	0
B, A	1	0	0
B, B	0	$\frac{\alpha}{1-\beta}$	$\frac{1-\alpha-\beta}{1-\beta}$

(b)

Figure 1.2: (a) A two-state Markov chain representing the correlation of the user's requests $X_t, t \in \mathbb{N}$. (b) The optimal ON-OFF privacy scheme for $n = 2$ that achieves the bound in (1.2) for $\alpha + \beta < 1$. The query Q_t is probabilistic and depends on the current request X_t, the previous query Q_{t-1} and the last private request X_τ. The scheme consists of the following two cases: (i) if $Q_{t-1} = \{AB\}$, i.e., the previous query was for two messages, then the current query $Q_t = X_t$; (ii) if $Q_{t-1} \neq \{AB\}$, i.e., the previous query was for one message, then the current query Q_t is chosen based on the probabilities $p\left(q_t | x_\tau, x_t, q_{t-1}\right)$ given in this table.

server [3] that we have to download everything to preserve privacy. However, when privacy is OFF, simply ignoring the privacy requirement may reveal information about the requests when privacy is ON because the user's requests are correlated over time. Thus, the goal is to design an ON-OFF privacy scheme with the maximum download rate, i.e., minimum download cost, that guarantees the privacy of the user's requests for which his privacy is ON (1.1), despite of correlation over time.

1.2.2 Main Contributions

The main results of Chapter 2 can be summarized as follows,

(i) We prove achievable rates obtained by our ON-OFF privacy schemes which have polynomial time complexity in the number of sources n.

(ii) We give general upper bounds on the download rate for $n \geq 2$ information sources.

(iii) For Markov chains with $n = 2$ sources, depicted in Figure 1.2(a), we show that our proposed scheme is optimal, i.e., the upper bound is tight.

For instance, for $n = 2$, let $X_t \in \{A, B\}$ be the user's download request at time $t \in \mathbb{N}$ modeled as a two-state Markov chain, where the transition probabilities are given by $\alpha = \Pr(X_{t+1} = B \mid X_t = A)$ and $\beta = \Pr(X_{t+1} = A \mid X_t = B)$. The rate R_t of a scheme at time t is measured by the ratio of the average length of the downloaded data to the message length. We prove the following result that fully characterizes the ON-OFF privacy optimal rate for $n = 2$ sources.

Theorem (Theorem 3 in Chapter 2)**.** For $n = 2$ sources, the rate tuple $(R_t : t \in \mathbb{N})$ is achievable if and only if

$$\frac{1}{R_t} \geq 1 + |1 - \alpha - \beta|^{t-\tau}, \tag{1.2}$$

where $\tau = \max\{i : i \leq t, \text{privacy is ON}\}$, i.e., the last time privacy was ON.

Our scheme, described in Figure 1.2b, achieves the rate in (1.2) when $\alpha + \beta < 1$. Our scheme is probabilistic such that the user either requests the message he wants, i.e., $Q_t = X_t$, or requests both messages, i.e., $Q_t = \{AB\}$. Notice that in both cases, the user is guaranteed to receive the message he wants, i.e., $X_t \in \{A, B\}$. At some point, when the previous query is for both messages, i.e., $Q_{t-1} = \{AB\}$, the user can directly ask for the desired message at time t without being concerned about leaking information about X_τ. Moreover, when privacy is ON, i.e., $t = \tau$, the user has to query for both messages, i.e., $Q_t = \{AB\}$, this is directly reflected in (1.2) as $\frac{1}{R_t} = 2$ for $t = \tau$.

We revisit this result with more details in Chapter 2.

1.3 ON-OFF Privacy for Past and Future Correlated Requests

In Chapter 3, we extend the setting of Chapter 2 to encompass the more stringent requirement of protecting the user's *past and future* requests for which privacy is turned ON. The results of this topic were published in [11, 12].

1.3.1 Setting

Here we consider a more stringent privacy requirement than Section 1.2 and want to preserve privacy for both past and future requests. We follow a setup similar to the one above in which a user's correlated requests are modeled by Markov chain, but with one significant difference. We assume that the user knows the requests in a small window of positive size ω in the future.[1] In practice, this may happen in applications where the user can queue up his requests.

Let X_t be the user's request at time t, and let Q_t be his query to the server at time t. The privacy constraint we consider is the following

[1] If the window size $\omega = 0$, i.e., no future requests are known, we have to relax the the stringent privacy requirement in this work to a weaker sense where only past requests are protected, i.e., the model described in Section 1.2.

$$I\left(X_{\overline{\mathcal{B}}_t}; Q_t | Q_0, Q_1, \ldots, Q_{t-1}\right) = 0, \quad \text{for all } t \in \mathbb{N},$$

where $\overline{\mathcal{B}}_t := \{i : i \leq t, \text{privacy is ON}\} \cup \{i : i \geq t+1\}$, i.e., the past times privacy was ON in addition to all future times. We assume that the user does not know whether privacy is ON or OFF in the future. Therefore, in the privacy definition above, we have adopted a worst-case formulation which assumes that privacy is always ON in the future.

1.3.2 Main Contributions

The main takeaways are similar to Section 1.2, and our main results can be summarized as follows

(i) We give an achievable rate obtained by an ON-OFF privacy scheme having polynomial time complexity in the number of sources n.

(ii) We prove a general upper bound on the download rate for $n \geq 2$ sources.

(iii) Our proposed scheme is optimal, i.e., the upper bound is tight, for all Markov chains with $n = 2$ sources.

Here we also show the following result, which a corollary to Theorem 4 in Chapter 3, that fully characterizes the optimal rate for ON-OFF privacy for past and future correlated requests when $n = 2$ sources.

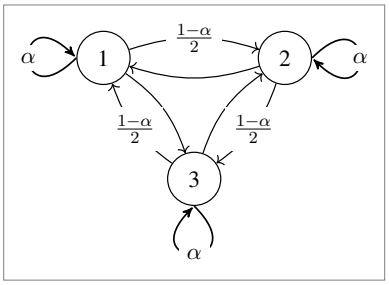
Corollary (Corollary 2 in Chapter 3)**.** For $n = 2$ sources, the rate tuple $(R_t : t \in \mathbb{N})$ is achievable if and only if

$$\frac{1}{R_t} \geq \sum_{x_t} \max_{x_\tau, x_{t+1}} p(x_t | x_\tau, x_{t+1}), \tag{1.3}$$

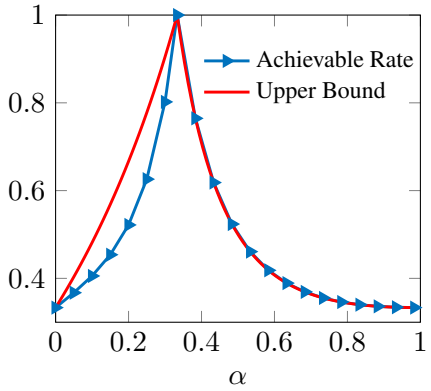
where $\tau = \max\{i : i \leq t, \text{privacy is ON}\}$, i.e., the last time privacy was ON.

Note that the rate in (1.3) does not depend on the window size ω. Thus, a window of size $\omega = 1$ is sufficient to achieve the optimal rate, and we show that this is also true for any $n \geq 2$.

For $n \geq 2$ sources, in addition to proving a general upper bound on the download rate, we fully characterize it for a family of symmetric Markov chains. A Markov chain is symmetric if its transition matrix P is given by



(a)



(b)

Figure 1.3: (a) A graphical representation of the 3-state symmetric Markov chain where $0 \leq \alpha \leq 1$. (b) A plot of the achievable rate and the upper bound as a function of α, when $\tau = 0$ and $t = 1$ in the setting of ON-OFF privacy for past and future correlated requests. The two bounds match for all $\frac{1}{n} \leq \alpha \leq 1$. More details about this result are in Section 3.4.

$$P_{i,j} = \begin{cases} \alpha, & \text{if} \quad i = j, \\ (1-\alpha)/(n-1), & \text{if} \quad i \neq j, \end{cases}$$

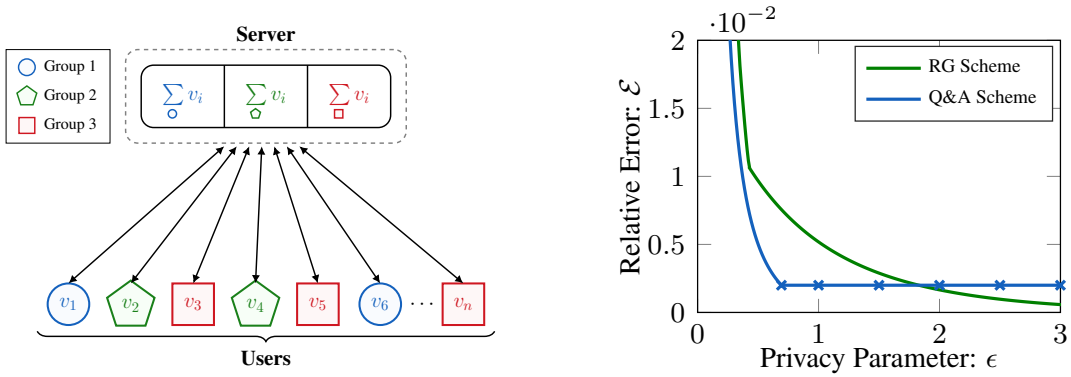
where $0 \leq \alpha \leq 1$ and $P_{i,j} = p(x_t = j | x_{t-1} = i)$ denotes the transition probability from state i to state j, e.g., see Figure 1.3a.

Corollary (Corollary 3 in Chapter 3)**.** For the symmetric Markov chain such that $\frac{1}{n} \leq \alpha \leq 1$, the rate tuple $(R_t : t \in \mathbb{N})$ is achievable if and only if

$$\frac{1}{R_t} \geq \alpha n \frac{(n-1)^{t-\tau} + (n-1)(n\alpha - 1)^{\tau}}{(n-1)^{t-\tau} + (n\alpha - 1)^{t-\tau+1}}. \tag{1.4}$$

In Figure 1.3b, we give an example of a symmetric Markov chain with $n = 3$ sources and plot the achievable rate and upper bound. We have two regimes, one for $\alpha < \frac{1}{3}$ and the other for $\alpha \geq \frac{1}{3}$. For $\alpha \geq \frac{1}{3} = \frac{1}{n}$, the rate is optimal, i.e., the upper bound is tight, which is also reflected by (1.4).

While we characterize the optimal rate for special cases of the Markov chain for $n > 2$, the optimal rate and how to achieve it in polynomial time remains in general an open problem. However, in Chapter 3, we give implicit characterizations of the optimal achievable rates, which rely on a linear program (LP) formulation, albeit with an exponential number (in n) of variables and constraints. More details can be found in Chapter 3.



(a) (b)

Figure 1.4: (a) An instance of the Private Multi-Group Aggregation problem with n users. Each user i, for $i \in \{1, \ldots, n\}$, has a scalar 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. (b) Privacy vs. Utility comparison of the Q&A and RG schemes for $k = 2$ groups, binary alphabet, i.e., $v_i \in \{-1, 1\}$, and fixed total communication cost. The Q&A scheme outperforms the RG scheme in the high privacy regime (small ϵ).

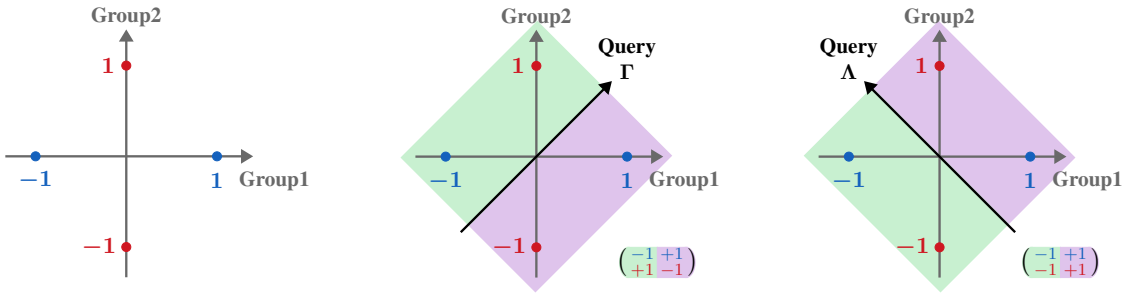
1.4 Private Multi-Group Aggregation

In the work discussed in Chapter 4, we focus on a different aspect of privacy vs. correlation by designing private distributed aggregation schemes under communication constraints. The results of this work were published in [13, 14].

1.4.1 Setting

The literature on private aggregation [4, 15–22] has made a lot of progress on the different aspects of this problem. Nevertheless, in our work, we introduce a variation of private aggregation: private multi-group aggregation (PMGA). It is motivated by the concept that users can belong to different groups based on their race, age, gender, or political views, to name a few. And the server wants a finer aggregate of the population, so he wants the aggregate for each group separately.

In our setting depicted in Figure 1.4(a), we consider n users that communicate with a central server. Each user $i \in \{1, 2, \ldots, n\}$ belongs to one of k private groups in addition to holding an integer value, $v_i \in \{\pm 1, \pm 2, \ldots, \pm m\}$ where $m \in \mathbb{N}^+$. The server's goal is to accurately compute the sum of values per group. We assume that the server does not know which group each user belongs to, and a major concern for the user is to reveal his group. This is because revealing his group can have serious consequences, for example, discrimination based on race or age or even



(a) User's mapping based on value and group.

(b) Query represented by Γ and its corresponding matrix.

(c) Query represented by Λ and its corresponding matrix.

Figure 1.5: Depiction of the queries and answers of the Q&A Scheme for two groups and a binary alphabet. The user's value and group are mapped to one of the four points, as depicted in (a). The user is assigned one of two queries, represented by a directed line, as in sub-figures (b) and (c). The user sends the server an answer corresponding to whether his mapping is left (green) or right (violet) of the directed line. The corresponding matrices to each query are further described in Section 4.4.

higher health insurance bills if the groupings were related to diseases. The notion of privacy we use is local differential privacy [2, 23].

There are three aspects that are important to the PMGA setting: (i) privacy, (ii) communication cost, or the number of bits communicated by the user to the server, (iii) and utility, i.e, the accuracy of the algorithm which is the aggregate estimator's mean square error (MSE).

1.4.2 Main Contributions

Our main contribution for PMGA is the novel scheme we call Query and Aggregate (Q&A). One of the key characteristics of Q&A is that it is an interactive aggregation scheme. More precisely, each user is assigned a query, to which he sends the server an answer based on his group and value. We show that using this query-based scheme can significantly decrease communication costs while maintaining privacy and good accuracy.

To illustrate the key ideas of this scheme, here we focus on the special case of two groups, $k = 2$, and a binary alphabet, $v_i \in \{-1, 1\}$ for all users $i \in \{1, \ldots, n\}$. In this special case,[2] the user starts by mapping his group and value to one of four points in a two dimensional space, as depicted in Figure 1.5a. Each axis corresponds to a group and contains two points, corresponding to the values in that group. For example, if the user has value $+1$ and is in group 1, he maps his

[2]Here, we describe a different, yet equivalent, interpretation of the Q&A scheme than that described in Section 4.4.1. We use this interpretation here because it can be easily visualized (as in Figure 1.5).

group and value to point $(+1, 0)$. Then the scheme proceeds as follows.

1. *Queries:* Each user responds to one of two queries, Γ or Λ, known to the server. A query corresponds to the question: is the user's mapping to the left (green) or right (violet) of line Γ in Figure 1.5b (or similarly line Λ in Figure 1.5c). The queries are chosen uniformly at random and independent or a user's group and value.

2. *User's answers*: Each user sends the server a 1-bit answer to the query he received. A 1 if his mapping is to the right (violet) of the query, and a 0 if it is to the left (green).

3. *Server's estimation:* The server receives the 1-bit answer from each user i. Each answer corresponds to a region containing two points, one point on the Group 1 axis and the other point on the Group 2 axis. The server maps the 1-bit answer he received from user i into a two dimensional vector $\hat{a}_i$. The first coordinate of $\hat{a}_i$ is the value of the point on the Group 1 axis and the second coordinate is the value of the point on the Group 2 axis. For example, if the user's answer to query Γ is right (violet), then it is mapped to $(+1, -1)^\top$. After decoding the answers, the server forms the estimates of the the aggregate for each group by summing the decoded two dimensional vectors $\hat{a}_i$.

Observe that the user's group remains private because the server receives an answer corresponding to a region of two possible mappings. The server cannot determine with certainty whether the user belongs to group 1 or 2. In Section 4.4, we make this precise by showing that the Q&A scheme provides local differential privacy guarantees and elaborate on the effects that the correlation between the user's group and value has on privacy.

We generalize the Q&A scheme for a larger alphabet and more groups in Section 4.4 by representing the queries as matrices as opposed to the directed lines depicted in Figure 1.5. We characterize its performance in terms of accuracy (MSE), privacy, and communication cost in Theorem 5. Furthermore, we compare Q&A to the Randomized Group (RG) scheme, which is non-interactive and adapts existing randomized response schemes [24] to the PMGA setting. We observe, as in Figure 1.4(b), that typically Q&A outperforms RG, in terms of privacy vs. utility, in the high privacy regime. We revisit and elaborate on this comparison in Section 4.6.

1.5 Organization

The rest of this dissertation is organized as follows.

- In Chapter 2, we detail our results for ON-OFF privacy for correlated requests where the privacy requirement focuses on protecting past requests.
- In Chapter 3, we extend these results to the stronger privacy requirement of protecting both past and future requests.
- In Chapter 4, we describe the setting and our results for private multi-group aggregation.
- Finally, we conclude and discuss future directions in Chapter 5.

CHAPTER 2
ON-OFF PRIVACY FOR PAST CORRELATED REQUESTS

2.1 Introduction

The implicit assumption that is common in existing privacy models is that the user wants privacy *all the time*. We refer to it as privacy being always ON. However, as we discussed in Chapter 1, privacy-preserving algorithms incur high costs on the service provider, and can lead to degraded quality of service at the user's side. Therefore, one should think of privacy as an expensive commodity, which should be turned ON only when needed (depending on geographical location, device, network, etc.). This motivated us to introduce and study the problem of ON-OFF privacy [9, 10]. ON-OFF privacy algorithms enable a user to switch his privacy between ON and OFF.

One may be tempted to propose the simple solution in which the user has available to him two schemes, one private and one non-private. Over time, the user simply switches between these two schemes depending on whether privacy is turned ON or OFF. The problem with this solution is that it guarantees privacy only if the user's online activities are statistically independent over time. However, a user's online activities are typically personal, making them correlated over time. For example, a bilingual English/Spanish user, who is checking the news in Spanish now, is more likely to keep reading the news in Spanish for a while before switching to English. At that point, English becomes more probable. Another example is when the user is watching online videos. One may think of a scenario where the user is more likely to watch the top item from a list of recommended videos that depends on the previously watched videos. Thus, due to correlation, ignoring the privacy requirement when privacy is OFF may reveal information about the activities when privacy was ON.

2.1.1 Example

To be more concrete and to gently introduce our setup for ON-OFF privacy, we give the following example. Suppose a user is watching political or news videos online. At each time t, the user has a choice between two new videos each of which is produced by two different news sources, A or B. Source A is politically left-leaning and source B is right-leaning.

Let $X_t \in \{A, B\}$ be the source whose video the user wants to watch at time $t \in \mathbb{N}$. We model the correlation among the user's requests by assuming that X_t is the two-state Markov chain depicted in Figure 2.1, where the transition probabilities are given by $\alpha = \Pr(X_{t+1} = B \mid X_t = A)$ and $\beta = \Pr(X_{t+1} = A \mid X_t = B)$. For illustration, we choose $\alpha = \beta = 0.2$. This means that if the current video being watched is left-leaning, there is an 80% chance that the next video is also left-leaning, and vice versa.

For the sake of brevity, we focus on the two time instants $t = 0$ and $t = 1$, and assume that privacy is ON at $t = 0$ and is switched to OFF at $t = 1$. This means that the user would like to hide whether he was watching a left-leaning or a right-leaning video at time $t = 0$, but does not care about revealing the source of the video he watched at $t = 1$.

The goal is to devise an ON-OFF privacy scheme that always gives the user the video he wants, but never reveals the choice of sources when privacy is ON, i.e., $t = 0$ in this case. More precisely, the server observes queries at both times $t = 0$ and $t = 1$, i.e., Q_0 and Q_1, which should be independent of the user's interest at time $t = 0$ when privacy was ON, i.e., X_0. We are interested in schemes that minimize the download cost, or equivalently maximize the download rate (the inverse of the normalized download cost).

At $t = 0$, the problem is simple. The user achieves privacy by downloading both videos. We say that the user's query at $t = 0$ is $Q_0 = AB$. Therefore, the download rate at $t = 0$ is $R_0 = 1/2$.

At $t = 1$, the privacy is OFF. Now, the user must be careful not to directly declare his request, because this may reveal information about his request at $t = 0$ which is to remain private. The user can again download both videos, i.e., $Q_1 = AB$, and achieve privacy with a rate $R_1 = 1/2$.

Our key result is that the user can achieve a better expected rate at $t = 1$, without compromising privacy, by

- choosing randomly between downloading A ($Q_1 = A$) or both A and B ($Q_1 = AB$) if he wants $X_1 = A$,

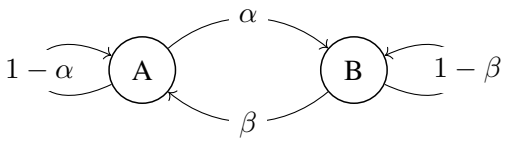


Figure 2.1: The two-state Markov chain representing the correlation of the user's requests $X_t, t \in \mathbb{N}$.

X_0	X_1	$Q_1 = A$	$Q_1 = B$	$Q_1 = AB$
A	A	0.25	0	0.75
A	B	0	1	0
B	A	1	0	0
B	B	0	0.25	0.75

Table 2.1: An example of our ON-OFF privacy scheme for $\alpha = \beta = 0.2$. The query Q_1 at $t = 1$ is a probabilistic function of X_0 and X_1, the requests at $t = 0$ and $t = 1$ respectively. The entries of the table represent the probabilities $p(Q_1 \mid X_0, X_1)$, where $Q_1 = AB$ means that the user downloads the videos from both sources A and B.

- choosing randomly between downloading B ($Q_1 = B$) or both A and B ($Q_1 = AB$) if he wants $X_1 = B$.

This random choice must also depend on the request X_0 at $t = 0$. The different probabilities defining the scheme are given in Table 2.1 and will be justified later when we explain the general scheme. For now, one can check that these probabilities lead to

$$\Pr(Q_1 = q) = \Pr(Q_1 = q \mid X_0 = x_0),$$

for any $q \in \{A, B, AB\}$ and any $x_0 \in \{A, B\}$. Thus, X_0 and Q_1 are independent and the proposed scheme in Table 2.1 achieves perfect privacy for the request at $t = 0$. Moreover, the scheme ensures that the user always obtains the video he is requesting.

For $t = 1$, the rate $R_1 = 1/(2 - \alpha - \beta) = 0.625$, which is strictly greater than 0.5, the rate of querying both files. We later show that this rate is actually optimal. In fact, the values in Table 2.1 were carefully chosen to achieve the privacy at the highest download rate. Any other choice of the probabilities $p(Q_1 \mid X_0, X_1)$ would either violate privacy or lose the optimality of the rate.

2.1.2 Related Work

The study of information-theoretic measures for privacy has received significant interest in the literature (see for e.g. [25–29]). The closest setting to the ON-OFF privacy problem studied in this work is that of private information retrieval (PIR) from a single server [3], which can be viewed as a special case (when privacy is always ON) of the ON-OFF privacy problem. In this case, it is known that to achieve information-theoretic privacy, the user must download all the messages, except in the case when the user has some side information [30, 31]. Recently, there has been significant

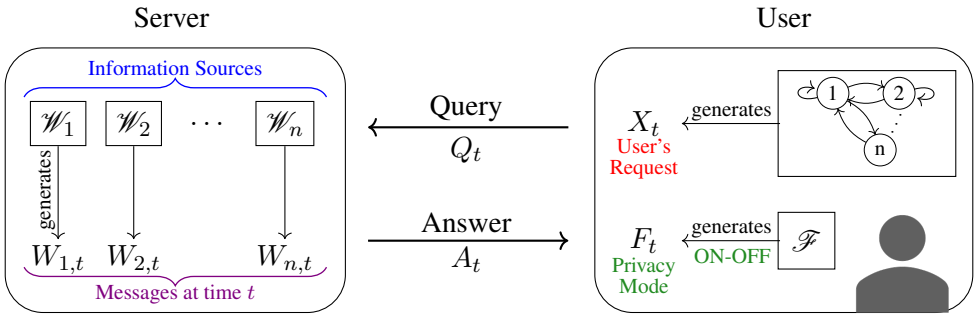


Figure 2.2: The setting at time t as described in Section 2.2.1. The server stores messages $W_{1,t}, \ldots, W_{n,t}$ generated by information sources $\mathscr{W}_1, \ldots, \mathscr{W}_n$, respectively. The user sends a query Q_t, which may be a function of all previously generated requests $\{X_i : i \leq t\}$ and privacy status $\{F_i : i \leq t\}$. The server replies with the answer A_t, which is a function of $W_{1,t}, \ldots, W_{n,t}$.

progress on PIR with multiple servers with a focus on download rate and coded data (e.g., [32–36] and references therein). However, the model there requires multiple servers and, in the parlance of this work, privacy is assumed to be always ON.

A related problem that considers privacy with correlation, namely location privacy, was studied in [37–45]. The privacy notions studied therein include k-anonymity [37, 38], (extended) differential privacy [39–41], and distortion privacy [42, 43], which all differ from the information-theoretic privacy measure studied in this work. The works of [44, 45] recently studied the information-theoretic privacy measure in location-privacy protection mechanisms, and their privacy metric was defined by the mutual information between the released data and the true traces. In this work's language, it can be viewed as the case when privacy is always ON. However, in this work, we want to prevent the adversary from inferring a selective part of the requests specified by an ON or OFF privacy status, and the simple time-sharing (switching between a private and a non-private scheme according to the privacy status) approach is not permissible due to the correlation.

The concept of ON-OFF privacy was also applied to preserve privacy of sensitive genotypes in genomics in [46].

2.1.3 Contributions

In this work, we introduce a model to capture the ON-OFF privacy problem when the user is downloading data from online sources. We consider the setup in which there are n information sources each generating a new message at each time t. At each time t, the user randomly chooses one of the

sources and requests its latest generated message.

The privacy constraint is information theoretic: the user wants to leak zero information about the identity of the sources in which he is interested in at each time t when the privacy is ON. The main challenge stems from the fact that the user's requests are not independent. As in the previous example, we model the dependence between these requests by an n-state Markov chain, and we assume that the transition probabilities of the Markov chain are known by both the user and the adversary, where the known transition probabilities can be viewed as public information estimated from a large population. The goal is to design an ON-OFF privacy scheme with the maximum download rate that satisfies the user's request and guarantees the privacy of the requests made when privacy is ON.

Our main contribution consists of giving general outer and inner bounds on the download rate in Theorems 1 and 2, respectively. We also devise an efficient algorithm to construct an ON-OFF privacy scheme achieving the inner bound. We prove optimality of the achievable scheme for $n = 2$ sources. For $n > 2$, finding tighter outer bounds and efficient constructions of ON-OFF privacy schemes that would achieve them remains an open question.

The rest of the chapter is organized as follows. In Section 2.2, we describe the formulation of the ON-OFF privacy problem. We present our main results in Section 2.3. The proof of the converse and achievability will be given in Section 2.4 and 2.5, respectively. A computational perspective will be discussed in Section 2.6, and the optimality for $n = 2$ sources will be discussed in Section 2.7.

2.2 Problem Formulation and Notation

2.2.1 Setting

A single server stores n information sources $\{\mathscr{W}_i : i \in \mathcal{N}\}$, where $\mathcal{N} := \{1, 2, \ldots, n\}$. The system is time-varying, and the time index t is assumed to be discrete throughout this chapter, i.e., $t \in \mathbb{N}$. Without loss of generality, we assume that each source $\mathscr{W}_i$ generates a message $W_{i,t}$ consisting of L symbols at each time t, independently and identically according to the uniform distribution over $\{0, 1\}^L$. Such that $\{W_{i,t} : i \in \mathcal{N}, t \in \mathbb{N}\}$ are mutually independent, i.e.,

$$H\left(W_{i,t} : i \in \mathcal{N}, t \in \mathbb{N}\right) = \sum_{i,t} H\left(W_{i,t}\right), \tag{2.1}$$

and

$$H(W_{i,t}) = L \quad \forall i \in \mathcal{N}, t \in \mathbb{N}. \tag{2.2}$$

At each time t, the user is interested in retrieving the latest message generated by a desired source, i.e., one of the messages from $\{W_{i,t} : i \in \mathcal{N}\}$. In particular, let X_t be the source of interest at time t, which takes values in $\mathcal{N}$. In the sequel, we will call X_t the *user's request* at time t. Since the user is always interested in the latest message generated at time t, we slightly abuse the notation by dropping t from $W_{i,t}$ when the time index t is clear in the context, i.e., $W_{i,t}$ will be written as W_i and we may write the retrieved message as W_{X_t}.

As mentioned previously, we are particularly interested in the case where the requests X_t, for $t \in \mathbb{N}$, form a time-invariant Markov chain, i.e., $\{X_t : t \in \mathbb{N}\}$ is generated by a Markov source $\mathscr{X}$. The transition matrix P of the Markov chain is known by both the server and the user, and the transition probability from state i to state j is denoted by $P_{i,j}$. We also denote the initial probability distribution of the Markov chain by π_0.

The user may or may not wish to hide the identity of his source of interest at time t. Specifically, the *privacy status* F_t at time t can be either ON or OFF, where F_t is ON when the user wishes to keep X_t private, and F_t is OFF when the user is not concerned with privacy. Denote $\mathcal{F} = \{\text{ON}, \text{OFF}\}$. We assume that the privacy status $\{F_t : t \in \mathbb{N}\}$ is generated by some information source $\mathscr{F}$ that is independent of the user's requests $\{X_t : t \in \mathbb{N}\}$. We also assume that at time any t, $\{F_i : i \leq t\}$ is known by both the server[1] and the user. For the ease of notation, we assume that $F_0 = \text{ON}$.

As discussed in Section 2.1, if the user directly downloads the desired message at time t when the privacy is OFF, the privacy in the past may be compromised. To guarantee his privacy, the user is allowed to generate unlimited local randomness, and in this work, we are not interested in the amount of randomness used. The local randomness S_t for $t \in \mathbb{N}$ is assumed to take values in a common alphabet $\mathcal{S}$.

In this chapter, we only consider a *causal* system. Specifically, at time t, the user may utilize the *causal* information, i.e., all the previous and current requests $\{X_i : i \leq t\}$, previous and current privacy status $\{F_i : i \leq t\}$, and the previously generated randomness $\{S_i : i < t\}$, to construct a query Q_t, which he sends to the server. In other words, the randomness S_t may be generated

[1]It is worth noting that in our formulation we are not interested in hiding the privacy status from the server.

according to $\{X_i : i \leq t\}$, $\{F_i : i \leq t\}$ and $\{S_i : i < t\}$, i.e.,

$$S_t \sim p_{X_{\lfloor t \rfloor}, F_{\lfloor t \rfloor}, S_{\lfloor t-1 \rfloor}}, \tag{2.3}$$

where $\lfloor t \rfloor := \{0, 1, \ldots, t\}$ and $X_{\lfloor t \rfloor} := \{X_i : i = 0, 1, \ldots, t\}$. Note that (2.3) encompasses the case in which the current query also depends on the previous queries, since they are also functions of $\{X_i : i \leq t\}$, $\{F_i : i \leq t\}$ and $\{S_i : i < t\}$.

Upon receiving the query Q_t, the server responds to the request by producing the answer A_t consisting of $\ell\left(Q_t\right)$ symbols, where A_t is a function of Q_t and messages $\{W_{i,t} : i = 1, \ldots, n\}$, The length of the answer A_t is a function of the query Q_t received. Thus, the average length of the answer A_t is given by

$$\ell_t = \mathbb{E}_{Q_t}[\ell\left(Q_t\right)]. \tag{2.4}$$

It is worth noting that Q_t should depend on the initial distribution π_0 of the Markov chain. However, since the rest of the discussion holds for any π_0, we drop it for ease of notation. Notice that ℓ_t is well defined for any π_0 because $\ell\left(Q_t\right)$ is trivially bounded by $n\,L$, i.e., downloading all n messages.

2.2.2 Encoding and Decoding Functions

Definition 1. An $(n, \mathscr{X}, \mathscr{F})$ causal ON-OFF privacy system consists of the following encoding and decoding functions:

- Query encoding function:

$$\gamma_t : \mathcal{N}^t \times \mathcal{F}^t \times \mathcal{S}^t \to \mathcal{Q}, \;\; t = 0, 1, 2, \ldots,$$

  where γ_t maps all previous (including current) requests and privacy status, together with the local randomness, to the query at time t, i.e., $Q_t = \gamma_t\left(X_{\lfloor t \rfloor}, F_{\lfloor t \rfloor}, S_{\lfloor t \rfloor}\right)$.

- Answer length function:

$$\ell : \mathcal{Q} \to \{0, 1, \ldots, nL\},$$

  i.e., the length of the answer at time t is a deterministic function of the current query, which is independent of a particular message and not time-varying over time t.

- Answer encoding function:

$$\rho_t : \mathcal{Q} \times \{0,1\}^{nL} \to \{0,1\}^{\ell(\mathcal{Q})}, \ \ t = 0, 1, 2, \ldots,$$

where ρ_t maps the current query and n latest messages to the answer of length $\ell(Q_t)$, i.e., $A_t = \rho_t\left(Q_t, W_{1,t}, \ldots, W_{n,t}\right)$.

- Message decoding function:

$$\psi_t : \{0,1\}^{\ell(\mathcal{Q})} \times \mathcal{N} \times \mathcal{S} \to \{0,1\}^{L}, \ \ t = 0, 1, 2, \ldots,$$

where ψ_t maps the received answer to the desired message, i.e.,

$$\hat{W}_{X_t} = \psi_t\left(A_t, X_t, S_t\right).$$

We would like to emphasize two points about the setup of the model. First, for any given causal privacy status $\{F_i : i \leq t\}$ at time t, the query Q_t may be treated as a stochastic function of all causal requests $\{X_i : i \leq t\}$ and previous queries $\{Q_i : i < t\}$. Since we are not interested in the randomness $\{S_i : i \leq t\}$ consumed, we will not write the local randomness explicitly in the sequel. Second, since messages $\{W_{i,t} : i \in \mathcal{N}\}$ are independent over time, at a given time t, the answer A_t only depends on the latest generated messages $W_{1,t}, \ldots, W_{n,t}$. Similarly, the current query Q_t is independent of previous answers $\{A_i : i < t\}$.

2.2.3 Privacy and Decodability

These functions need to satisfy the decodability and the privacy constraints, defined as follows.

1. Decodability: For any time t, the user should be able to recover the desired message from the answer with zero-error probability, i.e.,

$$\Pr\left(\hat{W}_{X_t} \neq W_{X_t}\right) = 0. \tag{2.5}$$

2. Privacy: For any time t, given all past queries received by the server, the query Q_t should not

reveal any information about all the past or present requests when the privacy is ON, that is

$$I\left(X_{\mathcal{B}_t}; Q_t | Q_{\lfloor t-1 \rfloor}\right) = 0, \quad \forall t \in \mathbb{N}, \tag{2.6}$$

where $\mathcal{B}_t := \{i : i \leq t, F_i = \text{ON}\}$. For notational simplicity, F_0 is assumed to be ON throughout this chapter, and hence $\mathcal{B}_t$ is always not empty.

The conditioning in the privacy formulation in (2.6) serves to ensure causality in the proposed achievable schemes. Barring this conditioning, privacy could be alternatively defined by

$$I\left(X_{\mathcal{B}_t}; Q_{\lfloor t \rfloor}\right) = 0, \quad \forall t \in \mathbb{N}. \tag{2.7}$$

However, this alternative definition implies that at any point $i < t$, the user has to know and protect future requests $\{X_j : j = i+1, \ldots, t, F_j = \text{ON}\}$, since (2.7) implies that

$$I\left(X_{\mathcal{B}_t \setminus [i]}; Q_i\right) = 0,$$

which contradicts the causality of the system.

Given the definition of the privacy, we introduce the following proposition, which is a direct but useful consequence of the Markov assumption of the requests and the privacy definition and whose proof can be found in Appendix A.1.

Proposition 1. If X_τ is independent of Q_t conditioning on $Q_{\lfloor t-1 \rfloor}$, then $X_{\mathcal{B}_t}$ is independent of Q_t conditioning on $Q_{\lfloor t-1 \rfloor}$.

By convention, at time t, the tuple ℓ_t is said to be achievable if there exists a code satisfying the decodability and the privacy constraint such that the average answer length is ℓ_t. The efficiency of the code can be measured by the download rate $R_t = \frac{L}{\ell_t}$, and hence we define the achievable region as follows.

Definition 2. The rate tuple $(R_t : t \in \mathbb{N})$ is achievable if there exists a code with message length L and average download cost ℓ_t such that $R_t \leq L/\ell_t$ for all $t \in \mathbb{N}$.

We are interested in characterizing the achievable region $(R_t : t \in \mathbb{N})$. In particular, the focus of this chapter is the characterization of R_t for each $t \in \mathbb{N}$.

2.2.4 Notation

We introduce some necessary notation which will be used in later sections. Let $\tau(t)$ be the last time privacy was ON, i.e.,

$$\tau(t) := \max\{i : i \leq t, F_i = \text{ON}\} = \max \mathcal{B}_t. \tag{2.8}$$

The time index t will be clear in the context in the following sections, so we may drop t from the notation and write τ instead of $\tau(t)$ for simplicity. It is worth noting that $\tau(t)$ is well-defined because of the assumption that $F_0 = \text{ON}$.

For any given $x \in \mathcal{N}$ and $q_{\lfloor t-1 \rfloor}$, suppose that we have the following ordering of the likelihood probabilities

$$\begin{aligned} p\left(X_t = x | X_\tau = x_\tau^{(x,1)}, Q_{\lfloor t-1 \rfloor} = q_{\lfloor t-1 \rfloor}\right) &\leq p\left(X_t = x | X_\tau = x_\tau^{(x,2)}, Q_{\lfloor t-1 \rfloor} = q_{\lfloor t-1 \rfloor}\right) \\ &\leq \cdots \leq p\left(X_t = x | X_\tau = x_\tau^{(x,n)}, Q_{\lfloor t-1 \rfloor} = q_{\lfloor t-1 \rfloor}\right), \end{aligned} \tag{2.9}$$

where $x_\tau^{(x,i)}$ for $i = 1, \ldots, n$ are distinct elements in $\mathcal{N}$. Then, for $i = 1, \ldots, n$, let

$$\lambda_i\left(t, q_{\lfloor t-1 \rfloor}\right) = \sum_{x \in \mathcal{N}} p\left(X_t = x | X_\tau = x_\tau^{(x,i)}, Q_{\lfloor t-1 \rfloor} = q_{\lfloor t-1 \rfloor}\right), \tag{2.10}$$

and

$$\theta_i\left(t, q_{\lfloor t-1 \rfloor}\right) = \min\left\{1, \lambda_i\left(t, q_{\lfloor t-1 \rfloor}\right)\right\} - \min\left\{1, \lambda_{i-1}\left(t, q_{\lfloor t-1 \rfloor}\right)\right\}, \tag{2.11}$$

where $\lambda_0\left(t, q_{\lfloor t-1 \rfloor}\right)$ is assumed to be 0. For notational simplicity, we will also write $\lambda_i\left(t, q_{\lfloor t-1 \rfloor}\right)$ by $\lambda_i\left(q_{\lfloor t-1 \rfloor}\right)$ and $\theta_i\left(t, q_{\lfloor t-1 \rfloor}\right)$ by $\theta_i\left(q_{\lfloor t-1 \rfloor}\right)$ when the time index t is clear in the context.

Moreover, we will use $\mathscr{P}(\mathcal{N})$ to denote the power set of $\mathcal{N}$, and $\mathbb{E}[X]$ to denote the expected value of a random variable X. We summarize some definitions and nomenclature in Table 2.2.

2.3 Main results

In this section, we present the main results of this chapter, i.e., inner and outer bounds for the achievable region $(R_t : t \in \mathbb{N})$.

The following theorem gives an outer bound on the achievable rate, and the proof can be found

Symbol	Definition
n	number of sources
$\mathcal{N}$	$\{1, 2, \ldots, n\}$
$\lfloor t \rfloor$	$\{0, 1, \ldots, t\}$ for any $t \in \mathbb{N}$
$\mathscr{P}(\mathcal{N})$	power set of $\mathcal{N}$
$W_{i,t}$	message generated by i-th source at time t, where $i = 1, \ldots, n$ and $t = 0, 1, \ldots$
X_t	user's request at time t ($X_t \in \mathcal{N}$)
F_t	privacy status at time t, i.e., $F_t \in \{\text{ON}, \text{OFF}\}$
Q_t	query sent by the user to the server at time t
A_t	answer sent by the server to the user at time t
$\mathcal{B}_t$	all the times privacy was ON , i.e., $\mathcal{B}_t = \{i : i \leq t, F_i = \text{ON}\}$
$\tau(t)$	last time privacy was ON, i.e., $\tau(t) = \max \mathcal{B}_t$
ℓ_t	average length of the answer A_t
R_t	download rate at time t
$\lambda_i\left(q_{\lfloor t-1 \rfloor}\right)$	the summation of i-th minimal likelihood probabilities (of x_τ) provided the observation x_t for given $q_{\lfloor t-1 \rfloor}$

Table 2.2: Nomenclature and definitions

in Section 2.4.

Theorem 1. (Outer bound 1) The rate tuple $(R_t : t \in \mathbb{N})$ must satisfy

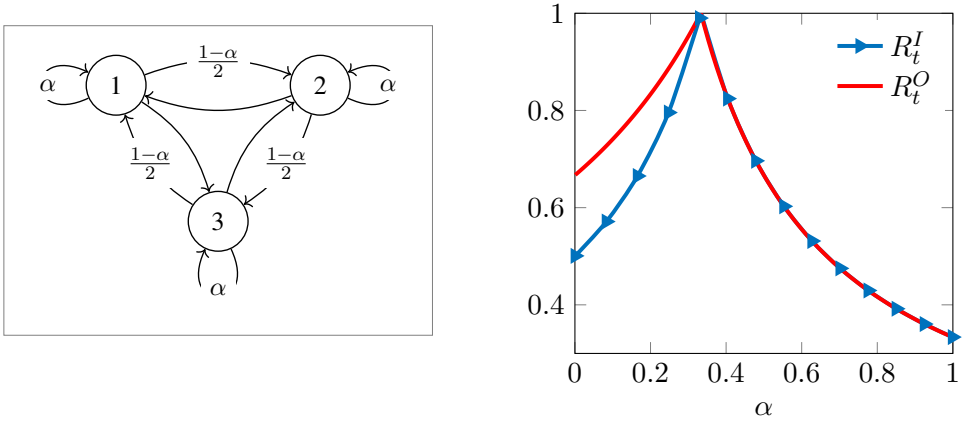
$$\frac{1}{R_t} \geq \sum_{q_{\lfloor t-1 \rfloor}} p\left(q_{\lfloor t-1 \rfloor}\right) \sum_{x_t} \max_{x_\tau} p\left(x_t | x_\tau, q_{\lfloor t-1 \rfloor}\right), \tag{2.12}$$

where $\tau = \max\{i : i \leq t, F_i = \text{ON}\}$.

It is worth noting that the right-hand side of (2.12) encompasses the previous queries, where the optimal previous queries maximizing the download rate for the current time instance are implicit, and hence the bound in (2.12) is generally hard to compute. Nevertheless, we can use the bound in (2.12) to derive the following corollary, which only involves the transition probabilities of the Markov chain and not the previous queries.

Corollary 1. (Outer Bound 2) The rate tuple $(R_t : t \in \mathbb{N})$ must satisfy

$$\frac{1}{R_t} \geq \sum_{x_t} \max_{x_\tau} p\left(x_t | x_\tau\right), \tag{2.13}$$



(a) A Symmetric Markov Chain.

(b) R_t^I and R_t^O as a function of α.

Figure 2.3: In Figure 2.3a, we graphically represent the 3-state symmetric Markov chain used in Example 1, where $0 \leq \alpha \leq 1$. In Figure 2.3b, we plot the achievable rate R_t^I and the upper bound R_t^O (c.f.(2.17)), as a function of α, when $\tau = 0$ and $t = 1$.

where $\tau = \max\{i : i \leq t, F_i = \text{ON}\}$.

Proof. See Appendix A.2. □

The following theorem gives an inner bound on the rate, and the detailed description of the achievable scheme will be discussed in Section 2.5.2.

Theorem 2. (Inner bound) The rate tuple $(R_t : t \in \mathbb{N})$ is achievable if

$$\frac{1}{R_t} \geq \sum_{q_{\lfloor t-1 \rfloor}} p\left(q_{\lfloor t-1 \rfloor}\right) \sum_{i=1}^{n} i\, \theta_i(q_{\lfloor t-1 \rfloor}). \tag{2.14}$$

We give the following example to illustrate the outer and inner bounds described in Theorem 1 and Theorem 2, respectively.

Example 1. Consider a symmetric Markov chain with transition matrix P given by

$$P_{i,j} = \begin{cases} \alpha, & \text{if} \quad i = j, \\ \frac{1-\alpha}{n-1}, & \text{if} \quad i \neq j, \end{cases} \tag{2.15}$$

where $0 \leq \alpha \leq 1$ and $P_{i,j}$ denotes the transition probability from state i to state j.

Suppose we are given $\tau = 0$, i.e., privacy was ON at $t = 0$, and privacy is OFF at $t = 1$.

X_τ, X_t \ Q_t	A	B	AB	A	B	AB	A	B	AB
A, A	$\frac{\beta}{1-\alpha}$	0	$\frac{1-\alpha-\beta}{1-\alpha}$	$\frac{1-\alpha}{\beta}$	0	$\frac{\alpha+\beta-1}{\beta}$	1	0	0
A, B	0	1	0	0	1	0	0	$\frac{1-\beta}{\alpha}$	$\frac{\alpha+\beta-1}{\alpha}$
B, A	1	0	0	1	0	0	$\frac{1-\alpha}{\beta}$	0	$\frac{\alpha+\beta-1}{\beta}$
B, B	0	$\frac{\alpha}{1-\beta}$	$\frac{1-\alpha-\beta}{1-\beta}$	0	$\frac{1-\beta}{\alpha}$	$\frac{\alpha+\beta-1}{\alpha}$	0	1	0
	(a) $\alpha+\beta<1$			(b) $\alpha+\beta>1$ and t is even			(c) $\alpha+\beta>1$ and t is odd		

Table 2.3: The optimal ON-OFF privacy scheme that achieves the bound in (2.19) for $n = 2$. The query Q_t is probabilistic and depends on the current request X_t, the previous query Q_{t-1} and the last private request X_τ. The scheme consists of the following two cases: (i) if $Q_{t-1} = \{1, 2\}$, i.e., the previous query was for two messages, then the current query $Q_t = X_t$; (ii) if $Q_{t-1} \neq \{1, 2\}$, i.e., the previous query was for one message, then the current query Q_t is chosen based on the probabilities $p\left(q_t|x_\tau, x_t, q_{t-1}\right)$ given in this table. For (a) $\alpha + \beta < 1$, (b) and (c) are for $\alpha + \beta > 1$ where $t - \tau$ is even or odd respectively.

Following a direct application of (2.12) and (2.14) for $t = 1$, we have two regimes: $\alpha < \frac{1}{n}$ and $\alpha \geq \frac{1}{n}$. This is because the ordering of probabilities (c.f.(2.9)) changes at $\alpha = \frac{1}{n}$.

For $\alpha \geq \frac{1}{n}$, the bounds (2.12) and (2.14) match, i.e., the rate at $t = 1$ is achievable if and only if

$$\frac{1}{R_1} \geq n\alpha. \tag{2.16}$$

As for $\alpha < \frac{1}{n}$, we have that

$$\frac{1}{R_1^O} := \frac{n(1-\alpha)}{n-1} \leq \frac{1}{R_1} \leq 2 - n\alpha := \frac{1}{R_1^I}. \tag{2.17}$$

We illustrate (2.16) and (2.17) for a three state symmetric Markov chain, i.e., $n = 3$, with more details in Figure 2.3.

For the special case when there are $n = 2$ information sources, the outer bound (2.12) and inner bound (2.14), presented above, match. Therefore, the proposed scheme achieves the optimal rate for $n = 2$. We restate this result in Theorem 3, where the Markov chain has two states and is defined by the probability transition matrix

$$P = \begin{bmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{bmatrix}, \tag{2.18}$$

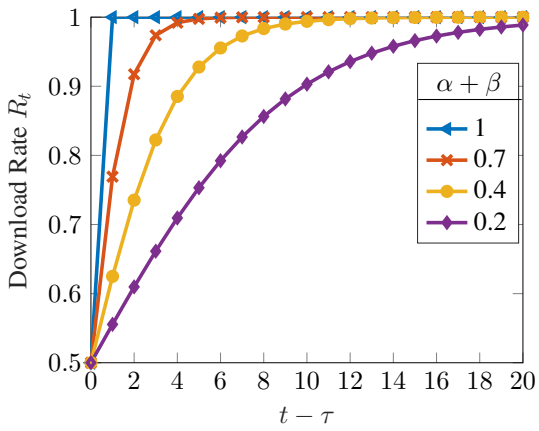


Figure 2.4: The maximum rate R_t, as given in Theorem 3, as a function of $t-\tau$ for different values of $\alpha+\beta$. As $\alpha+\beta$ approaches 1, the correlation between the requests decreases leading to an increase in the rate. For $\alpha+\beta=1$, the requests are independent. In this case, when privacy is ON at time t, which means $t-\tau=0$, the user has to download both messages, i.e., $R_t=1/2$. When privacy is OFF at time t , which means $t-\tau>0$, the user only downloads the desired message, i.e., $R_t=1$.

such that $0\leq\alpha,\beta\leq 1$.

Theorem 3. (Optimality for $n=2$) For $n=2$ information sources, the rate tuple $(R_t : t\in\mathbb{N})$ is achievable if and only if

$$\frac{1}{R_t}\geq 1+|1-\alpha-\beta|^{t-\tau}, \tag{2.19}$$

where $\tau=\max\{i : i\leq t, F_i=\text{ON}\}$.

Theorem 3 reflects the fact that, when the Markov chain is ergodic, the information carried by X_t about X_τ is decreasing exponentially as $t-\tau$ grows, so the user can eventually directly ask for the desired message at time t without being concerned about leaking information about X_τ. Table 2.3 gives an explicit scheme that achieves the rate in (2.19). The details of this construction will be further discussed in Section 2.7.2. Figure 2.4 shows the rate R_t as a function of time for different values of $\alpha+\beta$. As $\alpha+\beta$ approaches 1, the correlation between the request decreases leading to an increase in the rate.

2.4 Proof of the Outer Bound in Theorem 1

Recall that the inverse of the rate is expressed as

$$\frac{1}{R_t} = \frac{\ell_t}{L} = \frac{1}{L}\mathbb{E}\left[\ell(Q_t)\right]. \tag{2.20}$$

Hence, to obtain an upper bound on the rate R_t (a lower bound on $1/R_t$), we will derive a lower bound on the average downloading cost $\mathbb{E}[\ell(Q_t)]$ under the privacy and the decodability constraints.

First, we define an auxiliary random variable Y_t taking values in $\mathscr{P}(\mathcal{N})$ based on the decodability of the subset of messages. Specifically, let Y_t be a function of Q_t such that $Y_t = \mathcal{D}$ for $\mathcal{D} \in \mathscr{P}(\mathcal{N})$ if the answer A_t can decode the messages $W_{\mathcal{D}}$ but not any message W_i for $i \in \mathcal{N}\backslash\mathcal{D}$. Roughly speaking, Y_t represents the capability of decoding messages from the query Q_t. Note that since the query Q_t and messages $W_{\mathcal{N}}$ are independent, the decodability of any message is known by the server only through Q_t, that is, Y_t is a function of Q_t. In this way, the alphabet $\mathcal{Q}$ of the query is partitioned into 2^n classes based on the decodability of the subset of the messages. Clearly, from the definition of Y_t, we have

$$\ell(Q_t) \geq |Y_t|\, L, \tag{2.21}$$

and hence (2.20) can be written as

$$\frac{1}{R_t} \geq \mathbb{E}\left[|Y_t|\right]. \tag{2.22}$$

Thus, it remains for us to give a lower bound on $\mathbb{E}\left[|Y_t|\right]$ under the privacy and the decodability constraints.

Now, we start to interpret the privacy and the decodability constraints. By the definition of Y_t, the decodability can be rewritten as

$$p\left(x_t, y_t\right) = 0, \forall x_t \notin y_t. \tag{2.23}$$

Recall the privacy constraint that we require is $I\left(X_{\mathcal{B}_t}; Q_t | Q_{\lfloor t-1 \rfloor}\right) = 0$. Since

$$I\left(X_{\mathcal{B}_t}; Q_t | Q_{\lfloor t-1 \rfloor}\right) \geq I\left(X_\tau; Q_t | Q_{\lfloor t-1 \rfloor}\right) \geq I\left(X_\tau; Y_t | Q_{\lfloor t-1 \rfloor}\right),$$

we can relax the privacy constraint by

$$I\left(X_\tau; Y_t | Q_{\lfloor t-1 \rfloor}\right) = 0. \tag{2.24}$$

Therefore, to obtain an upper bound on the rate R_t (a lower bound on $1/R_t$), it remains for us to give a lower bound on $\mathbb{E}\left[|Y_t|\right]$ such that (2.23) and (2.24) are satisfied, which relies on the following lemma. The proof of the lemma can be found in Appendix A.3.

Lemma 1. For any random variables U, X and Y, taking values in the alphabet $\mathcal{N}$, $\mathcal{N}$ and $\mathscr{P}\left(\mathcal{N}\right)$ respectively, if Y is independent of U, and $p(x, y) = 0$ for $x \notin y$, then

$$\mathbb{E}\left[|Y|\right] \geq \sum_{x \in \mathcal{N}} \max_{u \in \mathcal{N}} p\left(x|u\right). \tag{2.25}$$

For any given $q_{\lfloor t-1 \rfloor}$, we can see that Lemma 1 immediately gives a lower bound on $\mathbb{E}\left[|Y_t||q_{\lfloor t-1 \rfloor}\right]$, i.e.,

$$\mathbb{E}\left[|Y_t||q_{\lfloor t-1 \rfloor}\right] \geq \sum_{x_t} \max_{x_\tau} p\left(x_t|x_\tau, q_{\lfloor t-1 \rfloor}\right). \tag{2.26}$$

Thus, by summing over all $q_{\lfloor t-1 \rfloor}$, we can obtain that

$$\mathbb{E}\left[|Y_t|\right] = \sum_{q_{\lfloor t-1 \rfloor}} p\left(q_{\lfloor t-1 \rfloor}\right) \mathbb{E}\left[|Y_t||q_{\lfloor t-1 \rfloor}\right] \geq \sum_{q_{\lfloor t-1 \rfloor}} p\left(q_{\lfloor t-1 \rfloor}\right) \sum_{x_t} \max_{x_\tau} p\left(x_t|x_\tau, q_{\lfloor t-1 \rfloor}\right). \tag{2.27}$$

By substituting (2.27) in (2.20), we finally get

$$\frac{1}{R_t} \geq \sum_{q_{\lfloor t-1 \rfloor}} p\left(q_{\lfloor t-1 \rfloor}\right) \sum_{x_t} \max_{x_\tau} p\left(x_t|x_\tau, q_{\lfloor t-1 \rfloor}\right),$$

which completes the proof.

2.5 Inner Bound in Theorem 2

Before we move on to describe the achievable scheme, we present an example for $n = 3$ sources, which illustrates the basic idea of the scheme that achieves the bound in (2.14).

2.5.1 Example of an Achievable Scheme

x_0	x_1 \ q_1	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2.3\}$	$\{1,2,3\}$	Budget $(P_{i,j})$
1	1	0.1	0	0	0	0	0	0	0.1
	2	0	0.3	0	0	0	0	0	0.3
	3	0	0	0.1	0	$0.1+0.2$	0.1	0.1	0.6
2	1	0.1	0	0	0	$0.1+0.2$	0	0.1	0.5
	2	0	0.3	0	0	0	0.1	0	0.4
	3	0	0	0.1	0	0	0	0	0.1
3	1	0.1	0	0	0	0.1	0	0	0.2
	2	0	0.3	0	0	0	0.1	0.1	0.5
	3	0	0	0.1	0	0.2	0	0	0.3

Table 2.4: The constructed distribution $p(q_1, x_1|x_0)$ for the given $p(x_1|x_0)$ in Example 2.5.1.

Suppose the transition probabilities of the Markov chain are given by

$$P = \begin{bmatrix} 0.1 & 0.3 & 0.6 \\ 0.5 & 0.4 & 0.1 \\ 0.2 & 0.5 & 0.3 \end{bmatrix}, \tag{2.28}$$

where $P_{i,j} = \Pr(X_t = j | X_{t-1} = i)$.

Assume that privacy is ON at time $t = 0$ and privacy is OFF at time $t = 1$. At time $t = 0$, we know the user has to send the query $Q_0 = \{1, 2, 3\}$. Our goal is to design the query Q_1 at $t = 1$. In particular, in this example, the query Q_1 is uncoded and is a probabilistic function of the previous request X_0, the current request X_1 and the previous query Q_0. As such, we will show how to design the query encoding function $p(q_1|x_1, x_0)$[2], or equivalently $p(q_1, x_1|x_0)$, for all $x_0, x_1 \in \{1, 2, 3\}$ and $q_1 \in \mathscr{P}(\{1, 2, 3\})$. The distribution $p(q_1, x_1|x_0)$ is represented in Table 2.4. Throughout this example, we will show how to fill in the values of the cells in Table 2.4.

As requested, the query Q_1 should satisfy the decodability and the privacy constraints. The two constraints can be translated into the following rules for filling Table 2.4.

1. Satisfying the decodability constraint is straightforward. We set $p(q_1, x_1|x_0) = 0$ for all $x_1 \notin q_1$, i.e., setting all the gray highlighted cells in Table 2.4 to zero. This guarantees that

[2]We drop q_0 in $p(q_1|x_1, x_0, q_0)$ since $q_0 = \{1, 2, 3\}$ is a constant.

the user always receives messages containing the one he wants when the server responds to his query.

2. The privacy constraint requires that Q_1 is independent of X_0, i.e.,

$$p\left(Q_1 = q_1 | X_0 = 1\right) = p\left(Q_1 = q_1 | X_0 = 2\right) = p\left(Q_1 = q_1 | X_0 = 3\right),$$

for all $q_1 \in \mathscr{P}\left(\{1,2,3\}\right)$. By the law of total probability, this can be written as

$$\sum_{x_1} p\left(q_1, x_1 | X_0 = 1\right) = \sum_{x_1} p\left(q_1, x_1 | X_0 = 2\right) = \sum_{x_1} p\left(q_1, x_1 | X_0 = 3\right).$$

To translate this in Table 2.4, each column is divided into 3 blocks (pertaining to $x_0 \in \{1,2,3\}$), and the sum of the cells in each block in a given column are to be equal, e.g., in column $\{1,3\}$ each block sum to 0.3.

3. Since the entries are probabilities, this requires the sum of row j in a given block i to be equal to $p\left(X_1 = j | X_0 = i\right)$, i.e., $P_{i,j}$ in the matrix P. We will refer to $P_{i,j}$ as our *budget* for row j in block i, it is highlighted in blue in Table 2.4.

We now introduce an ordering of probabilities, such that

$$\Pr\left(X_1 = j | X_0 = x_0^{(j,1)}\right) \leq \Pr\left(X_1 = j | X_0 = x_0^{(j,2)}\right) \leq \Pr\left(X_1 = j | X_0 = x_0^{(j,3)}\right)$$

for each $j \in \{1,2,3\}$. For example, for $X_1 = 1$, we observe that $P_{1,1} \leq P_{1,3} \leq P_{1,2}$, so $x_0^{(1,1)} = 1$, $x_0^{(1,2)} = 3$, and $x_0^{(1,3)} = 2$. We summarize the values of the rest of the variables in the Table 2.5.

It is worth noting that downloading all messages is always a feasible solution here. More precisely, setting the probability of querying three messages to be equal to the budget, i.e.,

$$p\left(Q_1 = \{1,2,3\}, X_1 = x_1 | X_0 = x_0\right) = p\left(X_1 = x_1 | X_0 = x_0\right)$$

for all $x_0, x_1 \in \{1,2,3\}$, always satisfies rules one-three. Next, we present the algorithm that better fills the table and satisfies the aforementioned rules. The main idea is to assign values as large as possible to Q_1 with small cardinality, and this will ultimately lower the communication cost.

- **Step 1:** We start with queries q_1 of cardinality one, i.e., $|q_1| = 1$. We adopt a greedy-like approach, which means we try to maximize the value filled in the first three columns. Due to the second and third rules mentioned above, the maximum values we can choose are

$$p\left(Q_1 = \{x_1\}, X_1 = x_1 | X_0 = x_0\right) = \min_{x_0} p\left(x_1 | x_0\right) = p\left(x_1 | x_0^{(x_1,1)}\right) = \begin{cases} 0.10, & x_1 = 1, \\ 0.30, & x_1 = 2, \\ 0.10, & x_1 = 3. \end{cases} \tag{2.29}$$

  Note that in some rows the rest of the cells, e.g., row 1 in block 1, have to be zero, because from rule 3 we know that their budget has been consumed.

- **Step 2:** When $|q_1| = 2$, the construction is more complicated because each block has two cells to fill. We describe it as follows.

  - For $X_1 = 1$, we know that $x_0^{(1,1)} = 1$ and $x_0^{(1,2)} = 3$. Since, in Step 1 (2.29), we consumed the probability $p\left(X_1 = 1 | X_0 = x_0^{(1,1)}\right)$, we deduct it from the the second minimal value $p\left(X_1 = 1 | X_0 = x_0^{(1,2)}\right)$, and calculate

$$p\left(X_1 = 1 | X_0 = x_0^{(1,2)}\right) - p\left(X_1 = 1 | X_0 = x_0^{(1,1)}\right) = 0.1.$$

    Then, we may find some $\hat{q}$ (to be determined), such that $|\hat{q}| = 2$ and $1 \in \hat{q}$ and set

$$p\left(Q_1 = \hat{q}, X_1 = x_1 | X_0 = x_0\right) = \begin{cases} 0.1, & x_1 = 1 \wedge x_0 \neq x_0^{(1,1)} \\ & \text{or } x_1 = \hat{q} \backslash \{1\} \wedge x_0 = x_0^{(1,1)}, \\ 0, & \text{others.} \end{cases} \tag{2.30}$$

    Here, we have two options for $\hat{q}$, either $\{1, 2\}$ or $\{1, 3\}$. If $\hat{q} = \{1, 2\}$, from rule 2, we know that the summation of each block must be the same. However, if we inspect first block i.e., $X_0 = 1$, we can find that the budget for the first two rows of the first block is zero, which means that we do not have enough budgets to assign values according to

$x_0^{(j,i)}$	$j=1$	$j=2$	$j=3$	λ_i	θ_i
$x_0^{(j,1)}$	1	1	2	0.5	0.5
$x_0^{(j,2)}$	3	2	3	0.9	0.4
$x_0^{(j,3)}$	2	3	1	1.6	0.1

Table 2.5: Useful Variables for Example 2.5.1.

(2.30). Therefore, if we choose $\hat{q} = \{1, 2\}$, then it will violate rule 2, so that $\hat{q}$ is chosen to be $\{1, 3\}$, and fill in the table according to (2.30).

- For $X_1 = 2$ the procedure is the same as we did for $X_1 = 1$ and details are omitted.
- For $X_1 = 3$, we know that $x_0^{(3,1)} = 2$ and $x_0^{(3,2)} = 3$. Also, we have

$$p\left(X_1 = 3 | X_0 = x_0^{(3,2)}\right) - p\left(X_1 = 3 | X_0 = x_0^{(3,1)}\right) = 0.2.$$

Then, we follow the same procedure as above by determining $\hat{q} = \{1, 3\}$. However, since we have assigned a value 0.1 to the cell

$$p\left(Q_1 = \{1, 3\}, X_1 = 3 | X_0 = 1\right)$$

in previous steps, we augment its value by 0.2, and finally we have

$$p\left(Q_1 = \{1, 3\}, X_1 = 3 | X_0 = 1\right) = 0.1 + 0.2 = 0.3.$$

- **Step 3:** When $|q_1| = 3$, since this is the last column, we just need to complete the table such that the budget of all rows is fully consumed.

Finally, let us evaluate the achievable rate R_1, equivalently $1/\mathbb{E}[|Q_1|]$, achieved by the constructed $p\left(q_1, x_1 | x_0\right)$. It is easy to see that we assign $\theta_1 = \lambda_1 = 0.5$ to cells such that $|q_1| = 1$, $\theta_2 = \lambda_2 - \lambda_1 = 0.4$ to cells such that $|q_1| = 2$, and $\theta_3 = 1 - \lambda_2 = 1 - 0.9 = 0.1$ to cells such that $|q_1| = 3$ for each block, so that we have

$$\mathbb{E}[|Q_1|] = \sum_{i=1}^{3} i\,\theta_i = 1.6,$$

where λ_i and θ_i are defined in (2.10) and (2.11) respectively. Thus, $R_1 = 5/8$ is achievable in this example. One may notice that the outer bound in Corollary 1 gives

$$\frac{1}{R_1} \geq \sum_{x_1} \max_{x_0} p\left(x_1|x_0\right) = 0.5 + 0.5 + 0.6 = 1.6,$$

which indicates that $R_1 = 5/8$ is optimal for this example. However, we would like to mention that this example is special because it shows an instance where the bounds (2.12) and (2.14) match. In general, for a choice of transition probabilities different from those given in (2.28), there might be a gap, as illustrated in Example 1.

2.5.2 Proof of Theorem 2

We will build on the previous example to describe the generalized scheme achieving the rate given in (2.14). The proposed coding scheme retrieves messages in the uncoded form, so we assume that $\mathcal{Q} = \mathscr{P}\left(\mathcal{N}\right)$ in the remaining parts of this section.

Answer encoding function: The answer encoding function ρ_t is given by

$$A_t = \rho_t\left(Q_t, W_{\mathcal{N}}\right) = W_{\mathcal{A}} \tag{2.31}$$

for any $Q_t = \mathcal{A} \in \mathscr{P}\left(\mathcal{N}\right)$.

Answer length function: The length of the answer is given by $\ell\left(Q_t\right) = \left|Q_t\right| L$, and the normalized average length is then given by

$$\frac{1}{R_t} = \frac{\ell_t}{L} = \mathbb{E}\left[\left|Q_t\right|\right]. \tag{2.32}$$

Query encoding function: At time t, suppose that the query Q_t is a stochastic function of X_t, X_τ and $Q_{\lfloor t-1 \rfloor}$. Recall that $\tau = \max \mathcal{B}_t$, i.e., the last time privacy was ON. For any given $q_{\lfloor t-1 \rfloor}$, we claim that there exists an encoding function $w\left(q_t|x_\tau, x_t, q_{\lfloor t-1 \rfloor}\right)$ giving

$$\mathbb{E}\left[\left|Q_t\right|\left|q_{\lfloor t-1 \rfloor}\right.\right] \leq \sum_{i=1}^{n} i\,\theta_i(q_{\lfloor t-1 \rfloor}), \tag{2.33}$$

as well as satisfying two constraints, i.e.,

$$p\left(x_t, q_t | q_{\lfloor t-1 \rfloor}\right) = 0, \forall x_t \notin q_t, \tag{2.34}$$

and

$$I(Q_t; X_\tau | Q_{\lfloor t-1 \rfloor} = q_{\lfloor t-1 \rfloor}) = 0, \tag{2.35}$$

Note that (2.34) guarantees the decodability from the answer encoding function ρ_t given by (2.31), and (2.35) is a relaxed privacy constraint, where we recall the original privacy constraint

$$I\left(Q_t; X_{\mathcal{B}_t} | Q_{\lfloor t-1 \rfloor}\right) = 0.$$

The following lemma justifies the existence of such a query encoding function.

Lemma 2. For any given random variables $U, X \in \mathcal{N}$, suppose that

$$p\left(X = x | U = u^{(x,1)}\right) \leq p\left(X = x | U = u^{(x,2)}\right) \leq \cdots \leq p\left(X = x | U = u^{(x,n)}\right). \tag{2.36}$$

Then, there exists a random variable $Y \in \mathscr{P}(\mathcal{N})$ such that Y is independent of U, $p(x, y) = 0$ for $x \notin y$, and

$$\mathbb{E}\left[|Y|\right] \leq \sum_{i=1}^{n} i\, \theta_i, \tag{2.37}$$

where $\theta_i = \min\left\{1, \sum\limits_{x \in \mathcal{N}} p\left(X = x | U = u^{(x,i)}\right)\right\} - \min\left\{1, \sum\limits_{x \in \mathcal{N}} p\left(X = x | U = u^{(x,i-1)}\right)\right\}$ for $i = 1, \ldots, n$.

Proof. We prove Lemma 2 by designing a distribution $p\left(y, x | u\right)$ for any given distribution $p\left(x | u\right)$ satisfying the constraints $Y \perp U$, $p(x, y) = 0$ for $x \notin y$, and

$$p\left(|Y| \leq i\right) \geq \sum_{j=1}^{i} \theta_j,\ i = 1, \ldots, n.$$

Moreover, we show that $\mathbb{E}\left[|Y|\right] \leq \sum_{i=1}^{n} i\, \theta_i$ follows from the last constraint. The proof of Lemma 2 is constructive, i.e., we provide an algorithm that outputs the desired distribution. The details of the construction will be presented at the end of this section, and the justification of the algorithm and

analysis of its complexity will be deferred to Appendix A.4. □

Before the detailed proof, we give the following reflections on the lemma.

1. This lemma generalizes the process we used to fill Table 2.4 for $n = 3$ in Subsection 2.5.1. However, one may notice that the table therein contains about $n^2\, 2^n$ entries, so any linear time approach such as filling them one by one will introduce an exponential blowup in complexity. Hence, the proof of the lemma not only justifies the existence of an admissible $p\,(y, x|u)$, but also proposes a poly(n) time algorithm to construct a $p\,(y, x|u)$ for any given distribution $p\,(x|u)$ to satisfy the constraints.

2. If we treat each probability $p\,(y, x|u)$ for $x, u \in \mathcal{N}$ and $y \in \mathscr{P}\,(n)$ as a decision variable, we can see that both the objective function $\mathbb{E}[|Y|]$ and two constraints, i.e., Y is independent of U and $p(x, y) = 0$ for $x \notin y$, are linear, and hence the problem can be indeed formulated as a linear programming problem with roughly $n^2\, 2^n$ variables and constraints, which makes the numerical solution impossible when n goes large. The lemma here is aimed at finding a solution efficiently (avoid exponential overhead) and analytically (evaluate the objective value). More interpretations on this linear programming perspective will be discussed in Section 2.6.

For any given $q_{\lfloor t-1 \rfloor}$, by letting $U \sim p_{X_\tau|q_{\lfloor t-1 \rfloor}}$ and $X \sim p_{X_t|q_{\lfloor t-1 \rfloor}}$ in Lemma 2, this lemma justifies the existence of a query encoding function $w\left(q_t|x_\tau, x_t, q_{\lfloor t-1 \rfloor}\right)$ satisfying (2.33), (2.34) and (2.35). It remains to show that the relaxed privacy constraint (2.35) implies the desired privacy constraint (2.6) for the given scheme, i.e., $I(Q_t; X_\tau|Q_{\lfloor t-1 \rfloor}) = 0$ implies $I\left(Q_t; X_{\mathcal{B}_t}|Q_{\lfloor t-1 \rfloor}\right) = 0$, which can be justified by Proposition 1. Therefore, we finish showing that for any given $q_{\lfloor t-1 \rfloor}$, there exists an encoding function $w\left(q_t|x_\tau, x_t, q_{\lfloor t-1 \rfloor}\right)$ satisfying the decodability and the privacy constraint. Also, we know from Lemma 2 that the encoding function $w\left(q_t|x_\tau, x_t, q_{\lfloor t-1 \rfloor}\right)$ yields

$$\mathbb{E}\left[|Q_t||q_{\lfloor t-1 \rfloor}\right] \leq \sum_{i=1}^{n} i\, \theta_i(q_{\lfloor t-1 \rfloor}).$$

By averaging over all $q_{\lfloor t-1 \rfloor}$, we have

$$\mathbb{E}\left[|Q_t|\right] \leq \sum_{q_{\lfloor t-1 \rfloor}} p(q_{\lfloor t-1 \rfloor}) \sum_{i=1}^{n} i\, \theta_i(q_{\lfloor t-1 \rfloor}), \tag{2.38}$$

which implies that R_t is achievable (c.f.(2.32)) if

$$\frac{1}{R_t} \geq \sum_{q_{\lfloor t-1 \rfloor}} p(q_{\lfloor t-1 \rfloor}) \sum_{i=1}^{n} i\, \theta_i(q_{\lfloor t-1 \rfloor}). \tag{2.39}$$

2.5.3 Constructive Proof of Lemma 2

First, let us recall some definitions and notation which will be used frequently in this section. For a fixed $x \in \mathcal{N}$, suppose that

$$p\left(X = x|U = u^{(x,1)}\right) \leq p\left(X = x|U = u^{(x,2)}\right) \leq \cdots \leq p\left(X = x|U = u^{(x,n)}\right), \tag{2.40}$$

where $u^{(x,i)}$ for $i = 1, \ldots, n$ are n distinct elements in $\mathcal{N}$. Let

$$\lambda_i = \sum_{x \in \mathcal{N}} p\left(X = x|U = u^{(x,i)}\right), \tag{2.41}$$

and

$$\theta_i = \min\{1, \lambda_i\} - \min\{1, \lambda_{i-1}\}, \tag{2.42}$$

where λ_0 is assumed to be 0. Note that $\sum_{i=1}^{n} \theta_i = 1$. Also, let

$$\sigma = \max\{i : \lambda_i \leq 1\}. \tag{2.43}$$

In this section, we will prove Lemma 2 by designing a distribution $p\,(y, x|u)$ for any given distribution $p\,(x|u)$ satisfying the constraints $Y \perp U$, $p(x, y) = 0$ for $x \notin y$, and

$$p\,(|Y| \leq i) \geq \sum_{j=1}^{i} \theta_j,\ i = 1, \ldots, n. \tag{2.44}$$

One can check that (2.44) yields

$$\begin{aligned}\mathbb{E}\left[|Y|\right] &= \sum_{i=1}^{n} i\, p\left(|Y|=i\right) = \sum_{i=1}^{n}\sum_{j=1}^{i} p\left(|Y|=i\right) \\ &= \sum_{j=1}^{n}\sum_{i=j}^{n} p\left(|Y|=i\right) = \sum_{j=1}^{n} p\left(|Y|\geq j\right) \\ &= \sum_{j=1}^{n}\left(1-p\left(|Y|\leq j-1\right)\right) \\ &\leq \sum_{j=1}^{n}\left(1-\sum_{i=1}^{j-1}\theta_i\right) = \sum_{j=1}^{n}\sum_{i=j}^{n}\theta_i \\ &= \sum_{i=1}^{n}\sum_{j=1}^{i}\theta_i = \sum_{i=1}^{n} i\,\theta_i,\end{aligned}$$

i.e., (2.37) to be proved in Lemma 2.

In particular, let Z be a multiset $(\mathcal{N}, m)$, where $\mathcal{N}$ is the ground set and m is the multiplicity function. The cardinality of the multiset Z is the summation of multiplicities of all its element, i.e.,

$$|Z| = \sum_{x\in n} m(x). \tag{2.45}$$

For example, given the ground set $\{a, b\}$ and the multiset $\{a, a, b\}$, the multiplicities of a and b are $m(a) = 2$ and $m(b) = 1$ respectively, and the cardinality of $|\{a, a, b\}|$ is 3. For ease of notation, denote $\mathcal{Z} = \{Z : Z \in (\mathcal{N}, m), |Z| \leq n\}$, i.e., the multiset whose elements are chosen from $\mathcal{N}$ and whose cardinality is upper bounded by n.

We will prove that for any given X and U, i.e., given any distribution $p\left(x|u\right)$ for $x, u \in \mathcal{N}$, there exists a random variable Z taking values in $\mathcal{Z}$ such that $Z \perp U$, $p(x, z) = 0$ for $x \notin z$, and

$$p\left(|Z|=i\right) = \theta_i,\ \forall i = 1, \ldots, \sigma+1, \tag{2.46}$$

Note that $\theta_i = 0$ for $i > \sigma + 1$ from the definition (2.42). By letting $Y = \text{Set}(Z)$, i.e., Y is the corresponding set of the multiset Z, we can easily see that if $Z \perp U$ and $p(x, z) = 0$ for $x \notin z$, then $Y \perp U$ and $p(x, y) = 0$ for $x \notin y$. Also, one can easily check that if (2.46) is satisfied, then (2.44) holds. Therefore, it is sufficient for us to justify the existence of such a Z for any given X and U.

Now, we start the constructive proof, i.e., for any given distribution $p(x|u)$, we will give an algorithm to construct some Z satisfying that

$$p(z,x)=0,\ \forall x\notin z, \tag{2.47}$$

and

$$p(z|u)=p\left(z|u'\right),\ \forall z\in\mathcal{Z} \text{ and } u,u'\in\mathcal{N}. \tag{2.48}$$

Finally, we will show that the constructed Z gives (2.46), i.e.,

$$p(|Z|=i)=\theta_i,\ \forall i=1,\ldots,\sigma+1.$$

Input: A distribution $p(x|u)$ for $x,u\in\mathcal{N}$.

Pre-calculation:

1. For any given distribution $p(x|u)$, by sorting $p(x|u)$ for each $x\in\mathcal{N}$, we can easily obtain parameters
$$\left\{u^{(x,i)},\lambda_i,\theta_i,\sigma : x\in\mathcal{N},i=1,\ldots,n\right\}$$
as defined in (2.40)-(2.43). We will refer to these notations directly in the sequel.

2. Then, we randomly pick a set of real numbers $\{\delta_j : j=1,\ldots,n\}$ such that
$$p\left(X=j|U=u^{(j,\sigma)}\right)\le\delta_j\le p\left(X=j|U=u^{(j,\sigma+1)}\right),\ \forall j\in\mathcal{N}, \tag{2.49}$$
and
$$\sum_{j=1}^{n}\delta_j=1. \tag{2.50}$$
The existence of such a set of $\{\delta_j : j=1,\ldots,n\}$ can be guaranteed by the definition of σ, since
$$\begin{aligned}\lambda_\sigma&=\sum_{j=1}^{n}p\left(X=j|U=u^{(j,\sigma)}\right)\\&\le\sum_{j=1}^{n}\delta_j\le\sum_{j=1}^{n}p\left(X=j|U=u^{(j,\sigma+1)}\right)=\lambda_{\sigma+1},\end{aligned}$$

and $\lambda_\sigma \leq 1 < \lambda_{\sigma+1}$.

Specification: Here we specify a deterministic way of picking δ_j for $j = 1, \ldots, n$. For notational simplicity, let $a_j = p\left(X = j | U = u^{(j,\sigma)}\right)$ and $b_j = p\left(X = j | U = u^{(j,\sigma+1)}\right)$ for $j = 1, \ldots, n$. Then, provided two non-negative arrays $(a_1, \ldots, a_n)$ and $(b_1, \ldots, b_n)$ such that

$$\sum_{j=1}^{n} a_j \leq 1 < \sum_{j=1}^{n} b_j,$$

our goal is to output an array $(\delta_1, \ldots, \delta_n)$ such that

$$a_j \leq \delta_j \leq b_j, \ \forall j = 1, \ldots, n,$$

and

$$\sum_{j=1}^{n} \delta_j = 1.$$

We may choose δ_j sequentially and greedily. In particular, initialize $T = 0$. For $j = 1, \ldots, n$, update T by $T + (b_j - a_j)$. If

$$T \leq 1 - \sum_{j=1}^{n} a_j,$$

then let $\delta_j = b_j$, otherwise let

$$\delta_j = 1 - \sum_{k=1}^{j-1} b_j - \sum_{k=j+1}^{n} a_j$$

and $\delta_k = a_k$ for $k = j + 1, \ldots, n$ to finish the process.

Let Q be an auxiliary $n \times n$ matrix which will be updated during the algorithm. Also, let $Q_{i,j}^{-} = a$ denote $Q_{i,j} = Q_{i,j} - a$, i.e., subtracting a from $Q_{i,j}$.

Initialization: Let

$$Q_{i,j} = \max\left\{p\left(X = j | U = i\right) - \delta_j, 0\right\}, \ i, j \in \mathcal{N}. \tag{2.51}$$

Procedure: For $|Z| = \ell = 1, \ldots, \sigma + 1$, we consider the following process. For $x = 1, \ldots, n$, identify $\{u^{(x,i)} : i = 1, \ldots, \ell - 1\}$.

1. For each $u^{(x,i)}$, we randomly choose a collection of pairs

$$I_i \times V_i = \{(x_{i,j}, v_{i,j}) : j = 1, 2, \ldots\} \tag{2.52}$$

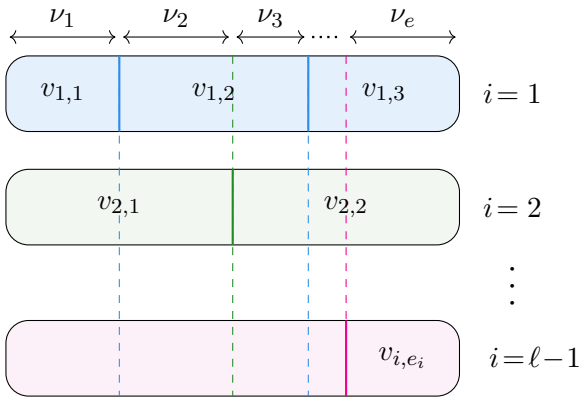


Figure 2.5: The rows represents $V_1, \ldots, V_{\ell-1}$. A given row i is divided, by *boundaries*, into e_i parts of different sizes, corresponding to $v_{i,1}, \ldots v_{i,e_i}$, e.g., V_1 is divided into $v_{1,1}$, $v_{1,2}$, and $v_{1,3}$. Moreover, rows are the same size in total to satisfy (2.55). Then, every ν_k represents the number between two consecutive *boundaries*.

such that

$$0 \leq v_{i,j} \leq Q_{u^{(x,i)},x_{i,j}}, \tag{2.53}$$

and

$$\sum_j v_{i,j} = \min\left\{\delta_x, p\left(X = x|U = u^{(x,\ell)}\right)\right\} - p\left(X = x|U = u^{(x,\ell-1)}\right). \tag{2.54}$$

Note that the right-hand side of (2.54) only depends on ℓ and x and is independent of $u^{(x,i)}$, which means that

$$\sum_{j=1} v_{1,j} = \cdots = \sum_{j=1} v_{\ell-1,j}, \tag{2.55}$$

though the cardinality of V_i for each i may or may not be the same. For ease of notation, suppose that

$$|I_i| = |V_i| = e_i \leq n.$$

After that, we update the matrix Q by

$$Q^{-}_{u^{(x,i)},x_{i,j}} = v_{i,j}. \tag{2.56}$$

It is clear from (2.53) and (2.56) that Q is always non-negative, so the existence of such a

collection $I_i \times V_i$ can be guaranteed if the following condition is satisfied

$$\sum_{k=1}^{n} Q_{u^{(x,i)},k} \geq \min\left\{\delta_x, p\left(X = x|U = u^{(x,\ell)}\right)\right\} - p\left(X = x|U = u^{(x,\ell-1)}\right), \quad (2.57)$$

which will be verified in Appendix A.4.

Specification: We specify a deterministic way of choosing $I_i \times V_i$ under the assumption that (2.57) holds. If the right-hand side of (2.54) is zero, then one can simply choose $I_i \times V_i$ to be the empty set. If the right-hand side of (2.54) is strictly positive, we initialize $T = 0$ and $j = 1$. Then for $k = 1, \ldots, n$ such that $Q_{u^{(x,i)},k} > 0$, if

$$T + Q_{u^{(x,i)},k} < \text{R.H.S of (2.54)}, \quad (2.58)$$

let $v_{i,j} = Q_{u^{(x,i)},k}$, $x_{i,j} = k$. Then increae j by one and update T by adding $Q_{u^{(x,i)},k}$ to it. Otherwise, let

$$v_{i,j} = \text{R.H.S of (2.54)} - T$$

and $x_{i,j} = k$ to finish the process.

2. For fixed ℓ and x, given I_i and V_i for $i = 1, \ldots, \ell - 1$, we randomly pick a collection of pairs $\{(\zeta_k, \nu_k) : k = 1, 2, \ldots\}$ such that

$$\zeta_k \in I_1 \times I_2 \times \cdots \times I_{\ell-1}, \quad (2.59)$$

and

$$\sum_{k:\zeta_k(i)=x_{i,j}} \nu_k = v_{i,j}, \ \forall 1 \leq i \leq \ell - 1 \text{ and } 1 \leq j \leq e_i, \quad (2.60)$$

where $\zeta_k(i)$ is the i-th element of ζ_k. The existence of such a collection can be basically illustrated by Figure 2.5. For notational simplicity, denote

$$|\{(\zeta_k, \nu_k) : k = 1, 2, \ldots\}| = e_{x,\ell}.$$

Specification: We specify a deterministic way to construct such a collection

$$\{(\zeta_k, \nu_k) : k = 1, 2, \ldots, e\}.$$

Let us initially push $(v_{1,1}, v_{2,1}, \ldots, v_{\ell-1,1})$ and $(x_{1,1}, x_{2,1}, \ldots, x_{\ell-1,1})$ into buffers B_v and B_x, respectively. Let $\nu_1 = \min \mathrm{B_v}$ and $\zeta_1 = \mathrm{B}_x$. Assume that the minimal value of B_v appears in the m-th position for some $m \in \{1, \ldots, \ell - 1\}$, i.e., $v_{m,1}$ is the minimal. If the minimal is not unique, just randomly choose one. We update B_v by subtracting $v_{m,1}$ from all elements in B_v and then push $v_{m,2}$ into the buffer to replace $v_{m,1} - v_{m,1}$, i.e.,

$$\mathrm{B}_v = (v_{1,1} - v_{m,1}, \ldots, v_{m,2}, \ldots, v_{\ell-1,1} - v_{m,1}).$$

Also, update B_x by letting $\mathrm{B}_x = (x_{1,1}, \ldots, x_{m,2}, \ldots, x_{\ell-1,1})$.

Then, let $\nu_2 = \min \mathrm{B}_v$ and $\zeta_2 = \mathrm{B}_x$, and update B_v and B_x by the same process as stated above. Keep doing this repeatedly until all values $v_{i,j}$ for $1 \leq i \leq \ell - 1$ and $1 \leq j \leq e_i$ have been dealt with. Note that (2.55) guarantees that the process ends properly. In this process, we deal with one $v_{i,j}$ every round, so we have

$$e_{x,\ell} = \sum_{i=1}^{\ell-1} e_i \leq (\ell - 1)n. \tag{2.61}$$

3. For each $k = 1, \ldots, e_{x,\ell}$, let $z_k = \{\zeta_k, x\}$. Then we let $\mathcal{A}_{k,x,\ell}$ be a collection of tuples defined as follows:

$$\begin{aligned}\mathcal{A}_{k,x,\ell} = &\Big\{ (\bar{z}, \bar{x}, \bar{u}) : \bar{z} = z_k, \bar{x} = \zeta_k(i), \bar{u} = u^{(x,i)}, i = 1, \ldots, \ell - 1 \Big\} \\ &\bigcup \Big\{ (\bar{z}, \bar{x}, \bar{u}) : \bar{z} = z_k, \bar{x} = x, \bar{u} \in \mathcal{N} \setminus \big\{ u^{(x,i)} : i = 1, \ldots, \ell - 1 \big\} \Big\}, \end{aligned} \tag{2.62}$$

where $|\mathcal{A}_{k,x,\ell}| = n$. To avoid ambiguity in the following discussion, denote

$$\nu_{k,x,\ell} = \nu_k. \tag{2.63}$$

4. For a fixed ℓ, denote

$$\mathcal{A}_\ell = \bigcup_{1\le x\le n} \bigcup_{1\le k\le e_{x,\ell}} \mathcal{A}_{k,x,\ell}, \tag{2.64}$$

and for any $(\bar{z},\bar{x},\bar{u}) \in \mathcal{A}_\ell$, let

$$q\left(\bar{z},\bar{x},\bar{u}\right) = \sum_{x=1}^{n} \sum_{k:(\bar{z},\bar{x},\bar{u})\in\mathcal{A}_{k,x,\ell}} \nu_{k,x,\ell}, \tag{2.65}$$

where $k = 1,\ldots,e_{x,\ell}$.

Output: The output of the algorithm is $\mathrm{OUT} = \{\mathcal{A}_\ell, q\left(\mathcal{A}_\ell\right) : \ell = 1,\ldots,\sigma+1\}$. Later, we will see that this pair indeed stores the non-zero valued arguments and corresponding values of $p\left(z,x|u\right)$, i.e.,

$$p\left(z,x|u\right) = \begin{cases} q\left(z,x,u\right), & (z,x,u)\in\mathcal{A}, \\ 0, & \text{otherwise}, \end{cases} \tag{2.66}$$

where $\mathcal{A} := \cup_\ell \mathcal{A}_\ell$. Note that $\mathcal{A}_\ell$ are disjoint with each other since $|z| = \ell$ for any $(z,x,u) \in \mathcal{A}_\ell$ from (2.59).

For the better illustration, we summarize the constructive proof in Algorithm 1.

Algorithm 1

Input: A given distribution $p\left(x|u\right)$ for $x,u \in \mathcal{N}$

Output: The non-zero valued arguments $\mathcal{A} = \{(z,x,u) : x,u\in\mathcal{N}, z\in\mathcal{Z}, p\left(z,x|u\right) > 0\}$ and probabilities $q\left(\mathcal{A}\right) = \{p\left(z,x|u\right) : (z,x,u)\in\mathcal{A}\}$ for a distribution $p\left(z,x|u\right)$ such that $p\left(z|u\right) = p\left(z\right)$ and $p\left(z,x|u\right) = 0$ for any $x \notin z$

Pre-calculation
Initialize
for $\ell = 1,\ldots,\sigma+1$ **do**
  for $x \in \mathcal{N}$ **do**
    for $u \in \{u^{(x,i)} : i = 1,\ldots,\ell-1\}$ **do**
      Find a collection of pairs $I_i \times V_i$ satisfying (2.53) and (2.54)
    end for
    Given $\{I_i \times V_i : i = 1,\ldots,\ell-1\}$, find a collection $\{(\zeta_k,\nu_k) : k = 1,2,\ldots,e_{x,\ell}\}$ of pairs satisfying (2.59) and (2.60)
    Obtain $\{\mathcal{A}_{k,x,\ell}, \nu_{k,x,\ell} : k = 1,\ldots,e_{x,\ell}\}$ from (2.62) and (2.63)
  end for
  Merge $\{\mathcal{A}_{k,x,\ell} : x\in\mathcal{N}, k = 1,\ldots,e_{x,\ell}\}$ to obtain $\mathcal{A}_\ell$ and corresponding values $q\left(\mathcal{A}_\ell\right)$ from (2.64) and (2.65)
end for
$\mathrm{OUT} = \{\mathcal{A}_\ell, q\left(\mathcal{A}_\ell\right) : \ell = 1,\ldots,\sigma+1\}$

2.6 Linear Programming Perspective

Inspired by the proposed scheme in the last section, we restrict our discussion to uncoded queries. Then the key step is to design a query encoding function $w\left(q_t|x_\tau, x_t, q_{\lfloor t-1 \rfloor}\right)$, that minimizes the download cost $\mathbb{E}\left[|Q_t|\right]$ subject to two constraints, i.e., the decodability constraint (2.34) and a relaxed privacy constraint (2.35) (protecting the last time when privacy was ON).

For any given $q_{\lfloor t-1 \rfloor}$, or more precisely given the input distribution $p\left(x_t|x_\tau, q_{\lfloor t-1 \rfloor}\right)$, the problem can then be alternatively formulated as a linear programming (LP) instance as follows,

$$\begin{aligned} \underset{p\left(q_t|x_\tau, x_t, q_{\lfloor t-1 \rfloor}\right)}{\text{minimize}} \quad & \mathbb{E}\left[|Q_t||q_{\lfloor t-1 \rfloor}\right] = \sum_{q_t} p\left(q_t|q_{\lfloor t-1 \rfloor}\right)|q_t| \\ \text{subject to} \quad & p\left(x_t, q_t|q_{\lfloor t-1 \rfloor}\right) = 0,\ x_t \notin q_t, \quad \text{(decodability)} \\ & p\left(q_t|x_\tau, q_{\lfloor t-1 \rfloor}\right) = p\left(q_t|q_{\lfloor t-1 \rfloor}\right) \ \text{(relaxed privacy)} \end{aligned} \tag{2.67}$$

This linear programming problem has $n^2\, 2^n$ variables and $(n+2)n\, 2^{n-1}$ constraints, i.e., each probability $p\left(q_t, x_t, x_\tau|q_{\lfloor t-1 \rfloor}\right)$ is a variable where $x_t, x_\tau \in \mathcal{N}$ and $q_t \in \mathscr{P}(\mathcal{N})$. The scale of the problem is intractable in complexity with any generic linear programming solver, for instance Vaidya's algorithm [47] gives $\mathcal{O}\left(\left(n^2\, 2^n\right)^{2.5}\right)$.

One possible strategy dealing with the complexity issue is to impose a restriction on the cardinality of q_t, i.e., $|q_t|$ is chosen from $\{1, 2, \ldots, c, n\}$ where c is a constant and n is included to guarantee the problem is feasible.

$$\begin{aligned} \underset{p\left(q_t|x_\tau, x_t, q_{\lfloor t-1 \rfloor}\right)}{\text{minimize}} \quad & \mathbb{E}\left[|Q_t||q_{\lfloor t-1 \rfloor}\right] = \sum_{q_t} p\left(q_t|q_{\lfloor t-1 \rfloor}\right)|q_t| \\ \text{subject to} \quad & p\left(x_t, q_t|q_{\lfloor t-1 \rfloor}\right) = 0,\ x_t \notin q_t, \\ & p\left(q_t|x_\tau, q_{\lfloor t-1 \rfloor}\right) = p\left(q_t|q_{\lfloor t-1 \rfloor}\right), \\ & |q_t| \in \{1, 2, \ldots, c, n\}. \end{aligned} \tag{2.68}$$

In this way, the number of variables drops dramatically as the alphabet of q_t is reduced from 2^n to the order of n^c, i.e., setting $p\left(q_t|x_\tau, x_t, q_{\lfloor t-1 \rfloor}\right) = 0$ for $|q_t| = \{c+1, \ldots, n-1\}$. Then, the LP instance roughly has n^{c+2} variables, which makes solving the problem numerically possible. For instance if we choose $c = 1$, i.e., the user either downloads the message he wants or all messages

on the server, we can obtain the optimal value to (2.68), which is

$$\mathbb{E}\left[|Q_t||q_{\lfloor t-1\rfloor}\right] = \theta_1(q_{\lfloor t-1\rfloor}) + n\left(1-\theta_1(q_{\lfloor t-1\rfloor})\right), \tag{2.69}$$

where θ_1 was previously defined (c.f.(2.11)) to be

$$\sum_{x\in\mathcal{N}} \min_{x_\tau} p\left(X_t = x | X_\tau = x_\tau, Q_{\lfloor t-1\rfloor} = q_{\lfloor t-1\rfloor}\right).$$

Instead of attempting to solve the linear programming problem numerically, Lemma 2 in the last section actually identifies a feasible solution to the problem (2.67) *efficiently*, and bounds the objective $\mathbb{E}\left[|Q_t||q_{\lfloor t-1\rfloor}\right]$ *analytically*, i.e., a feasible solution attains an objective such that

$$\mathbb{E}\left[|Q_t||q_{\lfloor t-1\rfloor}\right] \leq \sum_{i=1}^{n} i\,\theta_i(q_{\lfloor t-1\rfloor}). \tag{2.70}$$

One can easily see that (2.70) outperforms (2.69).

A helpful observation here is that any algorithmic tractable solution should only visit a small proportion of the power set, i.e, the support set of the query q_t. Otherwise, since the power set is exponentially large, it will introduce an exponential overhead for configuring the probabilities $p\left(q_t|x_\tau, x_t, q_{\lfloor t-1\rfloor}\right)$ for $x_\tau, x_t \in \mathcal{N}$ and $q_t \in \mathscr{P}(\mathcal{N})$.

2.7 Proof of Tightness for $n = 2$ in Theorem 3

In this section, we revisit the case $n = 2$ information sources. As previously stated, we will show the bounds obtained in Theorem 1 and Theorem 2 are tight for the case $n = 2$. We will give an alternate proof to the specially designed one for $n = 2$ presented in [9], this alternate proof relies on the general results presented in Theorem 1 and Theorem 2.

Before starting the proof we discuss some consequences of Theorem 3.

- If $F_t =$ ON, then $\tau = t$ from the definition of τ, then $\frac{1}{R_t} \geq 2$. This means that it is necessary to download both messages, which is consistent with the well-known result for the single server PIR [3].

- If $F_t =$ OFF, it is possible for the user to download less than two messages since $0 \leq \alpha+\beta \leq$

2. We can see that the rate as a function of α and β is symmetric around $\alpha+\beta=1$. When $\alpha+\beta=1$, the Markov chain is independent, i.e., the user's requests are independent, the user can directly ask for the desired message, and the rate is $R_t=1$ (maximum). When $\alpha=\beta=0$ or $\alpha=\beta=1$, i.e., the Markov chain is not ergodic, the user is required to ask for both messages, and then the rate is $R_t=1/2$ (minimum). Another observation is that when the Markov chain is ergodic, the rate goes to 1 when $t-\tau$ goes to infinity. Intuitively, as $t-\tau$ grows, the information carried by X_t about X_τ decreases, so the user can eventually directly ask for the desired message without being concerned about leaking information about X_t.

2.7.1 Converse

It is sufficient to show that the right-hand side of (2.13) equals to $1+|1-\alpha-\beta|^{t-\tau}$. We first write the right-hand side of (2.13) explicitly in terms of α and β. If $\alpha+\beta=0$, then $\alpha=\beta=0$, and we have

$$P^{t-\tau}=P=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix},$$

which yields

$$\sum_{x_t\in\mathcal{N}}\max_{x_\tau\in\mathcal{N}}p\left(x_t|x_\tau\right)=2. \tag{2.71}$$

If $\alpha+\beta\neq 0$, $p\left(x_t|x_\tau\right)$ is given by the transition matrix $P^{t-\tau}$, i.e.,

$$P^{t-\tau}=\frac{1}{\alpha+\beta}\times\begin{bmatrix}\beta+\alpha(1-\alpha-\beta)^{t-\tau} & \alpha-\alpha(1-\alpha-\beta)^{t-\tau}\\ \beta-\beta(1-\alpha-\beta)^{t-\tau} & \alpha+\beta(1-\alpha-\beta)^{t-\tau}\end{bmatrix}. \tag{2.72}$$

Then, we have

$$\sum_{x_t}\max_{x_\tau}p\left(x_t|x_\tau\right)=\begin{cases}1+(1-\alpha-\beta)^{t-\tau}, & (1-\alpha-\beta)^{t-\tau}\geq 0,\\ 1-(1-\alpha-\beta)^{t-\tau}, & (1-\alpha-\beta)^{t-\tau}<0,\end{cases}$$

which can also be written as

$$\sum_{x_t}\max_{x_\tau}p\left(x_t|x_\tau\right)=1+|1-\alpha-\beta|^{t-\tau}. \tag{2.73}$$

By combining (2.71) and (2.73), we get that $\sum_{x_t} \max_{x_\tau} p\left(x_t|x_\tau\right) = 1 + |1-\alpha-\beta|^{t-\tau}$ for any given α and β. Therefore, we have

$$\frac{1}{R_t} \geq \sum_{x_t} \max_{x_\tau} p\left(x_t|x_\tau\right) = 1 + |1-\alpha-\beta|^{t-\tau},$$

which completes the converse proof.

2.7.2 Achievability

From Theorem 2, we know that the rate R_t is achievable if

$$\frac{1}{R_t} \geq \sum_{q_{\lfloor t-1 \rfloor}} p(q_{\lfloor t-1 \rfloor}) \sum_{i=1}^{n} i\, \theta_i(q_{\lfloor t-1 \rfloor}). \tag{2.74}$$

Since $\lambda_1(q_{\lfloor t-1 \rfloor}) \leq 1$ and $\lambda_2(q_{\lfloor t-1 \rfloor}) \geq 1$ for $n = 2$, (2.74) can be rewritten as

$$\frac{1}{R_t} \geq \sum_{q_{\lfloor t-1 \rfloor}} p(q_{\lfloor t-1 \rfloor}) \left(2 - \sum_{x_t} \min_{x_\tau} p\left(x_t|x_\tau, q_{\lfloor t-1 \rfloor}\right)\right). \tag{2.75}$$

In this subsection, we will express the right-hand side of (2.75) explicitly in terms of α and β, and we will show that it is exactly equal to $1 + |1-\alpha-\beta|^{t-\tau}$, as given in (2.19). Also, we will explicitly illustrate the encoding function $w\left(q_t|x_t, x_\tau, q_{\lfloor t-1 \rfloor}\right)$.

From the discussion in Section 2.5, the query encoding function $w(q_t|x_t, x_\tau, q_{\lfloor t-1 \rfloor})$ is given by

$$w\left(q_t|x_t, x_\tau, q_{\lfloor t-1 \rfloor}\right) = \begin{cases} \frac{\pi\left(x_t, q_{\lfloor t-1 \rfloor}\right)}{p\left(x_t|x_\tau, q_{\lfloor t-1 \rfloor}\right)}, & |q_t| = 1, \\ 1 - \frac{\pi\left(x_t, q_{\lfloor t-1 \rfloor}\right)}{p\left(x_t|x_\tau, q_{\lfloor t-1 \rfloor}\right)}, & |q_t| = 2, \end{cases} \tag{2.76}$$

where $\pi\left(x_t, q_{\lfloor t-1 \rfloor}\right)$ is defined by

$$\pi\left(x_t, q_{\lfloor t-1 \rfloor}\right) := \min_{x_\tau \in \{1,2\}} p\left(x_t|x_\tau, q_{\lfloor t-1 \rfloor}\right).$$

Since $q_t \neq \bar{x}_t$ is always true (c.f.(2.34)), where $\bar{x}_t$ is the complement of x_t in the set $\{1, 2\}$, (2.76) is well-defined for any $q_t \in \{\{1\}, \{2\}, \{1, 2\}\}$. As consequences,

1. When $F_t = \text{ON}$, $\tau = t$ by definition, and

$$\min_{x_\tau} p\left(x_t | x_\tau, q_{\lfloor t-1 \rfloor}\right) = \min_{x'_t} p\left(x_t | x'_t, q_{\lfloor t-1 \rfloor}\right) = 0. \tag{2.77}$$

This immediately implies that

$$w\left(q_t | x_t, x_\tau, q_{\lfloor t-1 \rfloor}\right) = \begin{cases} 0, & |q_t| = 1, \\ 1, & |q_t| = 2, \end{cases} \tag{2.78}$$

for any x_t and $q_{\lfloor t-1 \rfloor}$, which means that the user will always download two messages when $F_t = \text{ON}$, i.e.,

$$p\left(|Q_t| = 2\right) = 1. \tag{2.79}$$

2. When $F_t = \text{OFF}$, $\tau \neq t$ by definition. Let

$$\hat{x}_\tau(x_t, q_{\lfloor t-1 \rfloor}) = \arg\min_{x_\tau} p\left(x_t | x_\tau, q_{\lfloor t-1 \rfloor}\right) \tag{2.80}$$

for any x_t and $q_{\lfloor t-1 \rfloor}$. For notational simplicity, $\hat{x}_\tau(x_t, q_{\lfloor t-1 \rfloor})$ will be written as $\hat{x}_\tau$ when x_t and $q_{\lfloor t-1 \rfloor}$ are clear from context. As such, we can see that

$$w\left(q_t | x_t, \hat{x}_\tau, q_{\lfloor t-1 \rfloor}\right) = \begin{cases} 1, & |q_t| = 1, \\ 0, & |q_t| = 2. \end{cases} \tag{2.81}$$

- If $\hat{x}_\tau(x_t, q_{\lfloor t-1 \rfloor})$ is unique, since X_τ and X_t take values in the binary alphabet, it is easy to check that

$$\hat{x}_\tau(x_t, q_{\lfloor t-1 \rfloor}) \neq \hat{x}_\tau(\bar{x_t}, q_{\lfloor t-1 \rfloor}) \tag{2.82}$$

for any given $q_{\lfloor t-1 \rfloor}$. This implies that x_τ and x_t can be determined from each other provided that $|q_t| = 2$. In particular, assume that $|q_{t-1}| = 2$, which implies that $F_{t-1} = \text{OFF}$. We know that x_τ and x_{t-1} can be determined by each other provided that $|q_{t-1}| =$

2, and hence we can easily obtain that

$$\sum_{x_t} \pi\left(x_t, q_{\lfloor t-1 \rfloor}\right) = \sum_{x_t} \min_{x_\tau} p\left(x_t | x_\tau, q_{\lfloor t-1 \rfloor}\right) = \sum_{x_t} \min_{x_{t-1}} p\left(x_t | x_{t-1}\right). \tag{2.83}$$

Correspondingly, we have

$$\begin{aligned} p\left(|q_t| = 1 || q_{t-1}| = 2\right) &= \sum_{q_{\lfloor t-2 \rfloor}} \sum_{x_\tau} p\left(x_\tau, q_{\lfloor t-2 \rfloor} | q_{t-1}\right) \sum_{x_t} p\left(q_t, x_t | x_\tau, q_{\lfloor t-1 \rfloor}\right) \\ &= \sum_{q_{\lfloor t-2 \rfloor}} \sum_{x_\tau} p\left(x_\tau, q_{\lfloor t-2 \rfloor} | q_{t-1}\right) \sum_{x_t} \pi\left(x_t, q_{\lfloor t-1 \rfloor}\right) \\ &= \sum_{q_{\lfloor t-2 \rfloor}} \sum_{x_\tau} p\left(x_\tau, q_{\lfloor t-2 \rfloor} | q_{t-1}\right) \sum_{x_t} \min_{x_{t-1}} p\left(x_t | x_{t-1}\right) \\ &= \sum_{x_t} \min_{x_{t-1}} p\left(x_t | x_{t-1}\right). \end{aligned} \tag{2.84}$$

- If $\hat{x}_\tau(x_t, q_{\lfloor t-1 \rfloor})$ is not unique, i.e., $p\left(x_t | x_\tau, q_{\lfloor t-1 \rfloor}\right) = p\left(x_t | \bar{x}_\tau, q_{\lfloor t-1 \rfloor}\right)$, then we can easily see that

$$w\left(q_t | x_t, x_\tau, q_{\lfloor t-1 \rfloor}\right) = \begin{cases} 1, & |q_t| = 1, \\ 0, & |q_t| = 2, \end{cases} \tag{2.85}$$

for any $x_\tau \in \{1, 2\}$. In particular, if $|q_{t-1}| = 1$, implying that $F_{t-1} =$ OFF, then $\tau < t-1$ by definition, and hence from the fact $x_{t-1} = q_{t-1}$ when $|q_{t-1}| = 1$, we can obtain that

$$\hat{x}_\tau(x_t, q_{\lfloor t-1 \rfloor}) = \arg\min_{x_\tau} p\left(x_t | x_\tau, q_{\lfloor t-1 \rfloor}\right) = \arg\min_{x_\tau} p\left(x_t | x_{t-1}\right), x_{t-1} = q_{t-1}.$$

We can easily see that $\hat{x}_\tau(x_t, q_{\lfloor t-1 \rfloor})$ is not unique in this case, which implies that

$$\sum_{x_t} \pi\left(x_t, q_{\lfloor t-1 \rfloor}\right) = \sum_{x_t} \min_{x_\tau} p\left(x_t | x_\tau, q_{\lfloor t-1 \rfloor}\right) = \sum_{x_t} p\left(x_t | x_{t-1}\right) = 1, \tag{2.86}$$

and

$$p\left(|q_t| = 1 || q_{t-1}| = 1\right) = 1. \tag{2.87}$$

In summary,

1. When $F_t = \text{ON}$, we have $\pi\left(x_t, q_{\lfloor t-1 \rfloor}\right) = 0$, and hence by substituting in (2.75), we can see that

$$\frac{1}{R_t} \geq \sum_{q_{\lfloor t-1 \rfloor}} p(q_{\lfloor t-1 \rfloor}) \left(2 - \sum_{x_t} \min_{x_\tau} p\left(x_t | x_\tau, q_{\lfloor t-1 \rfloor}\right)\right) = 2 \sum_{q_{\lfloor t-1 \rfloor}} p(q_{\lfloor t-1 \rfloor}) = 2. \quad (2.88)$$

2. When $F_t = \text{OFF}$, we have from (2.83) and (2.86) that

$$\sum_{x_t} \pi\left(x_t, q_{\lfloor t-1 \rfloor}\right) = \begin{cases} 1, & |q_{t-1}| = 1, \\ \sum_{x_t} \min_{x_{t-1}} p\left(x_t | x_{t-1}\right), & |q_{t-1}| = 2. \end{cases} \quad (2.89)$$

By substituting in (2.75), we get that

$$\begin{aligned} \frac{1}{R_t} &\geq \sum_{q_{\lfloor t-1 \rfloor}} p(q_{\lfloor t-1 \rfloor}) \left(2 - \sum_{x_t} \min_{x_\tau} p\left(x_t | x_\tau, q_{\lfloor t-1 \rfloor}\right)\right) \\ &= p\left(|Q_{t-1}| = 1\right) \times 2 + p\left(|Q_{t-1}| = 2\right) \times \left(2 - \sum_{x_t} \min_{x_{t-1}} p\left(x_t | x_{t-1}\right)\right) \\ &= 2 - p\left(|Q_{t-1}| = 2\right) \left(\sum_{x_t} \min_{x_{t-1}} p\left(x_t | x_{t-1}\right)\right). \end{aligned} \quad (2.90)$$

From (2.84) and (2.87), we can easily see the following proposition.

Proposition 2. $\{|Q_i| : \tau \leq i \leq t\}$ forms a Markov chain, and the transition matrix is given by

$$\begin{bmatrix} 1 & 0 \\ \sum_{x_t} \min_{x_{t-1}} p\left(x_t | x_{t-1}\right) & 1 - \sum_{x_t} \min_{x_{t-1}} p\left(x_t | x_{t-1}\right) \end{bmatrix}. \quad (2.91)$$

From the definition of τ, we know that $F_\tau = \text{ON}$, and hence $p\left(|Q_\tau| = 2\right) = 1$ from (2.79). Then from Proposition 2, we have

$$p\left(|Q_{t-1}| = 2\right) = \left(1 - \sum_{x_t} \min_{x_{t-1}} p\left(x_t | x_{t-1}\right)\right)^{t-1-\tau}.$$

Hence, (2.90) can be written as

$$\begin{aligned}\frac{1}{R_t} &\geq 2 - p\left(|Q_{t-1}| = 2\right)\left(\sum_{x_t} \min_{x_{t-1}} p\left(x_t|x_{t-1}\right)\right) \\ &= 2 - \left(1 - \sum_{x_t} \min_{x_{t-1}} p\left(x_t|x_{t-1}\right)\right)^{t-1-\tau} \times \left(\sum_{x_t} \min_{x_{t-1}} p\left(x_t|x_{t-1}\right)\right). \end{aligned} \tag{2.92}$$

By substituting α and β in (2.88) and (2.92), we can easily check that both inequalities can be written as ${R_t}^{-1} \geq |1 - \alpha - \beta|^{t-\tau}$. Moreover, one can also check that the encoding function $w\left(q_t|x_t, x_\tau, q_{\lfloor t-1 \rfloor}\right)$ given in (2.76) can be expressed as in Table 2.3.

CHAPTER 3

ON-OFF PRIVACY FOR PAST AND FUTURE CORRELATED REQUESTS

3.1 Introduction

In Chapter 2, we studied ON-OFF privacy in which it was required to ensure privacy for past requests for which privacy was turned ON. In this chapter, we consider a more stringent privacy requirement and want to preserve privacy for both past and future requests.

We follow a setup similar to the one in Chapter 2 in which the user's request are modeled by a Markov chain, but with one significant difference. We assume here that the user knows the requests in a small window of positive size $\omega > 0$ in the future. In practice, this may happen in applications where the user can queue up his requests, such as when watching online videos. Under this new setting, we study the download rate, which is measured by the ratio of the average length of downloaded data to the message length.

Under this model, our main result is summarized in Theorem 4 which: (i) gives a general upper bound on the download rate; and (ii) gives an achievable rate obtained by an ON-OFF privacy scheme having polynomial time complexity in n. One interesting implication of our result, is that the optimal rate does not depend on the window size. Thus, a window of size $\omega = 1$ is sufficient to achieve the optimal rate.

We show that our proposed scheme is optimal, i.e., the upper bound is tight, for a family of Markov chains for $n > 2$ and for all Markov chains with $n = 2$ sources. We also give an implicit characterization of the optimal achievable rate for $n \geq 2$, which relies on solving a linear program (LP) with an exponential number (in n) of variables and constraints. Thus, it is intractable to tackle it using standard LP solvers (e.g., [48]). From that perspective, our results can be viewed as leveraging the special structure of the problem to provide an efficiently computable upper bound and a polynomial time scheme.

3.1.1 Organization

The rest of the chapter is organized as follows. In Section 3.2, we describe the formulation of the ON-OFF privacy problem for past and future requests. We present our main result, Theorem 4, in Section 3.3, and its corollaries in Section 3.4. In Sections 3.5 and 3.6, we propose an efficient ON-OFF privacy scheme that gives the achievable bound in Theorem 4. In Section 3.7, we derive the upper bound, in Theorem 4, on the achievable rate. Finally, we present an implicit characterization of the optimal achievable rate in Section 3.8.

3.2 Problem Formulation

3.2.1 System Model

The system model in this chapter is similar to that presented in Chapter 2. The main difference is that here, at each time t, all past and future requests, when privacy was ON, must be protected. Moreover, we assume that the user knows his future requests in a window of positive size ω. For completion, below we state all the details of the system model.

A single server stores n information sources $\{\mathscr{W}_i : i \in \mathcal{N}\}$, where $\mathcal{N} := \{1, 2, \ldots, n\}$. The system is time-varying, and the time index is assumed to be discrete, i.e., $t \in \mathbb{N}$. At each time t, each source $\mathscr{W}_i$ generates a new message $W_{i,t}$ of length L, which is independent of previously generated messages $\{W_{i,j} : j = 0, \ldots, t-1\}$. Without loss of generality, we assume that $W_{i,t}$ for $i \in \mathcal{N}$ and $t \in \mathbb{N}$ are independently and identically drawn from the uniform distribution over $\{0,1\}^L$.

At time t, the user is interested in retrieving the latest message generated by a desired source, i.e., one of the messages from $\{W_{i,t} : i \in \mathcal{N}\}$. In particular, let X_t be the source of interest at time t, which takes values in $\mathcal{N}$. We call X_t the *user's request* at time t. For notational simplicity, we drop t from $W_{i,t}$ when the time index t is clear from context, i.e., $W_{i,t}$ will be denoted by W_i. To retrieve the desired message, the user is allowed to construct a query Q_t and send this query to the server. Upon receiving the query, the server responds to the user by producing an answer A_t. After receiving the answer, the user should be able to recover the message W_{X_t} that he is interested in.

Meanwhile, the user may wish to hide the identity of his source of interest at time t. Specifically, the user may choose the *privacy status* F_t to be ON or OFF. When F_t is ON, the user wishes to keep

X_t private and when F_t is OFF, the user is not concerned with hiding X_t. We assume that the privacy status $\{F_t : t \in \mathbb{N}\}$ is independent of the user's requests, and the user's privacy status $\{F_i : i \leq t\}$ is known and recorded by both the server and the user at time t.

Our model assumes that the privacy status is independent of the user's requests. As mentioned before, the user may choose privacy to be ON or OFF depending on factors such as location, network connection, device quality, etc.; these factors are generally independent of the user's requests.

In this work, we are particularly interested in the case where the requests X_t form a Markov chain, i.e., $\{X_t : t \in \mathbb{N}\}$ is generated by a (discrete) Markov source. The transition matrix P of the Markov chain is known by both the server and the user, and the transition probability from state i to state j is denoted by $P_{i,j}$.

Moreover, we assume that the user knows his future requests in a window of positive size ω,[1] which means that at time t, the user knows the future requests $\{X_{t+1}, \ldots, X_{t+\omega}\}$ in addition to the current and all past requests $\{X_0, \ldots, X_t\}$. This models several scenarios where user's requests are in a queue. One can think of the situation where the user places his requests in a playlist when watching videos.

The system mainly consists of two encoding functions, which we describe below. Let $\lfloor t \rfloor$ denote $\{0, 1, \ldots, t\}$ and $X_{\lfloor t \rfloor}$ denote $\{X_0, X_1, \ldots, X_t\}$ for $t \in \mathbb{N}$.

Query encoding function: The query Q_t, at time t, is generated by a query encoding function ϕ_t. Given the assumptions that the messages $W_{i,t}$ (as well as the answers A_t) are independent over time and the privacy status $F_{\lfloor t \rfloor}$ are known by both the user and the server, we suppose that ϕ_t is a probabilistic function of the user's known requests[2] $X_{\lfloor t+\omega \rfloor}$ for some $\omega \in \mathbb{N}^+$, i.e.,

$$Q_t = \phi_t \left(X_{\lfloor t+\omega \rfloor}, \mathsf{K} \right), \tag{3.1}$$

where K is the random key to generate a probabilistic query.

Answer encoding function: Accordingly, the answer A_t from the server is given by the answer encoding function ρ_t, which is assumed to be a deterministic function of the query Q_t and the latest

[1] If the window size $\omega = 0$, i.e., no future requests are known, we have to relax the the stringent privacy requirement (3.4) defined here to a weaker sense where only past requests are protected, i.e., the model studied in Chapter 2.

[2] One may also take all previous queries $Q_{\lfloor t-1 \rfloor}$ as variables of the function. However, since $Q_{\lfloor t-1 \rfloor}$ is also a probabilistic function of $X_{\lfloor t+\omega \rfloor}$, the variables of the function can be written as in (3.1).

messages, i.e.,

$$A_t = \rho_t \left(Q_t, W_1, \ldots, W_n \right). \tag{3.2}$$

In particular, the length of answer A_t is assumed to be a function of the query Q_t, and we denote this length by $\ell(Q_t)$. Then, the average length of the answer A_t is given by

$$\ell_t = \mathbb{E}_{Q_t}[\ell \left(Q_t \right)], \tag{3.3}$$

where $\mathbb{E}[\cdot]$ is the expectation operator.

After receiving the answer A_t, the user should be able to recover the desired message from the answer with zero-error probability. This is referred to as the *decodability* condition.

3.2.2 Adversary Model

The adversary is the untrusted server and is assumed to have full statistical knowledge of user's requests and the querying mechanism, that is, the Markov chain's transition probabilities modeling the user's requests and the querying mechanism that generates the queries for information retrieval, respectively.

We assume that the server has no memory constraint, so the server can use all the queries it received up to time t, represented by $Q_{\lfloor t \rfloor}$, and the statistical knowledge of user's requests and the querying mechanism, to infer user's private requests, i.e., all previous requests of which privacy was ON and all future requests. We also assume that the adversary has unbounded computational power and can launch any attack to infer any of the user's private requests.

Privacy is quantified by the mutual information between the user's private requests and the queries released to the server. It is worth noting that the information-theoretic privacy measure is preferable in this work, since it is independent of specific attacking strategies. We consider the most stringent privacy constraint, namely information-theoretic perfect privacy, which requires that absolutely zero information, measured by the mutual information, about the user's private requests is leaked to the server. Formally, it can be written as

$$I\left(X_{\overline{\mathcal{B}}_t}; Q_{\lfloor t \rfloor} \right) = 0, \ \forall t \in \mathbb{N}, \tag{3.4}$$

where $\overline{\mathcal{B}}_t := \{i : i \leq t, F_i = \text{ON}\} \cup \{i : i \geq t+1\}$ denotes the user's private requests, i.e., all previous requests of which privacy was ON and all future requests, and $I(\cdot)$ denotes the mutual information. We refer to (3.4) as the *privacy* condition.

Remark 1. The privacy requirement in (3.4) implies that at time t, only the previous privacy status $\{F_i : i \leq t\}$ is known, and the user may not know whether he will choose privacy to be ON or OFF in the future. For this reason, we have adopted a worst-case formulation in the privacy constraint by assuming that privacy is always ON in the future. In other words, at time t, all previous requests when privacy was ON, as well as all future requests need to be protected. This is characterized by the set $\overline{\mathcal{B}}_t := \{i : i \leq t, F_i = \text{ON}\} \cup \{i : i \geq t+1\}$ in (3.4).

For large messages, the upload cost is negligible relative to the download cost, so in this work, we are interested in minimizing the download cost of the answer at each time, i.e., the average length ℓ_t at time t. By convention, we measure the efficiency by the download rate $R_t = L/\ell_t$, and define the achievable rate region as follows.

Definition 3 (Achievable Rate)**.** The rate tuple $(R_t : t \in \mathbb{N})$ is achievable if there exists a scheme with average download cost ℓ_t such that $R_t \leq L/\ell_t$.

In the rest of this chapter, we will study the achievable region of $(R_t : t \in \mathbb{N})$. In particular, the focus of this work is the characterization of R_t for each $t \in \mathbb{N}$.

3.3 Main Result

Before stating the main result, we introduce some necessary notation. Let $\tau(t)$ be the last time privacy was ON, i.e.,

$$\tau(t) := \max\{i : i \leq t, F_i = \text{ON}\}. \tag{3.5}$$

Without loss of generality, we assume that $F_0 = \text{ON}$, so $\tau(t)$ is always well-defined. Also, when the time index t is clear from context, we drop t from the notation and write $\tau(t)$ as τ for simplicity. For our analysis, it is convenient to define

$$U_t := (X_\tau, X_{t+1}), \tag{3.6}$$

$$
\begin{array}{c} \\ u_{x,1} \\ u_{x,2} \\ \vdots \\ u_{x,m} \end{array}
\begin{array}{c}
\begin{array}{cccc} 1 & 2 & \cdots & n \end{array} \\
\begin{bmatrix}
p(1|u_{1,1}) & p(2|u_{2,1}) & \cdots & p(n|u_{n,1}) \\
p(1|u_{1,2}) & p(2|u_{2,2}) & \cdots & p(n|u_{n,2}) \\
\vdots & \ddots & & \vdots \\
p(1|u_{1,m}) & p(2|u_{2,m}) & \cdots & p(n|u_{n,m})
\end{bmatrix}
\end{array}
$$

Figure 3.1: For a given x, sort the probabilities $p\left(X_t = x|U_t = u\right)$ for $u \in \mathcal{N}^2$ in an ascending order, and store the values in column x where $m = n^2$. $\bar{\lambda}_i$ is the sum of the i^{th} row.

which represents the last request when privacy was ON and the next request of the user at time t, so the alphabet size of U_t is $\mathcal{N}^2$.

Stating our main results calls for the following notation, which we summarize in Figure 3.1. The inherent value of this notation will be apparent when we give the proofs of our main results in later sections. For any given $x \in \mathcal{N}$, we can order the likelihood probabilities $p\left(X_t = x|U_t = u_{x,i}\right)$ such that

$$p\left(X_t = x|U_t = u_{x,1}\right) \leq p\left(X_t = x|U_t = u_{x,2}\right) \leq \cdots \leq p\left(X_t = x|U_t = u_{x,m}\right), \tag{3.7}$$

where $m = n^2$ and $u_{x,i}$ for $i = 1, \ldots, m$ are distinct elements in $\mathcal{N}^2$. Note that probabilities $p\left(X_t = x|U_t = u\right)$ for $x \in \mathcal{N}$ and $u \in \mathcal{N}^2$ can be determined by the given Markov chain. These ordered probabilities can be stored in the columns of a matrix, as shown in Figure 3.1. Then, for $x \in \mathcal{N}$ and $i = 1, \ldots, m$, let $\bar{\lambda}_i(t)$ be the summation of row i of this matrix, more formally,

$$\bar{\lambda}_i\left(t\right) = \sum_{x \in \mathcal{N}} p\left(X_t = x|U_t = u_{x,i}\right). \tag{3.8}$$

Also, for $i = 1, \ldots, n$, let

$$\bar{\theta}_i(t) = \begin{cases} \bar{\lambda}_i(t) - \bar{\lambda}_{i-1}(t), & i < n, \\ 1 - \bar{\lambda}_{i-1}(t), & i = n, \end{cases} \tag{3.9}$$

where $\bar{\lambda}_0\left(t\right)$ is assumed to be 0. For notational simplicity, let

$$\frac{1}{R_t^I} := \sum_{i=1}^{n} i\, \bar{\theta}_i(t), \tag{3.10}$$

and

$$\frac{1}{R_t^O} := \bar{\lambda}_m(t) = \sum_{x\in\mathcal{N}} \max_{u\in\mathcal{N}^2} p(X_t = x|U_t = u), \tag{3.11}$$

where $\bar{\lambda}_m(t)$ is defined in (3.8). We may drop the time index t when it is clear from context, that is, we will write $\bar{\lambda}_i(t)$ and $\bar{\theta}_i(t)$ as $\bar{\lambda}_i$ and $\bar{\theta}_i$, respectively. With this notation, we are ready to state the main theorem.

Theorem 4. Suppose that $\{X_t : t \in \mathbb{N}\}$ is a Markov process with the transition matrix P. The rate tuple $(R_t : t \in \mathbb{N})$ is achievable if

$$\frac{1}{R_t} \geq \frac{1}{R_t^I}, \tag{3.12}$$

where $1/R_t^I$ is defined in (3.10). On the other hand, any achievable rate tuple $(R_t : t \in \mathbb{N})$ must satisfy

$$\frac{1}{R_t} \geq \frac{1}{R_t^O}, \tag{3.13}$$

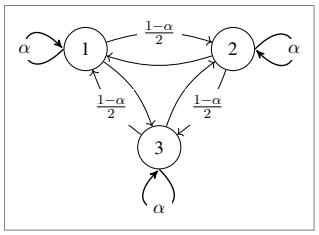
where $1/R_t^O$ is defined in (3.11).

Remark 2 (Single server PIR)**.** As mentioned earlier, the single server private information retrieval problem can be viewed as a special case of this setting where privacy is always ON. As a sanity check, if $F_t = \text{ON}$ for all $t \in \mathbb{N}$, we have $U_t = (X_t, X_{t+1})$ by the definition (3.6), and then we can easily see that $\max_{u_t} p(x_t|u_t) = 1$, for all $x_t \in \mathcal{N}$. Thus, we know from Theorem 4 that R_t is achievable only if

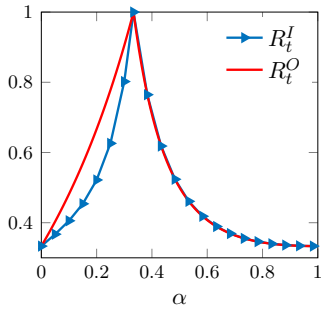
$$R_t \leq R_t^O = \frac{1}{\bar{\lambda}_m(t)} = \frac{1}{n},$$

which implies that it is necessary to download all messages when the privacy is ON. This is consistent with the well- known result in the literature on PIR[3].

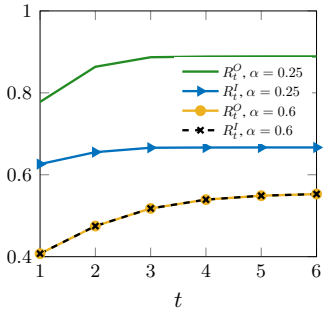
In the rest of this chapter we prove Theorem 4. In particular, we propose a polynomial-time querying scheme that achieves R_t^I in sections 3.5 and 3.6. As discussed in the previous remark, the user has to query for all the messages when privacy is ON, so our focus will be on the instances when privacy is OFF. Roughly speaking, in our proposed probabilistic querying scheme, the user asks for a subset of the messages containing the message in which he is interested. The user generates his query based on his knowledge of his previous requests when privacy was ON, his current request, and his next request. Moreover, the proof of the upper bound R_t^O will be presented in Section 3.7.



(a) A Symmetric Markov Chain.



(b) R_t^I and R_t^O as a function of α.



(c) R_t^I and R_t^O as a function of time (t).

Figure 3.2: In Figure 3.2a, we graphically represent the 3-state symmetric Markov chain used in Example 2, where $0 \leq \alpha \leq 1$. In Figure 3.2b, we plot the achievable rate R_t^I(c.f.(3.12)) and the upper bound R_t^O(c.f.(3.13)), as a function of α, when $\tau = 0$ and $t = 1$. In Figure 3.2c, we plot R_t^I and R_t^O as a function of time t for both $\alpha = 0.25$ and $\alpha = 0.6$.

3.4 Optimality for Special Families of Markov Chains

Before we proceed to prove Theorem 4, we give two corollaries that characterize two special classes of Markov chains for which the bounds in Theorem 4 are tight, i.e., $R_t^I = R_t^O$, which means that our proposed scheme is optimal for these special cases.

Corollary 2 (Optimality for $n = 2$)**.** For the case $n = 2$, the two bounds (3.12) and (3.13) match, i.e.,

$$\frac{1}{R_t^I} = \frac{1}{R_t^O} = \bar{\lambda}_m(t). \tag{3.14}$$

In other words, the rate tuple $(R_t : t \in \mathbb{N})$ is achievable if and only if

$$R_t \leq \frac{1}{\bar{\lambda}_m(t)}. \tag{3.15}$$

Definition 4 (Symmetric Markov Chain)**.** A Markov chain is symmetric if its transition matrix P is given by

$$P_{i,j} = \begin{cases} \alpha, & \text{if} \quad i = j, \\ \frac{1-\alpha}{n-1}, & \text{if} \quad i \neq j, \end{cases} \tag{3.16}$$

where $0 \leq \alpha \leq 1$ and $P_{i,j}$ denotes the transition probability from state i to state j.

Corollary 3 (Optimality for Symmetric Markov Chain)**.** For the symmetric Markov chain such that

$\frac{1}{n} \leq \alpha \leq 1$, the two bounds (3.12) and (3.13) match. In particular,

$$\frac{1}{R_t^I} = \frac{1}{R_t^O} = \alpha n \frac{(n-1)^{t-\tau} + (n-1)(n\alpha - 1)^{\tau}}{(n-1)^{t-\tau} + (n\alpha - 1)^{t-\tau+1}}. \tag{3.17}$$

In other words, the rate tuple $(R_t : t \in \mathbb{N})$ is achievable if and only if

$$\frac{1}{R_t} \geq \alpha n \frac{(n-1)^{t-\tau} + (n-1)(n\alpha - 1)^{\tau}}{(n-1)^{t-\tau} + (n\alpha - 1)^{t-\tau+1}}. \tag{3.18}$$

The proofs of Corollaries 2 and 3 can be found in Appendix B.1 and Appendix B.2, respectively.

Example 2. We study a special case described in Corollary 3. Suppose that we are given $\tau = 0$, and a 3-state Markov chain, as represented in Figure 3.2a, where $0 \leq \alpha \leq 1$.

In this case, we have two regimes, one for $\alpha < \frac{1}{3}$ and the other for $\alpha \geq \frac{1}{3}$. This is because the ordering of probabilities (c.f.(3.7)) changes at $\alpha = \frac{1}{3}$.

For $\alpha \geq \frac{1}{3}$, the bounds (3.12) and (3.13) match, e.g., for $t = 1$,

$$\frac{1}{R_1^O} = \frac{1}{R_1^I} = \frac{6\alpha^2}{3\alpha^2 - 2\alpha + 1}. \tag{3.19}$$

However, for $\alpha < \frac{1}{3}$ and $t = 1$, we have

$$\frac{1}{R_1^O} = \frac{3 - 3\alpha}{3\alpha + 1} \leq \frac{1}{R_1^I} = \frac{2}{3\alpha + 1} - \frac{4\alpha - 2}{3\alpha^2 - 2\alpha + 1} - 1. \tag{3.20}$$

We illustrate (3.19) and (3.20) in Figure 3.2b.

In Figure 3.2c, we analyze the rate over time for $\alpha = 0.25$, and $\alpha = 0.6$. It is notable that as t grows, the correlation between X_t (the current request) and X_τ (the request when privacy was last ON) decreases, which leads to an increase in the download rate R_t.

3.5 Achievability: Linear Programming Formulation

Towards finding an ON-OFF privacy scheme, we consider uncoded queries for retrieving messages, i.e., the query Q_t at time t takes values in the power set of $\mathcal{N}$, denoted by $\mathscr{P}(\mathcal{N})$. In other words, the user will query for a subset of the messages $W_{\mathcal{N}}$ at each time. Later in this section, we will see that designing an uncoded query scheme is equivalent to solving a linear programming problem.

Upon receiving the query $Q_t \subseteq \mathcal{N}$, the server generates a corresponding answer $A_t = W_{Q_t} \subseteq W_{\mathcal{N}}$. The length of the answer can be written as

$$\ell\left(Q_t\right) = \left|Q_t\right| L,$$

where L is the length of a message. Therefore, the average length ℓ_t is

$$\ell_t = \mathbb{E}\left[\left|Q_t\right|\right] L. \tag{3.21}$$

Next, we describe how to construct the query Q_t for each time t. The query Q_t is a probabilistic function of the current request X_t and $U_t = (X_\tau, X_{t+1})$(c.f.(3.6)). Therefore, the encoding of the query Q_t can be equivalently denoted by the probability distribution $w\left(q_t|x_t, u_t\right)$, where $x_t \in \mathcal{N}$, $u_t = (x_\tau, x_{t+1}) \in \mathcal{N}^2$ and $q_t \in \mathscr{P}\left(\mathcal{N}\right)$. In other words, given $x_t \in \mathcal{N}$ and $u_t = (x_\tau, x_{t+1}) \in \mathcal{N}^2$, the user will send $q_t \in \mathscr{P}\left(\mathcal{N}\right)$ with probability $w\left(q_t|x_t, u_t\right)$.

Since $A_t = W_{Q_t}$, if $X_t \in Q_t$, then the retrieved answer contains the desired message W_{X_t}. Therefore, if

$$p\left(q_t, x_t|u_t\right) = 0, \quad \forall x_t \notin q_t, \tag{3.22}$$

then decodability is guaranteed. Note that $p\left(q_t, x_t|u_t\right)$ can be written as

$$p\left(q_t, x_t|u_t\right) = p\left(x_t|u_t\right) w\left(q_t|x_t, u_t\right),$$

where $p\left(x_t|u_t\right)$ is given by the Markov chain, so $p\left(q_t, x_t|u_t\right)$ is determined by $w\left(q_t|x_t, u_t\right)$.

To guarantee the privacy(c.f.(3.4)), we introduce the following lemma. It states that if we design the encoding function $w\left(q_t|x_t, u_t\right)$ such that

$$p\left(q_t|u_t\right) = p\left(q_t\right), \quad \forall u_t \in \mathcal{N}^2, q_t \in \mathscr{P}\left(\mathcal{N}\right), \tag{3.23}$$

for all $t \in \mathbb{N}$, then the scheme satisfies the required privacy constraint (3.4).

Lemma 3. If Q_i is a probabilistic function of U_i and X_i, and Q_i is independent of U_i for $i = 0, 1, \ldots, t$, then $Q_{\lfloor t \rfloor}$ is independent of $X_{\overline{\mathcal{B}}_t}$, where $\overline{\mathcal{B}}_t = \{i : i \leq t, F_i = \text{ON}\} \cup \{i : i \geq t+1\}$.

Proof. See Appendix B.3. □

Since the download cost of the scheme is as given in (3.21), i.e., $\ell_t = \mathbb{E}\left[|Q_t|\right] L$, and we desire a scheme with low download cost (high rate), we would like to design an encoding function $w\left(q_t|x_t, u_t\right)$ that minimizes $\mathbb{E}\left[|Q_t|\right]$.

Hence, it remains to design the distribution $w\left(q_t|x_t, u_t\right)$ that minimizes $\mathbb{E}\left[|Q_t|\right]$ under the constraints (3.22) and (3.23). As such, any feasible solution to the following optimization problem corresponds to an admissible encoding function $w\left(q_t|x_t, u_t\right)$ as desired.

$$\begin{aligned} \underset{w(q_t|x_t,u_t)}{\text{minimize}} \quad & \mathbb{E}\left[|Q_t|\right] = \sum_{q_t} p\left(q_t\right)|q_t| \\ \text{subject to} \quad & p\left(x_t, q_t|u_t\right) = 0,\ x_t \notin q_t, \\ & p\left(q_t|u_t\right) = p\left(q_t\right). \end{aligned} \tag{3.24}$$

Note that the problem is always feasible, as

$$w\left(q_t = \mathcal{N}|x_t, u_t\right) = 1,\ \forall u_t, x_t, \tag{3.25}$$

is a feasible solution to (3.24).

One may also notice that if we treat each probability $w\left(q_t|x_t, u_t\right)$ for $x_t \in \mathcal{N}$, $u_t \in \mathcal{N}^2$ and $q_t \in \mathscr{P}\left(\mathcal{N}\right)$ as a decision variable, then both the objective function and two constraints are linear, and hence the optimization problem (3.24) is indeed a linear programming instance. However, this linear programming problem has $n^3\, 2^n$ variables and $n\, 2^{n-1} + n^2\, 2^n$ constraints. The scale of the problem is intractable in complexity with any generic linear programming solver. For example, using the techniques presented in [47], the complexity of this linear programming is $\mathcal{O}\left(\left(n^2\, 2^n\right)^{2.5}\right)$. This makes the numerical solution impossible when n is large.

Therefore, in the following section, we present a polynomial time algorithm that gives a feasible solution that might not always be optimal.

3.6 Efficient ON-OFF Privacy Query Scheme

Instead of attempting to solve the linear programming problem (3.24) numerically, we are going to identify a feasible solution $w^*\left(q_t|x_t,u_t\right)$ to the problem *efficiently*, and bound the objective $\mathbb{E}\left[|Q_t|\right]$ *analytically*, i.e., a feasible solution attains an objective such that

$$\mathbb{E}\left[|Q_t^*|\right] \leq 1/R_t^I = \sum_{i=1}^{n} i\,\bar{\theta}_i(t), \tag{3.26}$$

which means there exists a scheme such that the download cost ℓ_t is less than or equal to L/R_t^I, or R_t^I is achievable.

A key observation on (3.24) is that any tractable solution $w\left(q_t|x_t,u_t\right)$ must be sparse, i.e., a few non-zero valued probabilities $w\left(q_t|x_t,u_t\right)$ for $x_t \in \mathcal{N}$, $u_t \in \mathcal{N}^2$ and $q_t \in \mathscr{P}(\mathcal{N})$. Otherwise, simply initializing or outputting the solution $w\left(q_t|x_t,u_t\right)$ introduces an exponential overhead in complexity. This observation motivates our algorithm, which admits a sparse $w\left(q_t|x_t,u_t\right)$.

Since the time index t will be clear from context, in the sequel we drop it from the subscripts. For any given $p\left(x|u\right)$, we recall the optimization problem we are interested in,

$$\begin{aligned} \underset{w(q|x,u)}{\text{minimize}} \quad & \mathbb{E}\left[|Q|\right] = \sum_{q} p\left(q\right)|q| \\ \text{subject to} \quad & p\left(x,q|u\right) = 0,\ x \notin q, \\ & p\left(q|u\right) = p\left(q\right), \end{aligned} \tag{3.27}$$

where $x \in \mathcal{N}$, $u \in \mathcal{N}^2$ and $q \in \mathscr{P}(\mathcal{N})$.

Instead of finding a feasible solution to (3.27) directly, we introduce an auxiliary random variable Z. Let Z be a multiset $(\mathcal{N}, f)$, where $\mathcal{N}$ is the ground set and f is the multiplicity function. The cardinality of the multiset Z is the summation of multiplicities of all its element, i.e.,

$$|Z| = \sum_{x \in \mathcal{N}} f(x).$$

Let $\mathcal{Z}$ be the collection of all multisets such that cardinality is bounded by n, i.e.,

$$\mathcal{Z} = \left\{Z : Z \in (\mathcal{N}, f)\,, |Z| \leq n\right\}. \tag{3.28}$$

Then for any given $p(x|u)$, we can define an alternative optimization problem:

$$\begin{aligned} \underset{w(z|x,u)}{\text{minimize}} \quad & \mathbb{E}[|Z|] = \sum_{z} p(z)\,|z| \\ \text{subject to} \quad & p(x,z|u) = 0,\ x \notin z, \\ & p(z|u) = p(z), \end{aligned} \tag{3.29}$$

where $x \in \mathcal{N}$, $u \in \mathcal{N}^2$ and $z \in \mathcal{Z}$.

One can easily check that any feasible solution to (3.29) can be easily transformed to be a feasible solution to (3.27) by simply letting $Q = \text{Set}(Z)$, i.e., forcing the multiplicity of elements in Z to be 1. Moreover, the corresponding solution to (3.27) attains a better objective value, i.e., if $w(z|x,u)$ is a feasible solution to (3.29) and $w(q|x,u)$ is a feasible solution to (3.27), then $\mathbb{E}[|Q|] \leq \mathbb{E}[|Z|]$, where $\mathbb{E}[|Z|]$ is the objective value attained by $w(z|x,u)$ and $\mathbb{E}[|Q|]$ is the objective value attained by $w(q|x,u)$, respectively. Therefore, we will study the feasible region of (3.29) instead. In particular, we will find a feasible solution $w^*(z|x,u)$ such that

$$\mathbb{E}[|Z^*|] = \sum_{i=1}^{n} i\,\bar{\theta}_i. \tag{3.30}$$

Then there exists a corresponding feasible solution $w^*(q|x,u)$, by simply letting $Q = \text{Set}(Z)$, to the original problem (3.27) such that

$$\mathbb{E}[|Q^*|] \leq \sum_{i=1}^{n} i\,\bar{\theta}_i, \tag{3.31}$$

which is the same as (3.26) and is to be proved.

In the remainder of this section, we start by describing the algorithm in Subsection 3.6.1. We then analyze its complexity in Subsection 3.6.2, and finally in Subsection 3.6.3, we verify that the algorithm outputs a feasible solution as desired.

3.6.1 Algorithm Description

In this section, we describe the algorithm to construct a feasible solution $w(z|x,u)$ to (3.29), i.e., for any given distribution $p(x|u)$, we will give a constructive proof of some Z, satisfying that

$$p(z,x|u)=0,\ \forall x\notin z, \tag{3.32}$$

and

$$p(z|u)=p\left(z|u'\right),\ \forall z\in\mathcal{Z} \text{ and } u,u'\in\mathcal{N}^2. \tag{3.33}$$

In particular, we will show that the feasible solution $w(z|x,u)$ gives

$$p(|Z|=\ell)=\bar{\theta}_\ell,\ \ell=1,\ldots,n. \tag{3.34}$$

Note that $\bar{\theta}_i\geq 0$ for all $i=1,\ldots,n$ by the definition (3.9), which is stated in the following proposition.

Proposition 3. For any given Markov chain and time index t, $\bar{\theta}_i\geq 0$, for $i=1,\ldots,n$.

Proof. See Appendix B.4. □

One can see that the objective value attained by this feasible solution is

$$\mathbb{E}[|Z|]=\sum_{i=1}^{n} i\,\bar{\theta}_i.$$

Before describing the steps of the algorithm we give an intuitive explanation and overview of the algorithm. In order to minimize $\mathbb{E}[|Z|]$, we would like to construct some $w(z|x,u)$ that makes the probability $p(|Z|=\ell)$ larger for smaller ℓ, i.e., a greedy-like algorithmic approach is appealing. As a result of the two constraints (3.32) and (3.33), one can easily check that the maximum value of $p(|Z|=1)$ is $\bar{\theta}_1$, and the solution gives

$$p(z,x|u)=\min_{u'\in\mathcal{N}^2} p\left(x|u'\right),\ \forall u \text{ and } z=x.$$

We would like to keep this greedy manner to manage the probabilities $p(z,x|u)$ for $|z|=2,\ldots,n$.

However, when $|z| \geq 2$, it becomes more complicated. For instance, when $|z| = 2$, one of the two elements of the set z has to be x, in order to satisfy (3.32), which corresponds to the decodability constraint. Roughly speaking, we aim to use the second element of z to obfuscate each x with another x' in order to satisfy (3.33), which corresponds to the privacy constraint. The challenging part of this algorithm is this choice of x', and the corresponding probability $p(z, x|u)$, where $z = \{x, x'\}$.

The following algorithm, consisting of five main steps, rigorously describes how we design this obfuscation. In Step 1, we calculate preliminaries from the given probability distribution $p\left(x|u\right)$ and initialize the algorithm. In Step 2, we describe how to properly obfuscate each x with the other $\ell - 1$ elements for a given ℓ, and in Step 3, we describe how to design a common obfuscation (obtain some common sets z of cardinality ℓ and some proper values) for all $x \in \mathcal{N}$ simultaneously. Then, in Step 4, we augment the configurations to the initialized variables, and finally in Step 5, we output the configurations and the values. Details are given as follows:

• Step 1: Preliminaries

For any given distribution $p\left(x|u\right)$, by sorting $p\left(x|u\right)$ for each $x \in \mathcal{N}$, we can easily obtain parameters

$$\left\{u_{x,i} : x \in \mathcal{N}, i = 1, \ldots, m\right\},$$

where $u_{x,i}$ is as defined in (3.7) and $m = |\mathcal{N}^2| = n^2$. For notational simplicity, let

$$\bar{\lambda}_{x,i} = p\left(X_t = x|U_t = u_{x,i}\right).$$

Let M be an auxiliary $m \times n$ matrix determined by the given $p\left(x|u\right)$. In particular, we initialize M by

$$M_{i,j} = \max\left\{p\left(X = j|U = i\right) - \bar{\lambda}_{j,n-1}, 0\right\}. \tag{3.35}$$

for $i = 1, \ldots, m$, and $j = 1, \ldots, n$. This matrix will be updated during the following procedure. For the ease of notation, let $M_{i,j}^{-} = a$ denote $M_{i,j} = M_{i,j} - a$, i.e., subtracting a from $M_{i,j}$.

For $\ell = 1, \ldots, n-1$ and $x = 1, \ldots, n$, we access to $\{u_{x,i} : i = 1, \ldots, \ell - 1\}$. For ease of notation, let

$$\mathcal{U}_{\ell,x}^{-} = \left\{u_{x,i} : i = 1, \ldots, \ell - 1\right\},$$

and

$$\mathcal{U}_{\ell,x}^{+} = \{u_{x,i} : i = \ell, \ldots, m\}.$$

• **Step 2:**

For each i, or precisely $u_{x,i}$, we choose a collection of pairs

$$I_{\ell,x,i} \times V_{\ell,x,i} = \{(e_{\ell,x,i,j}, v_{\ell,x,i,j}) : j = 1, 2, \ldots, c_{\ell,x,i}\} \tag{3.36}$$

such that

$$0 \leq v_{\ell,x,i,j} \leq M_{u_{x,i},e_{\ell,x,i,j}}, \tag{3.37}$$

and

$$\sum_{j=1}^{c_{\ell,x,i}} v_{\ell,x,i,j} = \bar{\lambda}_{x,\ell} - \bar{\lambda}_{x,\ell-1}. \tag{3.38}$$

where $e_{\ell,x,i,j}$ for $j = 1, \ldots, c_{\ell,x,i}$ are distinct indices belonging to $\{1, \ldots, n\}$, and clearly we have $c_{\ell,x,i} \leq n$.

Then, we update the matrix M by

$$M^{-}_{u_{x,i},e_{\ell,x,i,j}} = v_{\ell,x,i,j}, \tag{3.39}$$

for all $e_{\ell,x,i,j} \in I_{\ell,x,i}$. We slightly abuse the notation here by using the same notation M to denote the matrix at different points. Nevertheless, the underlying ℓ, x and i we are dealing with will be clear from context.

Roughly speaking, we extract $\bar{\lambda}_{x,\ell} - \bar{\lambda}_{x,\ell-1}$ from the $u_{x,i}$-th row of the non-negative matrix M for given ℓ and x, where $e_{\ell,x,i,j}$ and $v_{\ell,x,i,j}$ specify the column indices and values extracted from each position of $u_{x,i}$-th row. The matrix M is always non-negative during the update from (3.37) and (3.39), so the existence of such a collection of $I_{\ell,x,i} \times V_{\ell,x,i}$ can be guaranteed if the summation of the $u_{x,i}$-th row of the initialized matrix M(c.f.(3.35)) is greater than or equal to the summation of the subtracted values (the right-hand side of (3.40)) for all x and ℓ during the process, which is given by the following proposition.

Proposition 4. For any $u = 1, \ldots, m$,

$$\sum_{x=1}^{n} \max\left\{p\left(X = x|U = u\right) - \bar{\lambda}_{x,n-1}, 0\right\} \geq \sum_{\ell=1}^{n-1} \sum_{x:u\in\mathcal{U}_{\ell,x}^{-}} \left(\bar{\lambda}_{x,\ell} - \bar{\lambda}_{x,\ell-1}\right). \tag{3.40}$$

Proof. See Appendix B.5. □

• Step 3:

For fixed ℓ and x, after finishing the above process for all $i = 1, \ldots, \ell - 1$, we obtain $I_{\ell,x,i}$ and $V_{\ell,x,i}$ for $i = 1, \ldots, \ell - 1$. Provided $I_{\ell,x,i}$ and $V_{\ell,x,i}$ for $i = 1, \ldots, \ell - 1$, we pick a collection of pairs

$$\{(\zeta_{\ell,x,k}, \nu_{\ell,x,k}) : k = 1, 2, \ldots, c_{\ell,x}\}$$

such that

$$\zeta_{\ell,x,k} \in I_{\ell,x,1} \times I_{\ell,x,2} \times \cdots \times I_{\ell,x,\ell-1},$$

and

$$\sum_{k:\zeta_{\ell,x,k}(i)=e_{\ell,x,i,j}} \nu_{\ell,x,k} = v_{\ell,x,i,j}, \tag{3.41}$$

for all $i = 1, \ldots, \ell - 1$ and $j = 1, \ldots, c_{\ell,x,i}$, where $\zeta_{\ell,x,k}(i)$ is the i-th element of $\zeta_{\ell,x,k}$, i.e., $\zeta_{\ell,x,k}(i) \in I_{\ell,x,i}$.

A simple deterministic approach of picking such a collection of $(\zeta_{\ell,x,k}, \nu_{\ell,x,k})$ can be basically illustrated by Figure 3.3. Roughly speaking, there is a buffer tracking the front of the sets $V_{\ell,x,i}$ for $i = 1, \ldots, \ell - 1$. Each time, the buffer pushes the minimal value among them i.e., $\nu_{\ell,x,k}$, minus the value from the front, and adds one more value from the same set $V_{\ell,x,i}$ which has been pushed out. The corresponding positions of values in the buffer form the set $\zeta_{\ell,x,k}$. As such, we can easily see that

$$\sum_{k=1}^{c_{\ell,x}} \nu_{\ell,x,k} = \bar{\lambda}_{x,\ell} - \bar{\lambda}_{x,\ell-1}. \tag{3.42}$$

Also, one can easily check that this process returns

$$c_{\ell,x} = \sum_{i=1}^{\ell-1} c_{\ell,x,i} \leq n(\ell - 1). \tag{3.43}$$

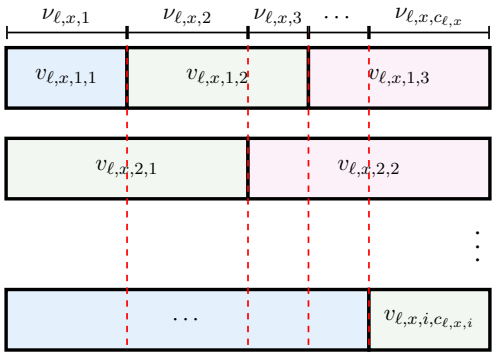


Figure 3.3: The rows represents $V_{\ell,x,1}, \ldots, V_{\ell,x,\ell-1}$ for given ℓ and x. Each block represents an element $v_{\ell,x,i,j}$ in the set $V_{\ell,x,i}$, where $j = 1, \ldots, c_{\ell,x,i}$. Each $\nu_{\ell,x,k}$ can be chosen to be the value of the difference between two consecutive boundaries of blocks, e.g., $\nu_{\ell,x,1} = v_{\ell,x,1,1}$ and $\nu_{\ell,x,2} = v_{\ell,x,2,1} - v_{\ell,x,1,1}$ etc. The corresponding $\zeta_{\ell,x,k}$ can be chosen to be $\zeta_{\ell,x,1} = (e_{\ell,x,1,1}, e_{\ell,x,2,1}, \cdots)$ and $\zeta_{\ell,x,2} = (e_{\ell,x,1,2}, e_{\ell,x,2,1}, \cdots)$ etc.

• Step 4: Augment

For each $k = 1, \ldots, c_{\ell,x}$, let

$$z_{\ell,x,k} = \{\zeta_{\ell,x,k}, x\}, \tag{3.44}$$

and

$$\begin{aligned} \mathcal{F}_{\ell,x,k} = &\left\{(\bar{z}, \bar{x}, \bar{u}) : \bar{z} = z_{\ell,x,k}, \bar{x} = x, \bar{u} \in \mathcal{U}^{+}_{\ell,x}\right\} \bigcup \\ &\left\{(\bar{z}, \bar{x}, \bar{u}) : \bar{z} = z_{\ell,x,k}, \bar{x} = \zeta_{\ell,x,k}(i), \bar{u} = u_{x,i} \in \mathcal{U}^{-}_{\ell,x}\right\}. \end{aligned} \tag{3.45}$$

For each ℓ and x, we can obtain $\mathcal{F}_{\ell,x,k}$ and $\nu_{\ell,x,k}$ for $k = 1, \ldots, c_{\ell,x}$. The tuple $(\bar{z}, \bar{x}, \bar{u})$ in $\mathcal{F}_{\ell,x,k}$ is indeed the non-zero valued position and $\nu_{\ell,x,k}$ is the value that we will assign to the probability $p(\bar{z}, \bar{x}|\bar{u})$. However, since there may exist duplicated tuples in $\mathcal{F}_{\ell,x,k}$ for different x, we augment the value $\nu_{\ell,x,k}$ corresponding to the same tuple $(\bar{z}, \bar{x}, \bar{u})$, i.e.,

$$\mathcal{F}_{\ell} = \bigcup_{x=1}^{n} \bigcup_{k=1}^{c_{\ell,x}} \mathcal{F}_{\ell,x,k}, \tag{3.46}$$

and for any $(\bar{z}, \bar{x}, \bar{u}) \in \mathcal{F}_{\ell}$,

$$g(\bar{z}, \bar{x}, \bar{u}) = \sum_{x=1}^{n} \sum_{k:(\bar{z},\bar{x},\bar{u}) \in \mathcal{F}_{\ell,x,k}} \nu_{\ell,x,k}. \tag{3.47}$$

After obtaining $\mathcal{F}_\ell$ for $\ell = 1, \ldots, n-1$, for $\ell = n$, let

$$\mathcal{F}_n = \{(\bar{z}, \bar{x}, \bar{u}) : \bar{z} = \mathcal{N}, M_{\bar{u},\bar{x}} > 0\}, \tag{3.48}$$

and

$$g(\bar{z}, \bar{x}, \bar{u}) = M_{\bar{u},\bar{x}}, \tag{3.49}$$

for any $(\bar{z}, \bar{x}, \bar{u}) \in \mathcal{F}_n$.

• **Step 5: Output**

The output of the algorithm is $\{\mathcal{F}, g(\mathcal{F})\}$, where

$$\mathcal{F} = \{\mathcal{F}_\ell : \ell = 1, \ldots, n\}.$$

stores the non-zero valued positions of an admissible distribution $p(z, x|u)$ for $x \in \mathcal{N}$, $u \in \mathcal{N}^2$ and $z \in \mathcal{Z}$, and $g(\mathcal{F})$ stores the corresponding probabilities.

3.6.2 Complexity

For the sake of completeness, we discuss the complexity of the algorithm. As said, the bottleneck is to represent the solution $w(z|x, u)$ for $z \in \mathcal{Z}$, $x \in \mathcal{N}$ and $u \in \mathcal{N}^2$, which has exponential number of values, so the complexity is indeed dominated by the size of $\mathcal{F}$, i.e., the non-zero valued positions of the output distribution $p(z, x|u)$.

It is notable that $|\mathcal{F}_{\ell,x,k}| = m$ and $c_{\ell,x} \leq n^2$ from (3.43), so we have

$$|\mathcal{F}| \leq \sum_{\ell=1}^{n} \sum_{x=1}^{n} \sum_{k=1}^{c_{\ell,x}} |\mathcal{F}_{\ell,x,k}| \leq mn^4 = n^6, \tag{3.50}$$

i.e., the complexity of the algorithm is $\mathcal{O}(n^6)$.

The purpose of the complexity analysis here is to justify that the proposed algorithm is with poly(n) complexity. One may possibly reduce the complexity by orders by utilizing some data structures, which is beyond the interest of this work.

3.6.3 Algorithm Verification

In this subsection, we will verify the algorithm, i.e., we will prove that it outputs a distribution $p(z, x|u)$ satisfying (3.32), (3.33) and (3.34) for any given distribution $p(x|u)$.

First, we show that the algorithm described in 3.6.1 outputs a distribution $p(z, x|u)$ satisfying (3.32) and (3.33) for any given distribution $p(x|u)$.

Proposition 5. For any given $p(x|u)$ for $u \in \mathcal{N}^2$ and $x \in \mathcal{N}$, $\{\mathcal{F}, g(\mathcal{F})\}$ returns non-zero valued positions and values of some distribution $p(z, x|u)$ such that $p(z, x|u) = 0$ for all $x \notin z$ and $p(z|u) = p(z|u')$ for all $z \in \mathcal{Z}$ and $u, u' \in \mathcal{N}^2$.

Proof. As claimed, $\mathcal{F}$ and $g(\mathcal{F})$ store the non-zero valued positions and values of $p(z, x|u)$, so it is equivalent for us to show that

1. For any $(\bar{z}, \bar{x}, \bar{u}) \in \mathcal{F}$, we have
$$\bar{x} \in \bar{z}. \tag{3.51}$$

2. For any given $\bar{z}$, $\bar{u}$ and $\bar{u}'$, we have
$$\sum_{x:(\bar{z},x,\bar{u})\in\mathcal{F}} g(\bar{z}, x, \bar{u}) = \sum_{x:(\bar{z},x,\bar{u}')\in\mathcal{F}} g(\bar{z}, x, \bar{u}'). \tag{3.52}$$

3. For any given $\bar{x}$ and $\bar{u}$, we have
$$\sum_{z:(z,\bar{x},\bar{u})\in\mathcal{F}} g(z, \bar{x}, \bar{u}) = p(\bar{x}|\bar{u}). \tag{3.53}$$

Details can be found in Appendix B.6. □

Next, we show that the algorithm described in 3.6.1 returns $p(z, x|u)$ satisfying (3.34).

Proposition 6. For any given $p(x|u)$ for $u \in \mathcal{N}^2$ and $x \in \mathcal{N}$, the algorithm returns some distribution $p(z, x|u)$ such that
$$p(|Z| = \ell) = \bar{\theta}_\ell,\ \ell = 1, \ldots, n. \tag{3.54}$$

Proof. See Appendix B.7 □

3.7 An Outer Bound

In this section, we will show that any ON-OFF privacy scheme must satisfy $R_t \leq R_t^O$.

First, we define an auxiliary random variable Y_t taking values in $\mathscr{P}(\mathcal{N})$ based on the decodability of the subset of messages. Specifically, let Y_t be a function of Q_t such that $Y_t = \mathcal{D}$ for $\mathcal{D} \in \mathscr{P}(\mathcal{N})$ if the user may decode the messages $W_{\mathcal{D}}$ but not any message W_i for $i \in \mathcal{N} \backslash \mathcal{D}$ from the answer A_t. Roughly speaking, Y_t represents the capability of decoding messages from the query Q_t. Note that since the query Q_t and messages $W_{\mathcal{N}}$ are independent, the decodability of any message is known by the server only through Q_t, that is, Y_t is a function of Q_t. In this way, the alphabet $\mathcal{Q}$ (may be infinite if the query is coded) of the query is partitioned into 2^n classes based on the decodability of the subset of the messages. Clearly, from the definition of Y_t, we have that the length of the answer $\ell(Q_t)$ satisfies

$$\ell(Q_t) \geq |Y_t| \, L,$$

since the answer A_t is at least of length $|Y_t| \, L$ if the user can decode $|Y_t|$ messages from the answer A_t. Hence, the download cost ℓ_t is bounded by

$$\ell_t \geq \mathbb{E}\left[|Y_t|\right] \, L. \tag{3.55}$$

Next, we start to reinterpret the privacy and the decodability constraints in terms of the auxiliary variable Y_t. By the definition of Y_t, the decodability can be written as

$$p\left(x_t, y_t\right) = 0, \forall x_t \notin y_t, \tag{3.56}$$

where $x_t \in \mathcal{N}$ and $y_t \in \mathscr{P}(\mathcal{N})$.

Recall the privacy constraint $I\left(X_{\overline{\mathcal{B}}_t}; Q_{\lfloor t \rfloor}\right) = 0$, and we must have

$$I\left(X_{\overline{\mathcal{B}}_t}; Q_{\lfloor t \rfloor}\right) \overset{\text{(a)}}{\geq} I\left(U_t; Q_t\right) \overset{\text{(b)}}{\geq} I\left(U_t; Y_t\right),$$

where (a) follows from $U_t = (X_\tau, X_{t+1}) \subset X_{\overline{\mathcal{B}}_t}$ and (b) follows because Y_t is a function of Q_t.

Thus, we can relax the privacy constraint by

$$I\left(U_t; Y_t\right) = 0. \tag{3.57}$$

For any given $p\left(x_t|u_t\right)$, if Y_t takes values in $\mathscr{P}\left(\mathcal{N}\right)$ and satisfies (3.56) and (3.57), then $\mathbb{E}\left[|Y_t|\right]$ is lower bounded by the following lemma.

Lemma 4. For any random variables U, X and Y, taking values in the alphabet $\mathcal{N}^2$, $\mathcal{N}$ and $\mathscr{P}\left(\mathcal{N}\right)$ respectively, if Y is independent of U, and $p(x, y|u) = 0$ for $x \notin y$, then

$$\mathbb{E}\left[|Y|\right] \geq \sum_{x \in \mathcal{N}} \max_{u \in \mathcal{N}^2} p\left(x|u\right). \tag{3.58}$$

Proof. See Appendix B.8. □

By substituting (3.58) in (3.55), we have

$$\ell_t \geq L \sum_{x_t \in \mathcal{N}} \max_{u_t \in \mathcal{N}^2} p\left(x_t|u_t\right). \tag{3.59}$$

Therefore, for any ON-OFF privacy scheme satisfying the decobability and privacy constraint, we know that the download cost is lower bounded by the right-hand side of (3.59). In other words, any ON-OFF privacy scheme must satisfy

$$\frac{1}{R_t} \geq \frac{1}{R_t^O} = \sum_{x_t \in \mathcal{N}} \max_{u_t \in \mathcal{N}^2} p\left(x_t|u_t\right) = \bar{\lambda}_m(t).$$

3.8 LP Formulation of Optimal Achievable Rate

In this section, we present an implicit characterization of the optimal rate, which is formulated by a linear program with an exponential number (in n) of variables and constraints.

As discussed in Section 3.5, the query design relies on solving the following linear program:

$$\begin{aligned} \underset{w(q|x,u)}{\text{minimize}} \quad & \mathbb{E}\left[|Q|\right] = \sum_{q} p\left(q\right)|q| \\ \text{subject to} \quad & p\left(x,q|u\right) = 0,\ x \notin q, \\ & p\left(q|u\right) = p\left(q\right), \end{aligned} \tag{3.60}$$

where $x \in \mathcal{N}$, $u \in \mathcal{N}^2$, $q \in \mathscr{P}(\mathcal{N})$ and probabilities $p\left(x|u\right)$ are given. We know that any feasible solution to the above problem yields an achievable scheme. In other words, the rate R_t is achievable if

$$\frac{1}{R_t} \geq C_1^*,$$

where C_1^* is the optimal value to (3.60).

On the other hand, one may notice that the key lemma, i.e., Lemma 4, to show the outer bound, indeed indicates that any achievable scheme must satisfy that

$$\frac{1}{R_t} \geq C_2^*,$$

where C_2^* is the optimal value to the following problem:

$$\begin{aligned} \underset{w(y|x,u)}{\text{minimize}} \quad & \mathbb{E}\left[|Y|\right] = \sum_{y} p\left(y\right)|y| \\ \text{subject to} \quad & p\left(x,y|u\right) = 0,\ x \notin y, \\ & p\left(y|u\right) = p\left(y\right), \end{aligned} \tag{3.61}$$

where $x \in \mathcal{N}$, $u \in \mathcal{N}^2$, $y \in \mathscr{P}(\mathcal{N})$ and probabilities $p\left(x|u\right)$ are given by the Markov chain.

Although problems (3.60) and (3.61) have different physical meanings, it is easy to see that they have the same optimal value, i.e., $C_1^* = C_2^*$. Therefore, by letting C_t be the optimal value to both problems, the achievable region can be fully characterized by

Corollary 4. The rate tuple $(R_t : t \in \mathbb{N})$ is achievable if and only if $R_t \leq C_t$.

However, it is notable that Corollary 4 is an *implicit* characterization, because as we discussed, the exponential blow-up of the number of variables and constraints makes the linear programming

problem intractable.

Remark 3 (Window size ω)**.** From our earlier discussion, we know that the feasible region of (3.60) denotes schemes that only require a window of size $\omega = 1$. Although we have assumed that the user knows the future requests within a window of positive size ω, increasing the window size into the future beyond $\omega = 1$ does not, in fact, increase the rate. Intuitively, this phenomenon stems from the Markov assumption we use to model the user's requests. If the window size $\omega = 0$, i.e., no future requests are known, the privacy defined in (3.4) has to be relaxed, and only past requests can be protected, which is the model studied in Chapter 2.

In this work, we actually proposed an explicit scheme to the ON-OFF privacy problem by finding a feasible solution that might not be optimal, to (3.60), which is of polynomial time complexity. Moreover, we show that our scheme is optimal for some cases in Corollary 2 and Corollary 3.

CHAPTER 4

PRIVATE MULTI-GROUP AGGREGATION

4.1 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 users. 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 [49] or, more recently, federated learning [4, 50–52].

In this work, we focus on the setting depicted in Figure 4.1, in which users belong to different groups. The server wants to find the aggregate for each group separately. As opposed to finding the aggregate over the whole population, as in typical distributed aggregation problems, e.g., [4, 18]. The users' groups can be based, for example, on their political views, immigration status, health condition, or race, to name a few. This raises additional privacy concerns since participating users may be rightfully wary of revealing their group.

Consider, for example, medical or clinical trials conducted to determine how having a certain illness, say diabetes, affects the efficacy of a new vaccine. A volunteer may want to contribute his vaccine test results, but does not want to reveal his medical condition (diabetes), i.e., the group

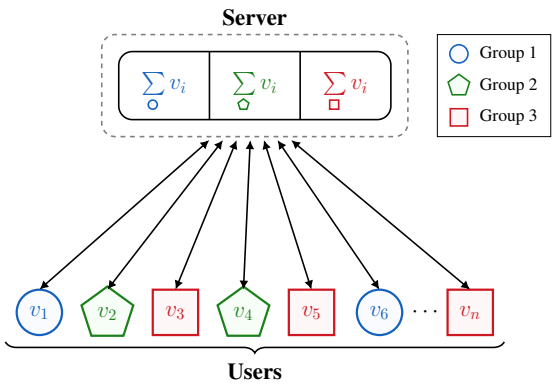


Figure 4.1: An instance of the Private Multi-Group Aggregation problem with n users. Each user i, for $i \in \{1, \ldots, n\}$, has a scalar value, v_i, and belongs to one of the $k = 3$ distinct groups. The server's goal is to estimate the sum of the values in each of the groups.

he belongs to. Another application is population polling during elections, where pollsters want to estimate how different groups of the population vote. Such groups could depend on race, gender, age, or income bracket. The poll participants want to indicate which political candidate they will vote for while keeping their group private.

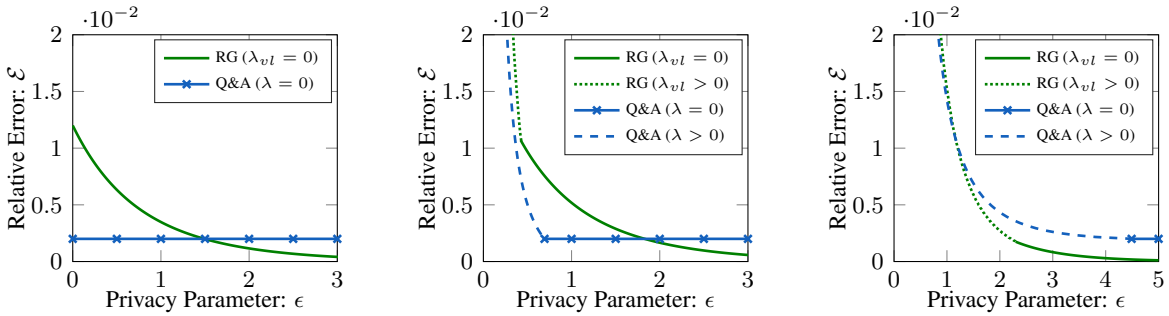
Motivated by these examples, we present the problem of Private Multi-Group Aggregation (PMGA), where local differential privacy [2, 23] guarantees are given over a user's group. We are interested in schemes that scale well with the number of groups since more groups allow the server more refined statistics about the population. Our main objective is to design schemes with low communication costs per user, 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 estimator's mean square error.

4.1.1 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 users' data (values). Differentially private schemes and bounds for private aggregation were studied in [18–22]. In [4], secure aggregation based on information-theoretic (secret sharing) and cryptographic techniques was developed for applications to federated learning (FL) [8]. Secure aggregation algorithms for FL with improved communication and computation overhead were proposed in [15, 16], and with robustness against adversarial users in [17]. These schemes have a per-user communication cost that grows with the number of users.

Although in this chapter we focus on estimating the sum, other works have focused on different estimation problems. For instance, distributed empirical mean estimation under communication constraints has been looked at in [53, 54]. Beyond estimating the mean, discrete distribution estimation under communication constraints has been studied in [55, 56], and under privacy constraints in [57–62]. Moreover, heavy hitters (most frequent items) estimation has been studied in [63, 64] under privacy and communication constraints. Recent work in [65] devises schemes that achieve optimal privacy and communication for mean and frequency estimation.

Another related problem is federated submodel learning [66–68]. In this setting, one or multiple servers hold various submodels (vectors) and each user wants to train (update) a private subset of



(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 $p_2(1) = 0.01$.

Figure 4.2: 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 has more users. The sub-figures (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 an 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. We present a more detailed comparison in Section 4.6.

these submodels. The notion of submodels in this setting 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 [67, 68] 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 [66] 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.

4.1.2 Contributions

We introduce the problem of private multi-group aggregation (Figure 4.1), where n users communicate with a central server. Each user holds a value and belongs to a private group. The goal is for the server to accurately compute the sum of values per group while keeping the user's group private. The notion of privacy we use is local differential privacy.

Our main contribution is 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 can be done offline since it does not depend on a user's group and value. Thus, the online user-to-server communication cost does not depend on the number of groups and users, as typically occurs in secure aggregation problems, e.g., [4]. In Theorem 5, we characterize 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 [24] 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 4.2.

4.1.3 Organization

The rest of the chapter is organized as follows. In Section 4.2, we describe the formulation of the Private Multi-Group Aggregation problem. In Section 4.3, we present our main results, which consist of the Query and Aggregate (Q&A) scheme and its performance (Theorem 5) compared to our proposed Randomized Group (RG) scheme. We present the details of the Q&A scheme in Section 4.4, and those of the RG scheme in Section 4.5. We compare the two schemes in Section 4.6.

4.1.4 Notation

We represent random variables by upper case letters, e.g., X, realizations of random variables by lower case letters, e.g., x, and the alphabets of the random variables by calligraphic letters, e.g., $\mathcal{X}$. We use $\log(x) = \log_2(x)$ and $\ln(x) = \log_e(x)$. Also, for any positive integer n, we denote $[n] := \{1, \ldots, n\}$. Moreover, we use a colon to refer to whole rows or columns in a matrix or vector. For instance, $A(:, i) = (A(1, i), A(2, i), \ldots, A(n, i))^\top$ is the i^{th} column of A. The L_2-norm of a vector $\mathbf{x}$ is denoted by $||\mathbf{x}||$.

4.2 Problem Formulation

We consider the setting depicted in Figure 4.1 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 $i \in [n] := \{1, \dots, n\}$ belongs to one of k groups, indexed from 1 to k. Moreover, user $i \in [n]$ holds a value $v_i \in \mathcal{V} := \{\pm 1, \dots, \pm m\}$. 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 faithfully participate in the scheme.

We denote by G_i the random variable representing the group that user i belongs to. We assume that the G_i, for all $i \in [n]$, are identical and independent random variables from the alphabet $\mathcal{G} = \{1, \dots, k\}$, where $k \geq 2$. 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 denote by $\theta := (\theta_1, \dots, \theta_k)$ the realization of the random vector Θ.

Each user i in group g holds an independent random scalar value V_i drawn from the alphabet $\mathcal{V} := \{\pm 1, \dots, \pm m\}$ according to the distribution $p_g(v) := \Pr(V_i = v | G_i = g)$. The values of the users in the same group are independent and identically distributed. We represent the users' value distributions by a $k \times 2m$ matrix

$$p := \begin{bmatrix} p_1(-m) & \dots & p_1(m) \\ \vdots & \ddots & \vdots \\ p_k(-m) & \dots & p_k(m) \end{bmatrix}.$$

The matrix p is unknown, to both the users and the server, and is assumed to be the realization of a random matrix 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 realizations 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

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

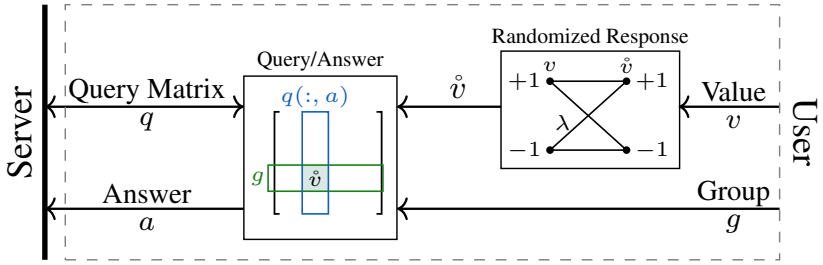


Figure 4.3: A block diagram representing the Q&A scheme for a binary alphabet, $\mathcal{V} = \{-1, 1\}$. The user is assigned a query matrix q. He sends the server an answer, a, which is an index of a column of this matrix. His answer is based on his group, g, and his randomized value, $\mathring{v}$. To randomize his value, he applies randomized response, parameterized by λ.

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; and (ii) server-centric, that measures the total communication the server receives from all the users. We refer to the latter as the *total communication cost*.

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

$$\mathcal{E}_\pi := \frac{1}{n^2}\text{MSE}(\hat{\mathbf{S}}_\pi), \tag{4.1}$$

where $\text{MSE}(\hat{\mathbf{S}}_\pi) = \mathbb{E}\left[||\hat{\mathbf{S}}_\pi - \mathbf{S}||^2 \middle| P = p, \Theta = \theta\right]$. For ease of notation, the conditioning on P and Θ is implicit in the rest of the chapter. The relative mean square error captures the accuracy of our estimate relative to the expected true aggregate $\mathbf{S}$. Since $\mathbb{E}\left[||\mathbf{S}||^2\right]$ grows as $\mathcal{O}(n^2)$, we normalize by n^2.

(iii) ***Privacy:*** We keep a user's group private. We use local differential privacy [2, 23] as our measure of privacy for a user's group. Since a user's value and group can be correlated, it is sometimes necessary (depending on the required privacy parameter) to also hide a user's value in addition to his group. To that end, a user's answer to the server is 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 group, value, query and local randomness.

Definition 5. 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 $a \in \mathcal{A}$,

$$\Pr(\mathcal{M}(G,V,Q) = a | G = g, Q = q, P = p, \Theta = \theta) \leq$$
$$e^{\epsilon} \Pr(\mathcal{M}(G,V,Q) = a | G = g', Q = q, P = p, \Theta = \theta), \quad (4.2)$$

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 might infer information about the distributions of P and Θ.

We note that if a randomized mechanism is ϵ_0 locally differentially private, then it is also locally differentially private for all $\epsilon > \epsilon_0$ motivating the following definition.

Definition 6. The privacy level of a scheme (randomized mechanism) is the smallest $\epsilon > 0$ such that (4.2) is satisfied.

4.3 Main Results

Query and Aggregate (Q&A) Scheme: We propose a new scheme for PMGA, which we refer to as the Query and Aggregate (Q&A) scheme. Q&A is characterized by its low communication cost per user, which is independent of the number of groups k and number of users n. It also offers an advantageous accuracy for the high privacy regime. Figure 4.3 summarizes this scheme, which mainly consists of two blocks:

1. *Query/Answer block:* The user is assigned a query matrix[1]. His answer is an index of a column of this matrix, and is determined by his value and group. This leverages the randomness

[1]The assigned query matrix, q, is independent of the user's group and value, and is known to both the server and the user.

in the user's value to hide his group and already provides a level of privacy over the user's group.

2. *Randomized Response block:* Here, the user adds noise to his value parameterized by $\lambda \geq 0$. This block is not always necessary, except for some cases, such as when the users' groups and values are highly correlated.

Theorem 5 characterizes the communication cost, privacy, and accuracy achieved by the Q&A scheme.

Theorem 5 (Q&A Scheme)**.** Given a PMGA instance with n users, k groups, alphabet $\mathcal{V} = \{\pm 1, \ldots, \pm m\}$, 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

$$e^{\epsilon_{\text{QA}}} = \max_{\substack{v,v' \in \mathcal{V}, \\ g,g' \in \mathcal{G}, g' \neq g}} \left\{ \frac{(2m(1-\lambda)-1)p_g(v) + \lambda}{(2m(1-\lambda)-1)p_{g'}(v') + \lambda} \right\}, \tag{4.3}$$

   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 relative mean square error

$$\mathcal{E}_{\text{QA}} = \alpha n^{-1}, \tag{4.4}$$

   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 relative mean square error is $\mathcal{O}\left(\frac{km^4}{n}\right)$.

We explain the Q&A scheme in more details in Section 4.4.

Randomized Group (RG) Scheme: To better understand the performance of the Q&A scheme described in Theorem 5, we compare it to the Randomized Group (RG) scheme which adds noise directly to the group. With probability λ_{gr}, the user sends the server his true group ($\log(k)$ bits) and true value or a noisy version of it ($\log(2m)$ bits). Otherwise, the user lies about his group and sends a mean zero random value that is independent of his true value.

This scheme is an adaptation of the randomized response [24, 57] method used in the differential privacy literature. In Theorem 6 in Section 4.5 we present the details and analysis of the RG scheme.

Comparison (Q&A vs. RG): The Q&A scheme requires $\log(2m)$ bits per user, while the RG scheme requires $\log(2km)$ bits per user. Therefore, from a user-centric perspective, the Q&A scheme always outperforms the RG scheme in terms of communication cost. However, they achieve different error and privacy trade-offs. We also look at the communication cost from a server-centric perspective by fixing the total communication cost at the server, and comparing the relative error versus privacy. This allows for a different number of users for each of the two schemes.[2]

Figure 4.2, gives an instance of this comparison for a fixed communication cost. The key take-away from this comparison is that there are two regimes, *(i) a high privacy regime* where for small values of the privacy parameter, ϵ, Q&A outperforms RG; *(ii) a low privacy regime* where for large enough privacy parameter, ϵ, RG outperforms Q&A. This is because, as ϵ goes to infinity, the error of the Q&A scheme converges to a constant strictly bounded away from zero as we cannot further tune the parameters of the scheme. On the other hand, the error of the RG scheme converges to zero. We defer a more detailed comparison to Section 4.6.

4.4 The Query and Aggregate (Q&A) Scheme

In this section, we describe the Q&A scheme. We begin by an example that illustrates the key ideas of this scheme by focusing on the special case of two groups and a binary alphabet. We then give the description of the general (Q&A) scheme in Section 4.4.2.

4.4.1 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

The scheme is composed of the following three steps.

[2] This is motivated by the idea that, in practice, the server might be choosing a batch of users from a larger population.

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 only looks 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 1-bit answer into the vector $q_i(:, a_i)$, 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 as follows:

$$\hat{\mathbf{S}}_{\text{QA}} = \sum_{i=1}^{n} q_i(:, a_i). \tag{4.5}$$

For example, consider a user i in group 2 who has value $+1$. If he receives the query $q_i = \left[\begin{smallmatrix} -1 & +1 \\ +1 & -1 \end{smallmatrix}\right]$, then his answer is $a_i = 1$, which the server maps into the vector $q_i(:, a_i) = \left[\begin{smallmatrix} -1 \\ +1 \end{smallmatrix}\right]$. Otherwise, if the user receives the query $q_i = \left[\begin{smallmatrix} -1 & +1 \\ -1 & +1 \end{smallmatrix}\right]$, then his answer is $a_i = 2$, which is mapped into $q_i(:, a_i) = \left[\begin{smallmatrix} +1 \\ +1 \end{smallmatrix}\right]$.

The key idea behind these queries is that they provide different, and equally likely, pairings of a value for a particular group with all possible values of the other group. For instance, if we look at the first column of the query matrices, notice that in the query $\left[\begin{smallmatrix} -1 & +1 \\ +1 & -1 \end{smallmatrix}\right]$, the value -1 for group 1 (first row) is paired with the value $+1$ of group 2 (second row), while in query $\left[\begin{smallmatrix} -1 & +1 \\ -1 & +1 \end{smallmatrix}\right]$ it is paired with the value -1 of group 2.

Next we give a brief analysis of this scheme, and see how it fairs on our three performance metrics: accuracy (MSE), privacy, and communication cost.

Accuracy

We show that the relative 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, its estimate is

$$\begin{aligned}\hat{\mathbf{S}}_{\text{QA}}(1) &= \sum_{i\in[n]:g_i=1} q_i(1,a_i) + \sum_{i\in[n]:g_i=2} q_i(1,a_i) \\ &= \underbrace{\mathbf{S}(1)}_{\substack{\text{True Aggregate} \\ \text{for Group 1}}} + \underbrace{\sum_{i\in[n]:g_i=2} q_i(1,a_i)}_{\text{Noise}}. \end{aligned} \tag{4.6}$$

Therefore, the estimate $\hat{\mathbf{S}}_{\text{QA}}(1)$ 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,

$$Q_i(1,A_i) = \begin{cases} -1 & \text{with probability } \frac{1}{2}, \\ +1 & \text{with probability } \frac{1}{2}. \end{cases} \tag{4.7}$$

The noise term can be interpreted as the position of a point on the integer number line, $\mathbb{Z}$, after n steps of a simple random walk starting at zero. Alternatively, 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 relative mean square error, $\mathcal{E}_{\text{QA}}$, goes to zero as $\mathcal{O}(n^{-1})$.

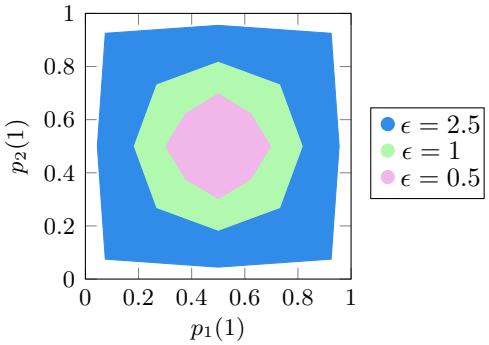


Figure 4.4: Privacy for the Q&A Scheme. The values of $p_1(1)$ and $p_2(1)$ in the shaded regions of the figures above guarantee the fixed privacy parameters $\epsilon = 2.5$, $\epsilon = 1$, and $\epsilon = 0.5$, respectively. The higher the privacy requirement, i.e., smaller ϵ, the smaller the region. We note that the indicated region is the full interior of the polygon.

Privacy

We show that the Q&A scheme is ϵ_{QA} locally differentially private. From Definition 5,

$$e^{\epsilon_{\text{QA}}} = \max_{\substack{g,g' \in \{1,2\}, \\ a \in \{1,2\}, q \in \mathcal{Q}}} \frac{\Pr(A_i = a | G_i = g, Q_i = q)}{\Pr(A_i = a | G_i = g', Q_i = q)}. \tag{4.8}$$

The first thing we notice is that the ratio in (4.8) 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 (4.8) 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 (4.8) to

$$e^{\epsilon_{\text{QA}}} = \max_{\substack{g,g' \in \{1,2\}, g \neq g' \\ v,v' \in \{-1,1\}, q \in \mathcal{Q}}} \frac{\Pr(V_i = v | G_i = g, Q_i = q)}{\Pr(V_i = v' | G_i = g', Q_i = q)} = \max_{\substack{g,g' \in \{1,2\}, g \neq g' \\ v,v' \in \{-1,1\}}} \frac{p_g(v)}{p_{g'}(v')}, \tag{4.9}$$

which follows from the independence of the random variables representing the user's value, V_i, and his assigned query, Q_i, and the definition $p_g(v) = \Pr(V_i = v | G_i = g)$.[3] Thus, we obtain an expression of the privacy which only depends on the users' value distributions.

We refer to the privacy parameter ϵ_{QA}, described in (4.9), as the *intrinsic privacy* of the scheme. Notice that it depends on the users' value distributions, $p_1(\cdot)$ and $p_2(\cdot)$; however, neither the server nor the users know these $p_1(\cdot)$ and $p_2(\cdot)$. Therefore, they cannot directly calculate the privacy

[3]To simplify our discussion, in the rest of this chapter, we assume that the probabilities $p_g(v)$ are in $(0, 1)$, for all $g \in [k]$ and $v \in \mathcal{V}$.

parameter ϵ_{QA}. Nonetheless, the privacy parameter, ϵ_{QA}, can be bounded if the users have prior information about their value distributions. For example, suppose the users know that $p_1(1)$ and $p_2(1)$ are bounded such that $c_{\min} \leq p_1(1), p_2(1) \leq c_{\max}$, where the constants $c_{\min}, c_{\max} \in (0,1)$. In this case, we can upper bound the intrinsic privacy level $\epsilon_{\text{QA}} \leq \ln\left(\frac{c_{\max}}{c_{\min}}\right)$.

Next we give more insights about the relationship between the users' value distributions, $p_g(\cdot)$, and the privacy parameter. Let us fix a privacy level ϵ, and define the region that describes the users' value distributions, $p_1(1), p_2(1) \in (0,1)$, which guarantee that the scheme is ϵ-LDP. In Figure 4.4, we plot this region for different values of ϵ. Looking at Figure 4.4 and (4.9), we observe the following.

- The less privacy we require, i.e., the larger the privacy level ϵ, the larger the highlighted region, i.e., more values of $p_1(1)$ and $p_2(1)$ can guarantee this level of privacy.
- The closer $p_1(1)$ and $p_2(1)$ are to 0.5, the higher the privacy guarantee. And perfect privacy, i.e., $\epsilon = 0$, is only guaranteed when $p_1(1) = p_2(1) = 0.5$. Intuitively, this occurs because when $p_1(1) = p_2(1) = 0.5$, a user's answer to the query is independent of his group.

A takeaway from the above observations is that not all privacy levels can be guaranteed for fixed user value distributions $p_1(\cdot)$ and $p_2(\cdot)$. In other words, the intrinsic privacy of the scheme may not always be enough. The reason is that, in its basic form, the Q&A scheme described above, does not guarantee privacy over the user's value. Thus, when the user's value and group are sufficiently correlated, the user's value might leak more information about his group than permitted by the ϵ-LDP requirement. In such cases, the general Q&A scheme adds a second layer of privacy to the user's value to further hide his group. In addition, this provides flexible privacy guarantees which do not depend only on the user's value distributions. This second layer of privacy is obtained by adding a randomized response block, parameterized by the probability of lying λ, which hides a user's value (see Figure 4.3). We give a full description of the general Q&A scheme in Section 4.4.2.

Communication

Since the user's answer is either 1 or 2, i.e., $a_i \in \{1, 2\}$, the scheme's communication cost is one bit per user. Moreover, we show in Theorem 5 that the general scheme's communication cost is

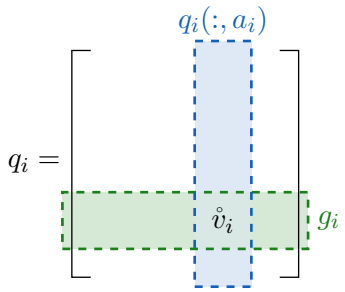


Figure 4.5: User i sends the answer a_i based on his assigned query matrix q_i, his group g_i, and his randomized value $\mathring{v}_i$. His answer is the index of the column that contains his randomized value. The server maps this answer to the a_i^{th} column of q_i.

always 1 bit per user when the alphabet, $\mathcal{V}$, 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.

4.4.2 The General Q&A Scheme

In this section, we describe the general Q&A scheme, for any number of groups $k \geq 2$, and alphabet parameter $m \in \mathbb{N}^+$. This scheme is obtained by generalizing the query matrices of the previous example and is presented in Figure 4.5. The Q&A scheme includes an additional randomized response block for improved privacy as described in Figure 4.3.

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

$$\mathcal{Q} := \left\{ q \in \mathcal{V}^{k \times 2m} \middle| q(g, :) \in \mathbf{Sym}(\mathcal{V}) \text{ for all } g \in [k] \right\}, \tag{4.10}$$

where $q(g, :) = (q(g, 1), \ldots, q(g, k))$ and $\mathbf{Sym}(\mathcal{V})$ is the set of all row vectors which are an ordered permutation of the finite set $\mathcal{V}$.[4] Each row of a matrix $q \in \mathcal{Q}$ is a permutation of all the possible $2m$

[4] An ordered permutation of a set $\mathcal{V}$ is a vector where each element is a distinct element of $\mathcal{V}$, e.g., $\mathbf{Sym}(\{\pm 1, \pm 2\})$

values. Notice that the values cannot be repeated within a row but rows can be repeated. We denote by q_i the query assigned to user i. More details about the choice of set (4.10) are in Appendix C.1.2.

We assume 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 can 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 hides his value using the randomized response block parameterized by λ, as in Figure 4.3. That is, given his true value v_i, the user first chooses a randomized value $\mathring{v}_i$ according to the distribution

$$\Pr(\mathring{V}_i = \mathring{v}_i | V_i = v_i) = \begin{cases} 1-\lambda & \text{for } \mathring{v}_i = v_i \\ \frac{\lambda}{2m-1} & \text{for } \mathring{v}_i \in \mathcal{V} - \{v_i\}, \end{cases} \tag{4.11}$$

where $\lambda \in [0, 1 - \frac{1}{2m})$.When $\lambda = 0$, i.e., no privacy over the user's value, then $V_i = \mathring{V}_i$.

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 $q_i(g_i, a_i) = \mathring{v}_i$ as explained in Figure 4.5.

Remark 4 (On Sending Columns and not Rows)**.** If the user's answer corresponded to a row of the query matrix, instead of a column, then the server would obtain full information about the user's group and none about his value, i.e., the scheme would fail in providing both privacy and utility.

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 mapped answers from all the users, and multiplies by an unbiasing term (see Appendix C.1.1 for more details), to find the estimate of the true aggregate, $\mathbf{S}$, i.e.,

$$\hat{\mathbf{S}}_{\text{QA}} = \frac{2m-1}{2m - 2m\lambda - 1} \sum_{i \in [n]} q_i(:, a_i). \tag{4.12}$$

Below we give examples of possible queries and answers.

has $4! = 24$ elements including $(-2, -1, 1, 2)$ and $(1, 2, -2, -1)$.

Example 3. 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., $V_i = \mathring{V}_i$. Suppose that 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., $q_1(:, a_1) = (+1, -1, -2)^\top$.

If the user is assigned the query

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

his answer will be $a_1 = 4$, which the server decodes into $q_1(:, a_1) = (+2, -1, -1)^\top$. In both cases $q_1(g_1, a_1) = v_1$.

We note a few characteristics of this design of queries and answers. Since every row of any query matrix $q \in \mathcal{Q}$ contains all possible values, the user's value is always one of the elements of the row vector corresponding to his group. Moreover, from the server's perspective, looking at the mapped answer $q_i(:, a_i)$, i.e., a column vector of the user's assigned query q_i, the user's value (or randomized value) is in row g_i of q_i. As for all the other elements of the vector, they are uniformly distributed over $\mathcal{V}$. This follows from the design of the query alphabet $\mathcal{Q}$ and mirrors (4.7) from the previous section. It is also the key idea for the accuracy proof of Theorem 5.

An interesting property of the Q&A scheme is that, depending on the required privacy, one can choose $\lambda = 0$, i.e., no randomized response block in Figure 4.3. The Q&A scheme still guarantees local differential privacy with

$$\epsilon_{\text{QA}} = \ln \left(\max_{\substack{v,v' \in \mathcal{V} \\ g,g' \in \mathcal{G}, g' \neq g}} \left\{ \frac{p_g(v)}{p_{g'}(v')} \right\} \right), \tag{4.13}$$

which follows from (4.3). As in the previous section, we refer to this as the *intrinsic privacy* of the scheme, which corresponds to the special case of $\lambda = 0$. If the intrinsic privacy is not enough

because of a high correlation between the user's group and value, external noise can be added to the values through the randomized response block with λ chosen appropriately depending on the required privacy ϵ.

Remark 5 (The choice of λ)**.** Given a required privacy parameter ϵ, the parameter λ that can guarantee this given ϵ is determined using (4.3). However, this requires the knowledge of the value distributions, $p_g(.)$ for all $g \in \mathcal{G}$. Nevertheless, one can still use (4.3) to find a bound on λ that is independent of the users' value distributions as follows,

$$\lambda \geq \frac{2m-1}{2m+e^{\epsilon}-1}. \tag{4.14}$$

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

$$\lambda \geq \frac{(2m-1)c_{\max} - c_{\min}e^{\epsilon}}{2m(c_{\max} - c_{\min}e^{\epsilon}) + e^{\epsilon} - 1}. \tag{4.15}$$

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

Remark 6 (Error Calculation)**.** Computing the mean square error relies on generalizing the approach in the example in Section 4.4.1. We have two types of errors in the estimate of the aggregate per group, i.e., $\hat{\mathbf{S}}_{\mathrm{QA}}(g)$. The first is the error introduced by the users that are not in group g. This can be approximated by a zero mean additive noise as shown in (4.6). The second is the error introduced by the randomized response block acting on the users' value. This error biases the sum $\sum q_i(:, a_i)$. Therefore, to unbias the estimator we multiply by $(2m-1)/(2m-2m\lambda-1)$ as seen in (4.12). The details of the error calculation can be found in the proof in Appendix C.1.1.

Theorem 5, first stated in Section 4.3, provides the performance of the Q&A scheme with respect to communication cost, privacy, and accuracy. For its proof see Appendix C.1.1. Moreover, we elaborate on our choice of (4.10) and (4.11) for the Q&A scheme in Appendix C.1.2.

4.5 The Randomized Group (RG) Scheme

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

$$\Pr(\mathring{G}_i = \mathring{g}_i | G_i = g_i) = \begin{cases} 1 - \lambda_{gr} & \text{for } \mathring{g}_i = g_i \\ \frac{\lambda_{gr}}{k-1} & \text{for } \mathring{g}_i \in [k] - \{g_i\}, \end{cases} \tag{4.16}$$

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.,

$$\Pr(\mathring{V}_i = \mathring{v}_i | \mathring{g}_i \neq g_i) = \frac{1}{2m} \tag{4.17}$$

   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

$$\Pr(\mathring{V}_i = \mathring{v}_i | V_i = v_i, \mathring{g}_i = g_i) = \begin{cases} 1 - \lambda_{vl} & \text{for } \mathring{v}_i = v_i, \\ \frac{\lambda_{vl}}{2m-1} & \text{for } \mathring{v}_i \in \mathcal{V} - \{v_i\}. \end{cases} \tag{4.18}$$

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.

Theorem 6 characterizes the scheme's performance with respect to communication cost, privacy, and accuracy.

Theorem 6. Given a PMGA instance with n users, k groups, alphabet $\mathcal{V} = \{\pm 1, \ldots, \pm m\}$, and the users' value distribution $p_g(v)$ for all $g \in \mathcal{G}, v \in \mathcal{V}$; the Randomized Group scheme (RG) is parameterized by the randomization parameters $\lambda_{gr} \in (0, 1)$, and $\lambda_{vl} \in \left[0, 1 - \frac{1}{2m}\right)$ and satisfies the following properties.

1. The RG scheme has a communication cost of $\log(2km)$ bits per user.

2. The RG scheme is ϵ_{RG}-LDP with

$$e^{\epsilon_{\text{RG}}} = \max\left\{\beta_1(p_{\max}\beta_2 + \lambda_{vl}), \frac{1}{\beta_1(p_{\min}\beta_2 + \lambda_{vl})}\right\}, \tag{4.19}$$

   where $p_{\max} = \max_{g\in\mathcal{G}, \in v\in\mathcal{V}} p_g(v)$, $p_{\min} = \min_{g\in\mathcal{G}, \in v\in\mathcal{V}} p_g(v)$, $\beta_1 = \frac{2m(k-1)(1-\lambda_{gr})}{(2m-1)\lambda_{gr}}$, and $\beta_2 = (2m(1-\lambda_{vl}) - 1)$.

3. The estimator of the RG scheme is unbiased and has relative mean square error

$$\mathcal{E}_{\text{RG}} = \beta_3 n^{-1}, \tag{4.20}$$

   where $\beta_3 = \mathbb{E}[V_1^2]\left(\frac{2m-1}{(1-\lambda_{gr})(2m(1-\lambda_{vl})-1))} - 1\right) + \frac{(4m^2-1)(m+1)(2m\lambda_{vl}(1-\lambda_{gr})+\lambda_{gr}(2m-1))}{6(1-\lambda_{gr})^2(2m(1-\lambda_{vl})-1)^2}$. The relative mean square error is $\mathcal{O}\left(\frac{m^4k^2}{ne^\epsilon}\right)$.

Proof. See Appendix C.2.1. □

The following corollary characterizes the relationship between the randomization parameters, λ_{gr}, λ_{vl}, and the privacy parameter, ϵ_{RG}.

Corollary 5. Let $\epsilon > 0$ be the required privacy parameter. Then, the optimal parameters λ_{gr}^* and λ_{vl}^* which guarantee the required privacy and minimize the relative error of the RG scheme are given below.

- If $e^{2\epsilon} < \frac{p_{\max}}{p_{\min}}$ and $p_{\max} \neq \frac{1}{2m}$, then

$$\lambda_{vl}^* = \frac{(2m-1)(p_{\max} - e^{2\epsilon}p_{\min})}{(2p_{\max}-1)+(1-2p_{\min})e^{2\epsilon}}$$

  and

$$\lambda_{gr}^* = \frac{2m(k-1)(p_{\max}-p_{\min})e^\epsilon}{2m(k-1)(p_{\max}-p_{\min})e^\epsilon+(1-2p_{\min})e^{2\epsilon}+2p_{\max}-1}.$$

- If $e^{2\epsilon} \geq \frac{p_{\max}}{p_{\min}}$ or $p_{\max} = \frac{1}{2m}$, then $\lambda_{vl}^* = 0$ and

$$\lambda_{gr}^* = \frac{2m(k-1)p_{\max}}{2m(k-1)p_{\max} + e^{\epsilon}}.$$

Proof. See Appendix C.3. □

The above Corollary describes the choice of parameters λ_{vl} and λ_{gr} that minimize the error for a given required privacy $\epsilon > 0$. It also shows that for a high privacy requirement, i.e., privacy parameter $\epsilon < \frac{1}{2} \ln \left(\frac{p_{\max}}{p_{\min}} \right)$, the parameter λ_{vl} cannot be zero. Intuitively, since the user's value and group are correlated, applying a privacy preserving mechanism only over the group is not always enough. This is similar to what occurs with the Q&A scheme. In the RG scheme, the second layer of privacy, that hides the user's value, is characterized by the parameter λ_{vl}.

4.6 Comparison of the RG and Q&A Schemes

From the user-centric perspective, the Q&A scheme has a communication cost of $\log(2m)$ bits per user, i.e., it does not depend on the number of groups k. However, the communication cost of the RG scheme is $\log(2km)$ bits per user. Thus, the Q&A scheme can achieve a communication cost per user that the RG scheme cannot.

From the server-centric perspective, to compare the accuracy vs. privacy trade-offs of the two schemes, we fix the total communication cost, i.e., the number of bits communicated by all the users to the server. We choose the parameter λ of the Q&A scheme that guarantees the required (given) privacy parameter ϵ (see Remark 5) and minimizes the error. We find this parameter λ by solving the following optimization problem numerically

$$\begin{aligned} \underset{\lambda}{\text{minimize}} \quad & \mathcal{E}_{\text{QA}} \\ \text{subject to} \quad & e^{\epsilon} \leq \max_{\substack{v,v' \in \mathcal{V}, \\ g,g' \in \mathcal{G}, g' \neq g}} \left\{ \frac{(2m(1-\lambda)-1)p_g(v) + \lambda}{(2m(1-\lambda)-1)p_{g'}(v') + \lambda} \right\}, \\ & 0 \leq \lambda < 1 - \frac{1}{2m}, \end{aligned}$$

where $\mathcal{E}_{\text{QA}}$ is from (4.4). Similarly, for the RG scheme, we choose the parameters, λ_{gr} and λ_{vl}, as in Corollary 5.

Figure 4.2 illustrates this comparison for a fixed total communication cost. Typically for a high enough privacy constraint, i.e., small ϵ, Q&A outperforms RG, while for a low enough privacy constraint, RG outperforms Q&A. Thus, we have two privacy regimes, a high privacy regime where Q&A should be used, and a low privacy regime where RG should be used. These observations are made rigorous in Theorem 7.

We begin by expressing the relative mean square error as a function of the privacy parameter ϵ for a fixed total communication cost of $b \geq 2$ bits. Since the Q&A scheme's communication cost per user is $\log(2m)$ bits, its number of users is given by $n_{\text{QA}} := b/\log(2m)$. Analogously, the number of users for the RG scheme is given by $n_{\text{RG}} := b/\log(2km)$. [5] Therefore, we normalize each scheme's mean square error by its respective number of users (squared), obtaining

$$\mathcal{E}_{\text{QA}}(\epsilon, b) = \text{MSE}(\hat{\mathbf{S}}_{\text{QA}}) n_{\text{QA}}^{-2} = \text{MSE}(\hat{\mathbf{S}}_{\text{QA}}) \left(\tfrac{\log(2m)}{b}\right)^2, \tag{4.21}$$

and

$$\mathcal{E}_{\text{RG}}(\epsilon, b) = \text{MSE}(\hat{\mathbf{S}}_{\text{RG}}) n_{\text{RG}}^{-2} = \text{MSE}(\hat{\mathbf{S}}_{\text{RG}}) \left(\tfrac{\log(2km)}{b}\right)^2. \tag{4.22}$$

With this notation we present the following theorem.

Theorem 7. Let $\mathcal{V} = \{\pm 1, \ldots, \pm m\}$ be the alphabet of values, $\mathcal{G} = [k]$ the set of possible groups, and fix the total communication cost to $b \in \{x \in \mathbb{N}^+ | x/\log(2m), x/\log(2km) \in \mathbb{N}^+\}$. Unless $k = 2$, $m = 1$ and $p_1(v) = p_2(v') \neq 0.5$ for all $v, v' \in \{-1, 1\}$; then, there exists,

(i) an $\epsilon_h > 0$, such that for all $\epsilon < \epsilon_h$, the relative error $\mathcal{E}_{\text{QA}}(\epsilon, b) < \mathcal{E}_{\text{RG}}(\epsilon, b)$, and

(ii) an $\epsilon_\ell > 0$, such that for all $\epsilon > \epsilon_\ell$, the relative error $\mathcal{E}_{\text{QA}}(\epsilon, b) > \mathcal{E}_{\text{RG}}(\epsilon, b)$.

Proof. See Appendix C.3. □

For the special case of $k = 2$ groups, and a binary alphabet of values, i.e., $\mathcal{V} = \{-1, 1\}$, and $p_1(v) = p_2(v') \neq 0.5$ for some $v, v' \in \mathcal{V}$, there exists an $\epsilon_0 > 0$ such that for all $\epsilon < \epsilon_0$, the difference in relative errors $\mathcal{E}_{\text{QA}}(\epsilon_0, n_{\text{QA}}) - \mathcal{E}_{\text{RG}}(\epsilon_0, n_{\text{RG}}) = \frac{1}{b}$, where b is the total communication cost.

[5] We assume that the parameter b is chosen such that $n_{\text{QA}}, n_{\text{RG}} \in \mathbb{N}^+$.

Remark 7 (User-centric Accuracy vs. Privacy Trade-off)**.** Since both the RG and Q&A schemes have a fixed communication cost per user, restricting it can deem a scheme infeasible. To elaborate, consider the case of 2 groups and 2 values. If we fix the communication cost per user to 1 bit, the RG scheme cannot satisfy this constraint because it requires 2 bits of communication per user. Whereas the Q&A scheme directly satisfies the constraint. On the other hand, if we fix the communication cost per user to 2 bits, then the Q&A scheme would not utilize the full communication allotment.

CHAPTER 5
CONCLUSION

5.1 ON-OFF Privacy for Correlated Requests

In Chapters 2 and 3, we looked at the problem of turning privacy ON and OFF in an information retrieval setting when the user's interests are correlated over time. We modeled this correlation by a Markov chain with n states. We gave new achievable schemes with polynomial time complexity and a general upper bound on the achievable rate. We prove the optimality of our scheme for special cases, namely, a family of symmetric Markov chains and two-state Markov chains.

Future directions of this work include finding tighter outer bounds on the rate and efficient constructions of ON-OFF privacy schemes that would achieve them, i.e., characterizing the capacity for $n > 2$. Moreover, in this dissertation, we focus on the single-server setting. However, motivated by the literature on private information retrieval, we can consider multiple non-colluding servers and analyze the (possibly) reduced communication cost in the ON-OFF privacy setting.

Another natural direction is to investigate settings where the correlation in the user's requests has a different model than the Markov chain. For example, we can consider correlation models that account for trends, seasonality, and more.

We can also assume the server does not know the user's privacy status. Moreover, we can consider the user's privacy status (ON or OFF) as additional private information. Furthermore, we can explore models in which the user's requests and desired privacy status are correlated. It is interesting to analyze the consequences of these assumptions on the download cost.

Finally, in this dissertation, we assume the user's privacy choice is binary ON or OFF. However, one can consider the more general setting where the privacy requirement has more variation. Moreover, we can relax the privacy constraint from the information-theoretic perfect privacy. For instance, we can use differential privacy as a relaxed and varying (over time) privacy measure.

5.2 Private Multi-Group Aggregation

In Chapter 4, we formulated the problem of private multi-group aggregation where the goal was to privately aggregate the users' values per group. Moreover, we used local differential privacy as our measure of privacy for a user's group. We characterized two schemes: Query&Aggregate (Q&A) and Randomized Group (RG). The Q&A scheme generally outperformed the RG scheme, in terms of privacy vs. accuracy, in the high privacy regime.

Future work for this problem includes finding theoretic bounds characterizing the best performance achievable for a given privacy and total communication cost. Another direction would involve further reducing the communication cost. This can be done by mapping a larger alphabet of values $\mathcal{V}$ to a smaller alphabet $\mathcal{V}'$. In this dissertation, we looked at a communication cost that didn't grow with the number of groups. Still, it did grow logarithmically with the number of values, and one could consider a method that further limits this cost to a constant number of bits.

Finally, in the work presented in this dissertation, we assumed that all users behaved honestly. However, one could consider settings where adversarial or byzantine users aim to skew the aggregate. Because the added noise we consider is bounded, a natural connection exists between robustness against adversarial users and our differential privacy guarantees. Therefore, it would be interesting to see how byzantine users can affect the results of private multi-group aggregation.

Appendices

APPENDIX A

ON-OFF PRIVACY

A.1 Proof of Proposition 1

We need to show that $I\left(X_\tau; Q_t | Q_{[t-1]}\right) = 0$ implies that $I\left(X_{\mathcal{B}_t}; Q_t | Q_{[t-1]}\right) = 0$.

Consider

$$\begin{aligned} I\left(X_{\mathcal{B}_t}; Q_t | Q_{[t-1]}\right) &= I\left(X_\tau; Q_t | Q_{[t-1]}\right) + I\left(X_{\mathcal{B}_t \setminus \{\tau\}}; Q_t | X_\tau, Q_{[t-1]}\right) \\ &\overset{(a)}{\leq} I\left(X_\tau; Q_t | Q_{[t-1]}\right) + I\left(X_{\mathcal{B}_t \setminus \{\tau\}}; X_t, S_t | X_\tau, Q_{[t-1]}\right) \\ &\overset{(b)}{=} I\left(X_\tau; Q_t | Q_{[t-1]}\right) + I\left(X_{\mathcal{B}_t \setminus \{\tau\}}; X_t | X_\tau, Q_{[t-1]}\right), \end{aligned}$$

where (a) follows because Q_t is a function of $\{X_\tau, X_t, S_t, Q_{[t-1]}\}$, and (b) follows because the local randomness is generated according to $p_{X_t, X_\tau, Q_{[t-1]}}$.

It remains to show that $I\left(X_{\mathcal{B}_t \setminus \{\tau\}}; X_t | X_\tau, Q_{[t-1]}\right) = 0$, which can be justified as follows:

$$\begin{aligned} & I\left(X_{\mathcal{B}_t \setminus \{\tau\}}; X_t | X_\tau, Q_{[t-1]}\right) \\ &= H\left(X_{\mathcal{B}_t \setminus \{\tau\}} | X_\tau, Q_{[t-1]}\right) - H\left(X_{\mathcal{B}_t \setminus \{\tau\}} | X_t, X_\tau, Q_{[t-1]}\right) \\ &\leq H\left(X_{\mathcal{B}_t \setminus \{\tau\}} | X_\tau, Q_{[t-1]}\right) - H\left(X_{\mathcal{B}_t \setminus \{\tau\}} | X_t, X_\tau, Q_{[\tau-1]}, X_{[\tau:t-1]}, S_{[\tau:t-1]}\right) \\ &\overset{(c)}{=} H\left(X_{\mathcal{B}_t \setminus \{\tau\}} | X_\tau, Q_{[t-1]}\right) - H\left(X_{\mathcal{B}_t \setminus \{\tau\}} | X_t, X_\tau, Q_{[\tau-1]}, X_{[\tau:t-1]}\right) \\ &= H\left(X_{\mathcal{B}_t \setminus \{\tau\}} | X_\tau, Q_{[t-1]}\right) - H\left(X_{\mathcal{B}_t \setminus \{\tau\}} | X_\tau, Q_{[\tau-1]}, X_{[\tau+1:t]}\right) \\ &\overset{(d)}{=} H\left(X_{\mathcal{B}_t \setminus \{\tau\}} | X_\tau, Q_{[\tau-1]}\right) - H\left(X_{\mathcal{B}_t \setminus \{\tau\}} | X_\tau, Q_{[\tau-1]}\right) \\ &= 0, \end{aligned}$$

where (c) follows because $S_{[\tau:t-1]}$ is independent of $X_{\mathcal{B}_t \setminus \{\tau\}}$ given $\left\{X_{[\tau:t-1]}, Q_{[t-1]}\right\}$, and (d) follows from the markovity of $\{X_t : t \in \mathbb{N}\}$.

A.2 Proof of Corollary 1

Recall that the inequality that needs to be shown is

$$\sum_{q_{[t-1]}} p\left(q_{[t-1]}\right) \sum_{x_t} \max_{x_\tau} p\left(x_t|x_\tau, q_{[t-1]}\right) \geq \sum_{x_t} \max_{x_\tau} p\left(x_t|x_\tau\right). \tag{A.1}$$

Since

$$\sum_{q_{[\tau]}} p\left(q_{[\tau]}\right) \sum_{x_t} \max_{x_\tau} p\left(x_t|x_\tau, q_{[\tau]}\right) = \sum_{x_t} \max_{x_\tau} p\left(x_t|x_\tau\right),$$

where the equality follows because $Q_{[\tau]}$ is a stochastic function of $X_{[\tau]}$, and hence $Q_{[\tau]}$ is independent of X_t given X_τ, i.e.,

$$Q_{[\tau]} \to X_\tau \to X_t, \tag{A.2}$$

due to the Markovity of $\{X_i : i \in \mathbb{N}\}$. Thus, we can easily see that (A.1) holds for $t = \tau + 1$.

For any $i \in [\tau + 2 : t]$, consider

$$\begin{aligned}
&\sum_{q_{[i-1]}} p\left(q_{[i-1]}\right) \sum_{x_t} \max_{x_\tau} p\left(x_t|x_\tau, q_{[i-1]}\right) && \text{(A.3)}\\
&\geq \sum_{q_{[i-2]}} p\left(q_{[i-2]}\right) \sum_{x_t} \max_{x_\tau} \sum_{q_{i-1}} p\left(q_{i-1}|q_{[i-2]}\right) p\left(x_t|x_\tau, q_{[i-1]}\right)\\
&\overset{\text{(a)}}{=} \sum_{q_{[i-2]}} p\left(q_{[i-2]}\right) \sum_{x_t} \max_{x_\tau} \sum_{q_{i-1}} p\left(q_{i-1}|q_{[i-2]}, x_\tau\right) p\left(x_t|x_\tau, q_{[i-1]}\right)\\
&= \sum_{q_{[i-2]}} p\left(q_{[i-2]}\right) \sum_{x_t} \max_{x_\tau} p\left(x_t|x_\tau, q_{[i-2]}\right), && \text{(A.4)}
\end{aligned}$$

where (a) follows from the privacy at time $i - 1$.

Since (A.4) holds for any $i \in [\tau + 2 : t]$, we can easily obtain that

$$\begin{aligned}
&\sum_{q_{[t-1]}} p\left(q_{[t-1]}\right) \sum_{x_t} \max_{x_\tau} p\left(x_t|x_\tau, q_{[t-1]}\right)\\
&\qquad \geq \sum_{q_{[\tau]}} p\left(q_{[\tau]}\right) \sum_{x_t} \max_{x_\tau} p\left(x_t|x_\tau, q_{[\tau]}\right)\\
&\qquad = \sum_{x_t} \max_{x_\tau} p\left(x_t|x_\tau\right),
\end{aligned}$$

where the last step follows because of the Markov chain $Q_{[\tau]} \to X_\tau \to X_t$, as in (A.2).

This completes the proof.

A.3 Proof of Lemma 1

Consider

$$\begin{aligned}
\max_{u\in\mathcal{N}} p\left(x|u\right) &\overset{\text{(a)}}{=} \max_{u\in\mathcal{N}} \sum_{y:x\in y} p\left(x,y|u\right) \\
&= \max_{u\in\mathcal{N}} \sum_{y:x\in y} p\left(y|u\right) p\left(x|y,u\right) \\
&\overset{\text{(b)}}{=} \max_{u\in\mathcal{N}} \sum_{y:x\in y} p\left(y\right) p\left(x|y,u\right) \\
&\leq \sum_{y:x\in y} p\left(y\right) \max_{u\in\mathcal{N}} p\left(x|y,u\right) \\
&\leq \sum_{y:x\in y} p\left(y\right),
\end{aligned}$$

where (a) follows from $p(x,y) = 0$ for $x \notin y$, and (b) follows because Y is independent of U.

Thus, we obtain that

$$\begin{aligned}
\sum_{x\in\mathcal{N}} \max_{u\in\mathcal{N}} p\left(x|u\right) &\leq \sum_{x\in\mathcal{N}} \sum_{y:x\in y} p\left(y\right) \\
&= \sum_{y\in\mathscr{P}(\mathcal{X})} \sum_{x:x\in y} p\left(y\right) \\
&= \sum_{y\in\mathscr{P}(\mathcal{N})} p\left(y\right) \sum_{x:x\in y} 1 \\
&= \sum_{y\in\mathscr{P}(\mathcal{N})} p(y)|y| \\
&= \mathbb{E}\left[|Y|\right],
\end{aligned}$$

which completes the proof.

A.4 Justification of the Algorithm for Lemma 2

A.4.1 Verification of (2.57)

First, let us verify (2.57), i.e., for any $\ell = 1, \ldots, \sigma + 1$ and $x \in \mathcal{N}$,

$$\sum_{k=1}^{N} Q_{u^{(x,i)},k} \geq \min\left\{\delta_x, p\left(X = x | U = u^{(x,\ell)}\right)\right\} - p\left(X = x | U = u^{(x,\ell-1)}\right),$$

for all $i = 1, \ldots, \ell - 1$. Roughly speaking, the summation of the $u^{(x,i)}$-th row of Q should be larger than or equal to the right-hand side of (2.57) at any points when the algorithm update. Equivalently, for any given $\ell = 1, \ldots, \sigma + 1$ and $x, u \in \mathcal{N}$, if $u \in \left\{u^{(x,i)} : i = 1, \ldots, \ell - 1\right\}$, we need to verify that

$$\sum_{k=1}^{N} Q_{u,k} \geq \min\left\{\delta_x, p\left(X = x | U = u^{(x,\ell)}\right)\right\} - p\left(X = x | U = u^{(x,\ell-1)}\right), \tag{A.5}$$

From (2.54) and (2.56), it is clear that we subtract exactly the same value as the right-hand side of (2.57) from $\sum_{k=1}^{N} Q_{u,k}$ during each update. Therefore, by summing over x and ℓ, it is sufficient to show that for any given u, we have

$$\sum_{k=1}^{N} Q_{u,k} \geq \sum_{\ell=1}^{\sigma+1} \sum_{x \in \mathcal{N}: u \in \left\{u^{(x,i)}: i=1,\ldots,\ell-1\right\}} \min\left\{\delta_x, p\left(X = x | U = u^{(x,\ell)}\right)\right\} - p\left(X = x | U = u^{(x,\ell-1)}\right),$$

where $Q_{u,k}$ denotes the initializations in (2.51). To be precise, we re-write it by

$$\begin{aligned}
&\sum_{k=1}^{N} \max\left\{p\left(X = k | U = u\right) - \delta_k, 0\right\} \\
&\geq \sum_{\ell=1}^{\sigma+1} \sum_{x \in \mathcal{N}: u \in \left\{u^{(x,i)}: 1 \leq i \leq \ell-1\right\}} \min\left\{\delta_x, p\left(X = x | U = u^{(x,\ell)}\right)\right\} - p\left(X = x | U = u^{(x,\ell-1)}\right).
\end{aligned} \tag{A.6}$$

To establish (A.6), for a given $u \in \mathcal{N}$, let us suppose that

$$u = u^{(1,\alpha_1)} = \cdots = u^{(N,\alpha_N)}. \tag{A.7}$$

Then, the left-hand side of (A.6) can be written as

$$\sum_{k:\alpha_k \geq \sigma+1} \left(p\left(X = k | U = u\right) - \delta_k \right),$$

while the right-hand side of (A.6) can be written as

$$\begin{aligned}
&\sum_{x \in \mathcal{N}} \sum_{\ell : u \in \left\{ u^{(x,i)} : i = 1, \ldots, \ell - 1 \right\}} \min \left\{ \delta_x, p\left(X = x | U = u^{(x,\ell)}\right) \right\} - p\left(X = x | U = u^{(x,\ell-1)}\right) \\
&= \sum_{x : \alpha_x \leq \sigma} \sum_{\ell = \alpha_x + 1}^{\sigma+1} \min \left\{ \delta_x, p\left(X = x | U = u^{(x,\ell)}\right) \right\} - p\left(X = x | U = u^{(x,\ell-1)}\right) \\
&= \sum_{x : \alpha_x \leq \sigma} \sum_{\ell = \alpha_x + 1}^{\sigma+1} \left(\delta_x - p\left(X = x | U = u^{(x,\alpha_x)}\right) \right) \\
&= \sum_{x : \alpha_x \leq \sigma} \left(\delta_x - p\left(X = x | U = u\right) \right).
\end{aligned}$$

Therefore, it remains to show that

$$\sum_{k:\alpha_k \geq \sigma+1} \left(p\left(X = k | U = u\right) - \delta_k \right) \geq \sum_{k \in \mathcal{N} : \alpha_k \leq \sigma} \left(\delta_k - p\left(X = k | U = u\right) \right),$$

which can be written as

$$\sum_{k \in \mathcal{N}} \left(p\left(X = k | U = u\right) - \delta_k \right) \geq 0. \tag{A.8}$$

Since $\sum_{k \in \mathcal{N}} p\left(X = k | U = u\right) = \sum_{k \in \mathcal{N}} \delta_k = 1$, we can easily see that (A.8) holds, which completes the proof.

One may notice that we indeed show that the equality holds in (A.6), which implies that Q would be an all-zeros matrix after iterations, i.e., for any given $\bar{x}, \bar{u} \in \mathcal{N}$,

$$\sum_{\ell=1}^{\sigma+1} \sum_{x=1}^{N} \sum_{i,j : x_{i,j} = \bar{x}, u^{(x,i)} = \bar{u}} v_{i,j} = \max \left\{ p\left(X = \bar{x} | U = \bar{u}\right) - \delta_{\bar{x}}, 0 \right\}. \tag{A.9}$$

Here, we slightly abuse the notation since $v_{i,j}$ should be independent of x and ℓ as described.

A.4.2 Justification of the Algorithm

In this subsection, we will verify that the proposed algorithm works, i.e., it ends up with producing a distribution $p(z, x|u)$ satisfying that

$$p(z, x) = 0, \ \forall x \notin z, \tag{A.10}$$

$$p(z|u) = p\left(z|u'\right), \ \forall z \in \mathcal{Z} \text{ and } u, u' \in \mathcal{N}, \tag{A.11}$$

and

$$p(|Z| = i) = \theta_i, \ \forall i = 1, \ldots, \sigma + 1. \tag{A.12}$$

As claimed, $q(z, x, u)$ stores the non-zero valued probability of $p(z, x|u)$. To establish this claim, we need to verify that

$$\sum_{z':(z',x,u)\in\mathcal{A}} q\left(z', x, u\right) = p\left(x|u\right), \ \forall x, u \in \mathcal{N}. \tag{A.13}$$

Since $\mathcal{A}$ is the union set of $\mathcal{A}_{k,x,\ell}$ for all possible k, x and ℓ, let us focus on $\mathcal{A}_{k,x,\ell}$ defined in (2.62). Recall that

$$\mathcal{A}_{k,x,\ell} = \Big\{ (\bar{z}, \bar{x}, \bar{u}) : \bar{z} = z_k, \bar{x} = \zeta_k(i), \bar{u} = u^{(x,i)}, i = 1, \ldots, \ell - 1 \Big\} \bigcup \Big\{ (\bar{z}, \bar{x}, \bar{u}) : \bar{z} = z_k, \bar{x} = x, \bar{u} \in \mathcal{N} \setminus \left\{ u^{(x,i)} : i = 1, \ldots, \ell - 1 \right\} \Big\}.$$

Denote

$$\mathcal{A}^{(1)}_{k,x,\ell} = \left\{ (\bar{z}, \bar{x}, \bar{u}) : \bar{z} = z_k, \bar{x} = \zeta_k(i), \ \bar{u} = u^{(x,i)}, i = 1, \ldots, \ell - 1 \right\},$$

and

$$\mathcal{A}^{(2)}_{k,x,\ell} = \Big\{ (\bar{z}, \bar{x}, \bar{u}) : \bar{z} = z_k, \bar{x} = x, \bar{u} \in \mathcal{N} \setminus \left\{ u^{(x,i)} : i = 1, \ldots, \ell - 1 \right\} \Big\}.$$

For any given $\bar{u}, \bar{x} \in \mathcal{N}$, we have

$$\begin{aligned}
&\sum_{z:(z,\bar{u},\bar{x})\in\mathcal{A}} q\left(z,\bar{x},\bar{u}\right) \\
&\overset{\text{(a)}}{=} \sum_{\ell=1}^{\sigma+1} \sum_{z:(z,\bar{u},\bar{x})\in\mathcal{A}_\ell} q\left(z,\bar{x},\bar{u}\right) \\
&\overset{\text{(b)}}{=} \sum_{\ell=1}^{\sigma+1}\sum_{x=1}^{N} \sum_{k:(\cdot,\bar{x},\bar{u})\in\mathcal{A}_{k,x,\ell}} \nu_{k,x,\ell} \\
&= \sum_{\ell=1}^{\sigma+1}\sum_{x=1}^{N} \left(\sum_{k:(\cdot,\bar{x},\bar{u})\in\mathcal{A}^{(1)}_{k,x,\ell}} \nu_{k,x,\ell} + \sum_{k:(\cdot,\bar{x},\bar{u})\in\mathcal{A}^{(2)}_{k,x,\ell}} \nu_{k,x,\ell} \right) \\
&= \sum_{\ell=1}^{\sigma+1}\sum_{x=1}^{N} \sum_{k:(\cdot,\bar{x},\bar{u})\in\mathcal{A}^{(1)}_{k,x,\ell}} \nu_{k,x,\ell} + \sum_{\ell=1}^{\sigma+1}\sum_{x=1}^{N} \sum_{k:(\cdot,\bar{x},\bar{u})\in\mathcal{A}^{(2)}_{k,x,\ell}} \nu_{k,x,\ell}.
\end{aligned} \tag{A.14}$$

where (a) follows because $\mathcal{A}_\ell$ are disjoint for distinct ℓ and (b) follows from (2.65).

For any given $\bar{u} \in \mathcal{N}$, suppose that

$$\bar{u} = u^{(1,\alpha_1)} = \cdots = u^{(N,\alpha_N)}. \tag{A.15}$$

Then, the first term of the right-hand side of (A.14) can be written as

$$\begin{aligned}
&\sum_{\ell=1}^{\sigma+1}\sum_{x=1}^{N} \sum_{k:(\cdot,\bar{x},\bar{u})\in\mathcal{A}^{(1)}_{k,x,\ell}} \nu_{k,x,\ell} \\
&= \sum_{\ell=1}^{\sigma+1} \sum_{x:\alpha_x\le\ell-1} \sum_{k:\bar{x}=\zeta_k(\alpha_x)} \nu_{k,x,\ell} \\
&\overset{\text{(a)}}{=} \sum_{\ell=1}^{\sigma+1} \sum_{x:\alpha_x\le\ell-1} v_{\alpha_x,\bar{x}} \\
&\overset{\text{(b)}}{=} \max\left\{p\left(X=\bar{x}|U=\bar{u}\right) - \delta_{\bar{x}}, 0\right\},
\end{aligned} \tag{A.16}$$

where (a) follows from (2.60), and (b) follows from (A.9). Note that we slightly abuse the notation ζ_k and $v_{\alpha_k,\bar{x}}$ here since they are independent of x and ℓ.

The second term of the right-hand side of (A.14) can be written as

$$\begin{aligned}
&\sum_{\ell=1}^{\sigma+1}\sum_{x=1}^{N}\sum_{k:(\cdot,\bar{x},\bar{u})\in\mathcal{A}_{k,x,\ell}^{(2)}}\nu_{k,x,\ell}\\
&=\sum_{\ell=1}^{\sigma+1}\sum_{x:x=\bar{x}}\mathbb{1}_{\alpha_x\geq\ell}\sum_{k}\nu_{k,x,\ell}\\
&=\sum_{\ell=1}^{\sigma+1}\mathbb{1}_{\alpha_{\bar{x}}\geq\ell}\sum_{k}\nu_{k,\bar{x},\ell}\\
&\overset{\text{(a)}}{=}\sum_{\ell=1}^{\sigma+1}\mathbb{1}_{\alpha_{\bar{x}}\geq\ell}\left(\min\left\{\delta_{\bar{x}},p\left(X=\bar{x}|U=u^{(\bar{x},\ell)}\right)\right\}\right.\\
&\quad\left.-p\left(X=\bar{x}|U=u^{(\bar{x},\ell-1)}\right)\right)\\
&=\sum_{\ell=1}^{\min\{\sigma+1,\alpha_{\bar{x}}\}}\left(\min\left\{\delta_{\bar{x}},p\left(X=\bar{x}|U=u^{(\bar{x},\ell)}\right)\right\}\right.\\
&\quad\left.-p\left(X=\bar{x}|U=u^{(\bar{x},\ell-1)}\right)\right)\\
&=\min\left\{\delta_{\bar{x}},p\left(X=\bar{x}|U=u^{(\bar{x},\alpha_{\bar{x}})}\right)\right\}\\
&=\min\left\{\delta_{\bar{x}},p\left(X=\bar{x}|U=\bar{u}\right)\right\},
\end{aligned}\tag{A.17}$$

where (a) follows from (2.54) and (2.60).

Finally, it is easy to see that

$$\begin{aligned}
&\sum_{\ell=1}^{\sigma+1}\sum_{x=1}^{N}\sum_{k:(\cdot,\bar{x},\bar{u})\in\mathcal{A}_{k,x,\ell}}\nu_{k,x,\ell}\\
&=\sum_{\ell=1}^{\sigma+1}\sum_{x=1}^{N}\sum_{k:(\cdot,\bar{x},\bar{u})\in\mathcal{A}_{k,x,\ell}^{(1)}}\nu_{k,x,\ell}+\sum_{\ell=1}^{\sigma+1}\sum_{x=1}^{N}\sum_{k:(\cdot,\bar{x},\bar{u})\in\mathcal{A}_{k,x,\ell}^{(2)}}\nu_{k,x,\ell}\\
&\overset{\text{(a)}}{=}\max\left\{p\left(X=\bar{x}|U=\bar{u}\right)-\delta_{\bar{x}},0\right\}+\min\left\{\delta_{\bar{x}},p\left(X=\bar{x}|U=\bar{u}\right)\right\}\\
&=p\left(X=\bar{x}|U=\bar{u}\right),
\end{aligned}$$

where (a) follows from (A.16) and (A.17). We finish justifying (A.13).

Now, let us verify the two constraints (A.10) and (A.11), i.e.,

$$p\left(z,x\right)=0,\ \forall x\notin z,$$

and

$$p\left(z|u\right)=p\left(z|u'\right),\ \forall z\in\mathcal{Z}\text{ and }u,u'\in\mathcal{N}.$$

As we have shown that

$$p\left(z,x|u\right)=\begin{cases}q\left(z,x,u\right), & (z,x,u)\in\mathcal{A},\\ 0, & (z,x,u)\notin\mathcal{A},\end{cases}$$

to verify the two constraints, it is equivalent to show that

1. For any $(z,x,u)\in\mathcal{A}$, it must have $x\in z$.

2. For any given $z\in\mathcal{Z}$ and $u,u'\in\mathcal{N}$, we have

$$\sum_{x:(z,x,u)\in\mathcal{A}}q\left(z,x,u\right)=\sum_{x:(z,x,u')\in\mathcal{A}}q\left(z,x,u'\right).$$

Since $\mathcal{A}$ is the union set of $\mathcal{A}_{k,x,\ell}$ for all possible k, x and ℓ, it is sufficient to show the following two claims:

1. For any $(z,x,u)\in\mathcal{A}_{k,x,\ell}$, it must have $x\in z$.

2. For any given $z\in\mathcal{Z}$ and $u,u'\in\mathcal{N}$, we have

$$\sum_{x,\bar{x},k:(z,x,u)\in\mathcal{A}_{k,\bar{x},\ell}}\nu_{k,\bar{x},\ell}=\sum_{x,\bar{x},k:(z,x,u')\in\mathcal{A}_{k,\bar{x},\ell}}\nu_{k,\bar{x},\ell},\tag{A.18}$$

where $\ell=|z|$, $x,\bar{x}=1,\ldots,N$ and $k=1,\ldots,e_{\bar{x},\ell}$.

Recall the definition of $\mathcal{A}_{k,x,\ell}$ for any k, x and ℓ, i.e.,

$$\mathcal{A}_{k,x,\ell}=\left\{(\bar{z},\bar{x},\bar{u}):\bar{z}=z_k,\bar{x}=\zeta_k(i),\bar{u}=u^{(x,i)},i=1,\ldots,\ell-1\right\}\bigcup\left\{(\bar{z},\bar{x},\bar{u}):\bar{z}=z_k,\bar{x}=x,\bar{u}\in\mathcal{N}\setminus\left\{u^{(x,i)}:i=1,\ldots,\ell-1\right\}\right\}.$$

Since $z_k=\{\zeta_k,x\}$ as previously defined, we can easily see that $\bar{x}\in\bar{z}$ for any $(\bar{z},\bar{x},\bar{u})\in\mathcal{A}_{k,x,\ell}$, which justifies the first claim.

For the second claim, we re-write (A.18) by

$$\sum_{\bar{x},k} \nu_{k,\bar{x},\ell} \sum_{x:(z,x,u)\in\mathcal{A}_{k,\bar{x},\ell}} 1 = \sum_{\bar{x},k} \nu_{k,\bar{x},\ell} \sum_{x:(z,x,u')\in\mathcal{A}_{k,\bar{x},\ell}} 1.$$

By inspecting the definition of $\mathcal{A}_{k,x,\ell}$, we see that there exists exactly one tuple $(z,\cdot,u) \in \mathcal{A}_{k,x,\ell}$ for any given u and z, so we have

$$\sum_{x:(z,x,u)\in\mathcal{A}_{k,\bar{x},\ell}} 1 = \sum_{x:(z,x,u')\in\mathcal{A}_{k,\bar{x},\ell}} 1,$$

which completes proving (A.18).

Finally, let us justify (A.12), i.e.,

$$p\left(|Z|=i\right) = \theta_i,\ \forall i = 1,\ldots,\sigma+1,$$

whose proof is given as follows:

$$\begin{aligned}
p\left(|Z|=\ell\right) &= \sum_{z:|z|=\ell} \sum_{u} p\left(u\right) p\left(z|u\right) \\
&\overset{\text{(a)}}{=} \sum_{z:|z|=\ell} p\left(z|\bar{u}\right) \\
&= \sum_{z:|z|=\ell} \sum_{x} q\left(z,x,\bar{u}\right) \\
&= \sum_{(z,x,\bar{u}):(z,x,\bar{u})\in\mathcal{A}_\ell} q\left(z,x,\bar{u}\right) \\
&\overset{\text{(b)}}{=} \sum_{(z,x,\bar{u})\in\mathcal{A}_\ell} \sum_{x'=1}^{N} \sum_{k:(z,x,\bar{u})\in\mathcal{A}_{k,x',\ell}} \nu_{k,x',\ell} \\
&= \sum_{x'=1}^{N} \sum_{k} \sum_{(z,x,\bar{u}):(z,x,\bar{u})\in\mathcal{A}_{k,x',\ell}} \nu_{k,x',\ell} \\
&\overset{\text{(c)}}{=} \sum_{x'=1}^{N} \sum_{k} \nu_{k,x',\ell} \\
&\overset{\text{(d)}}{=} \sum_{x'=1}^{N} \min\left\{\delta_{x'}, p\left(X=x'|U=u^{(x',\ell)}\right)\right\} - p\left(X=x'|U=u^{(x',\ell-1)}\right)
\end{aligned}$$

$$= \theta_\ell,$$

where (a) follows from (A.11), (b) follows from (2.65), (c) follows because there exists exactly one tuple $(\cdot, \cdot, \bar{u}) \in \mathcal{A}_{k,x,\ell}$ for given k, x and ℓ, and (d) follows from (2.54) and (2.60).

A.4.3 Complexity analysis of the Algorithm

In this subsection, we will discuss the complexity of the algorithm to construct the desired output distribution $p\left(z|x,u\right)$. The purpose of the complexity analysis here is to justify that the proposed algorithm is tractable, i.e., with poly(N) complexity. By utilizing some data structures, one may possibly reduce the complexity by one or two orders, which is beyond the interest of this paper.

Pre-calculation: The Pre-calculation involves two steps, i.e., sorting $p\left(x|u\right)$ for all $x \in \mathcal{N}$ and picking the set $\{\delta_j : j = 1, \ldots, N\}$. The complexity of sorting is $\mathcal{O}\left(N^2 \log N\right)$ and picking $\{\delta_j : j = 1, \ldots, N\}$ is $\mathcal{O}\left(N\right)$.

Initialization: The initialization of Q is $\mathcal{O}(N^2)$.

Procedure: The main procedure is divided into the following steps:

1. For the fixed ℓ, x and u_i, we 'randomly' choose a collection of pairs $I_i \times V_i$. We can easily see that if (2.57) is satisfied, then V_i (and I_i) can be chosen by linear time $\mathcal{O}(N)$, i.e., going through the u_i-th row of the matrix Q. Hence, we can obtain $\{I_i, V_i : i = 1, \ldots, \ell - 1\}$ for a fixed ℓ and x with $\mathcal{O}((\ell - 1)N)$.

2. For a fixed ℓ and x, we need to get a collection of pairs $\{(\zeta_k, \nu_k) : k = 1, 2, \ldots, e\}$ given $\{I_i, V_i : i = 1, \ldots, \ell - 1\}$. Each ν_k is obtained by finding the minimal value of B_v, which is a set of length $\ell - 1$, so finding each ν_k (and ζ_k) takes $\mathcal{O}(\ell - 1)$. For each z_k, the set $\mathcal{A}_k$ can be characterized by traversing z_k with linear time $\mathcal{O}(\ell - 1)$. As each e_i is bounded by $N - 1$, e is bounded by
$$e \leq \sum_{i=1}^{\ell-1} e_i = (\ell - 1)(N - 1),$$
and hence determining all $\{\mathcal{A}_k, \nu_k : k = 1, 2, \ldots, e\}$ takes $\mathcal{O}((\ell - 1)^2(N - 1))$. Therefore, obtaining $\{\mathcal{A}_{k,x,\ell}, \nu_{k,x,\ell} : 1 \leq \ell \leq \sigma + 1, 1 \leq x \leq N, 1 \leq k \leq e_{x,\ell}\}$ at most takes $\mathcal{O}(\sigma^3 N^2)$.

3. At the end, we need to finish the probability assignment (c.f.(2.65)). However, since the size

of the alphabet of Z is exponential, $p(z, x|u)$ has an exponential number of elements. To avoid the exponential overhead, we may take advantage of the sparsity of $p(z, x|u)$ to output non-zero positions and values (all others are assumed to be zero) instead of pushing out the distribution $p(z, x|u)$ entirely and directly. Indeed, $\mathcal{A}_{k,x,\ell}$ contains the non-zero positions and the corresponding value is $\nu_{k,x,\ell}$. However, since some positions may appear in $\cup_{k,x,\ell}\mathcal{A}_{k,x,\ell}$ multiple times, we may need to merge them, this can be done by simply checking all $\{\mathcal{A}_{k,x,\ell} : 1 \leq \ell \leq \sigma + 1, 1 \leq x \leq N, 1 \leq k \leq e_{x,\ell}\}$ which is $\mathcal{O}(\sigma^3 N^3)$.

In summary, the worst case complexity of the algorithm is $\mathcal{O}(N^6)$.

APPENDIX B

ON-OFF PRIVACY FOR PAST AND FUTURE CORRELATED REQUESTS

B.1 Optimality for $n = 2$

For $n = 2$ information sources, the two bounds (3.12) and (3.13) match, i.e., $R_t^I = R_t^O$. To see this we write R_t^I by

$$\frac{1}{R_t^I} = \sum_{i=1}^{m} i\,\bar{\theta}_i(t) = 2 - \bar{\lambda}_1(t).$$

For a given x_t, e.g., $x_t = 1$, suppose that $u^* = \arg\min_{u_t} p\left(x_t|u_t\right)$. Then we can see that, for $\bar{x}_t = 2$, $u^* = \arg\max_{u_t} p\left(\bar{x}_t|u_t\right)$ since $p\left(x_t|u_t\right) + p\left(\bar{x}_t|u_t\right) = 1$, for any u_t when $n = 2$. Thus,

$$\min_{u_t} p\left(x_t|u_t\right) + \max_{u_t} p\left(\bar{x}_t|u_t\right) = 1,$$

for any x_t and $\bar{x}_t = \mathcal{N}\backslash\{x_t\}$, which implies that

$$\begin{aligned}\bar{\lambda}_1(t) &= \sum_{x_t=1}^{2} \min_{u_t} p\left(x_t|u_t\right) = \sum_{x_t=1}^{2} \left(1 - \max_{u_t} p\left(\bar{x}_t|u_t\right)\right) \\ &= 2 - \sum_{\bar{x}_t=1}^{2} \max_{u_t} p\left(\bar{x}_t|u_t\right) = 2 - \bar{\lambda}_m(t).\end{aligned}$$

Therefore, we can obtain that

$$\frac{1}{R_t^I} = 2 - \bar{\lambda}_1(t) = \bar{\lambda}_m(t) = \frac{1}{R_t^O}.$$

B.2 Proof of Corollary 3

We first take the transition matrix P to the power of t, i.e.,

$$(P^t)_{i,j} = \begin{cases} \frac{(n-1)^{t-1} - (n\alpha-1)^t}{n(n-1)^{t-1}}, & \text{if} \quad i = j, \\ \frac{(n-1)^t - (n\alpha-1)^t}{n(n-1)^t}, & \text{if} \quad i \neq j, \end{cases}$$

for all $i, j \in \{1, \ldots, n\}$.

Then, the probabilities $p\left(x_t|u_t\right)$ can be written as

$$p\left(X_t = j|X_\tau = i, X_{t+1} = k\right) = \frac{P_{j,k}(P^\delta)_{i,j}}{(P^{\delta+1})_{i,k}}, \tag{B.1}$$

where $\delta = t - \tau$ and $i, j \in \{1, \ldots, n\}$. By invoking the symmetry of the given Markov chain, we notice that the right-hand side of (B.1) can only have a few of expressions depending on the choices of i, j and k, i.e.,

$$p\left(X_t = j|X_\tau = i, X_{t+1} = k\right) =$$
$$\begin{cases} \sigma_1 := \frac{\alpha\left((n-1)^\delta + (n\alpha-1)^\delta(n-1)\right)}{(n-1)^\delta + (n\alpha-1)^{\delta+1}}, & \text{if } i = j = k, \\ \sigma_2 := \frac{(1-\alpha)\left((n-1)^\delta + (n\alpha-1)^\delta(n-1)\right)}{(n-1)^{\delta+1} - (n\alpha-1)^{\delta+1}}, & \text{if } i = j \neq k, \\ \sigma_3 := \frac{\alpha\left((n-1)^{\delta+1} - (n\alpha-1)^\delta(n-1)\right)}{(n-1)^{\delta+1} - (n\alpha-1)^{\delta+1}}, & \text{if } i \neq j = k, \\ \sigma_4 := \frac{(1-\alpha)\left((n-1)^\delta - (n\alpha-1)^\delta\right)}{\left((n-1)^\delta + (n\alpha-1)^{\delta+1}\right)(n-1)}, & \text{if } i = k \neq j, \\ \sigma_5 := \frac{(1-\alpha)\left((n-1)^\delta - (n\alpha-1)^\delta\right)}{(n-1)^{\delta+1} - (n\alpha-1)^{\delta+1}}, & \text{if } i \neq j \neq k. \end{cases} \tag{B.2}$$

By examining σ_1 to σ_5 in (B.2), we have $\sigma_1 \geq \sigma_3 \geq \sigma_2 \geq \sigma_5 \geq \sigma_4$, for $\frac{1}{n} \leq \alpha \leq 1$.

For a fixed $j \in \{1, \ldots, n\}$, by counting the number of times each condition of (B.2), e.g., $i = j = k$, $i = j \neq k$, etc., is satisfied for $i, k \in \mathcal{N}$, we can get the following ordering of n^2 probabilities $p\left(X_t = j|X_\tau = i, X_{t+1} = k\right)$ (for a fixed j):

$$\underbrace{\sigma_4 \leq \cdots \leq \sigma_4}_{n-1} \leq \underbrace{\sigma_5 \leq \cdots \leq \sigma_5}_{(n-1)(n-2)} \geq \underbrace{\sigma_2 \leq \cdots \leq \sigma_2}_{n-1} \leq \underbrace{\sigma_3 \leq \cdots \leq \sigma_3}_{n-1} \leq \sigma_1.$$

Due to the symmetry, this ordering remains the same for all $j \in \{1, \ldots, n\}$. Given this ordering for any fixed j, we can check

$$\frac{1}{R_t^O} = \sum_{x \in \mathcal{N}} p\left(X_t = x|U_t = u_{x,n^2}\right) = \sum_{x \in \mathcal{N}} \sigma_1 = n\sigma_1,$$

where $p\left(X_t = x|U_t = u_{x,n^2}\right)$ is defined in (3.7).

Also, from (3.9), we can check that $\bar{\theta}_1 = n\sigma_4$, $\bar{\theta}_i = 0$ for $i = 2, \ldots, n-1$, and $\bar{\theta}_n = 1 - n\sigma_4$,

so we have

$$\frac{1}{R_t^I} = \sum_{i=1}^{n} i\bar{\theta}_i = n\sigma_4 + n - n^2\sigma_4.$$

By substituting the expression of σ_1 and σ_4 defined in (B.2), one can verify that $n\sigma_4 + n - n^2\sigma_4 = n\sigma_1$, which implies in $R_t^I = R_t^O$. This completes the proof of Corollary 3.

Remark 8. When $0 \leq \alpha < \frac{1}{n}$, we may follow the same steps as we did but divide the discussion into two cases: $\delta = t - \tau$ is even or odd. When δ is even, we have $\sigma_2 \geq \sigma_4 \geq \sigma_5 \geq \sigma_1 \geq \sigma_3$, and

$$\frac{1}{R_t^O} = n\sigma_2 \leq \frac{1}{R_t^I} = n\sigma_3 + n - n^2\sigma_3.$$

Similarly when δ is odd, we have $\sigma_5 \geq \sigma_4 \geq \sigma_2 \geq \sigma_3 \geq \sigma_1$, and

$$\frac{1}{R_t^O} = n\sigma_5 \leq \frac{1}{R_t^I} = \sigma_3(2n - n^2) - n\sigma_1 + n.$$

In both cases, we can see a gap between R_t^O and R_t^I, which is as per our observation in Example 2.

B.3 Proof of Lemma 3

First, let us recall that $\overline{\mathcal{B}}_t = \{i : i \leq t, F_i = \text{ON}\} \cup \{i : i \geq t+1\}$ and $U_t = (X_\tau, X_{t+1})$ where $\tau = \max\{i : i \leq t, F_i = \text{ON}\}$.

We prove the statement by induction on t. Consider the base case $t = 0$. From the assumption $F_0 = \text{ON}$ (assumption of this chapter), we know that $U_0 = \{X_0, X_1\}$ and $\overline{\mathcal{B}}_0 = \{i : i = 0, 1, \ldots\}$. If Q_0 is a stochastic function of X_0 and X_1, and Q_0 is independent of X_0 and X_1, then we have

$$I\left(Q_0; X_{\overline{\mathcal{B}}_0}\right) = I\left(Q_0; X_0, X_1\right) + I\left(Q_0; X_{\overline{\mathcal{B}}_0} | X_1, X_0\right) = 0,$$

i.e., Q_0 is independent of $X_{\overline{\mathcal{B}}_0}$. The last equality follows because Q_0 is independent of X_0 and X_1, and Q_0 is a stochastic function of X_0 and X_1.

Now, we start the inductive step. Assume that the statement is true for some $t-1$, i.e., if Q_i is a stochastic function of U_i and X_i, and Q_i is independent of U_i for $i = 0, 1, \ldots, t-1$, then $Q_{\lfloor t-1 \rfloor}$ is independent of $X_{\overline{\mathcal{B}}_{t-1}}$.

Next, for the case t, if Q_i is a stochastic function of U_i and X_i, and Q_i is independent of U_i for $i = 0, 1, \ldots, t$, then we know from the inductive assumption that $Q_{\lfloor t-1 \rfloor}$ is independent of $X_{\overline{\mathcal{B}}_{t-1}}$, i.e.,

$$I\left(Q_{\lfloor t-1 \rfloor}; X_{\overline{\mathcal{B}}_{t-1}}\right) = 0. \tag{B.3}$$

Then consider

$$\begin{aligned}
&I\left(Q_{\lfloor t \rfloor}; X_{\overline{\mathcal{B}}_t}\right) \\
&= I\left(Q_{\lfloor t-1 \rfloor}; X_{\overline{\mathcal{B}}_t}\right) + I\left(Q_t; U_t | Q_{\lfloor t-1 \rfloor}\right) + I\left(Q_t; X_{\overline{\mathcal{B}}_t \setminus \{\tau, t+1\}} | Q_{\lfloor t-1 \rfloor}, U_t\right) \\
&\overset{\text{(a)}}{\leq} I\left(Q_{\lfloor t-1 \rfloor}; X_{\overline{\mathcal{B}}_{t-1}}\right) + I\left(Q_t; U_t | Q_{\lfloor t-1 \rfloor}\right) + I\left(Q_t; X_{\overline{\mathcal{B}}_t \setminus \{\tau, t+1\}} | Q_{\lfloor t-1 \rfloor}, U_t\right) \\
&\overset{\text{(b)}}{=} I\left(Q_t; U_t | Q_{\lfloor t-1 \rfloor}\right) + I\left(Q_t; X_{\overline{\mathcal{B}}_t \setminus \{\tau, t+1\}} | Q_{\lfloor t-1 \rfloor}, U_t\right) \\
&= I\left(Q_t; U_t\right) + I\left(Q_t; Q_{\lfloor t-1 \rfloor} | U_t\right) - I\left(Q_t; Q_{\lfloor t-1 \rfloor}\right) + I\left(Q_t; X_{\overline{\mathcal{B}}_t \setminus \{\tau, t+1\}} | Q_{\lfloor t-1 \rfloor}, U_t\right) \\
&\leq I\left(Q_t; U_t\right) + I\left(Q_t; X_{\overline{\mathcal{B}}_t \setminus \{\tau, t+1\}}, Q_{\lfloor t-1 \rfloor} | U_t\right) \\
&\overset{\text{(c)}}{=} I\left(Q_t; U_t\right) + I\left(X_t; X_{\overline{\mathcal{B}}_t \setminus \{\tau, t+1\}}, Q_{\lfloor t-1 \rfloor} | U_t\right) \\
&= I\left(Q_t; U_t\right) + I\left(X_t; X_{\overline{\mathcal{B}}_t \setminus \{\tau, t+1\}} | U_t\right) + I\left(X_t; Q_{\lfloor t-1 \rfloor} | X_{\overline{\mathcal{B}}_t \setminus \{\tau, t+1\}}, U_t\right) \\
&\overset{\text{(d)}}{=} I\left(Q_t; U_t\right) + I\left(X_t; X_{\overline{\mathcal{B}}_t \setminus \{\tau, t+1\}} | U_t\right) \\
&\overset{\text{(e)}}{=} I\left(Q_t; U_t\right) = 0,
\end{aligned}$$

where we have that (a) follows from $\overline{\mathcal{B}}_t \subseteq \overline{\mathcal{B}}_{t-1}$ by inspecting the definition of $\overline{\mathcal{B}}_t$, (b) follows from (B.3), (c) follows because Q_t is a stochastic function of U_t and X_t, (d) follows from $X_{\overline{\mathcal{B}}_{t-1}} = \left\{X_{\overline{\mathcal{B}}_t \setminus \{\tau, t+1\}}, U_t, X_t\right\}$ and (B.3), and (e) follows from the Markovity of X_t.

B.4 Proof of Proposition 3

From the definitions in (3.8) and (3.9), we can easily see that $\bar{\lambda}_i$ is non-decreasing with i, so $\bar{\theta}_i \geq 0$ for all i if and only if $\bar{\lambda}_{n-1} \leq 1$. It is sufficient for us to show that $\bar{\lambda}_n \leq 1$.

For any given distribution $p\left(x_t | u_t\right)$ where $x_t \in \mathcal{N}$ and $u_t \in \mathcal{N}^2$, we claim that there exists some u such that

$$p\left(x_t = x | u_t = u\right) \geq p\left(x_t = x | u_t = u_{x,n}\right), \forall x \in \mathcal{N}. \tag{B.4}$$

To see this, one can choose any $u \in \mathcal{N}^2 \backslash \{u_{x,i} : x \in \mathcal{N}, i = 1, \ldots, n-1\}$. Note that since $|\mathcal{N}^2| = n^2$ and $|\{u_{x,i} : x \in \mathcal{N}, i = 1, \ldots, n-1\}| = n(n-1)$, the set $\mathcal{N}^2 \backslash \{u_{x,i} : x \in \mathcal{N}, i = 1, \ldots, n-1\}$ is non-empty.

By summing (B.4) over all x , we have

$$\sum_{x \in \mathcal{N}} p\left(x_t = x | u_t = u\right) \geq \sum_{x \in \mathcal{N}} p\left(x_t = x | u_t = u_{x,n}\right) = \bar{\lambda}_n.$$

Since $\sum_{x \in \mathcal{N}} p\left(x_t = x | u_t = u\right) = 1$ for a fixed u, we complete showing that $\bar{\lambda}_n \leq 1$.

B.5 Proof of Proposition 4

Recall that we need to show that for any $u = 1, \ldots, m$,

$$\sum_{x=1}^{n} \max\left\{p\left(X = x | U = u\right) - \bar{\lambda}_{x,n-1}, 0\right\} \geq \sum_{\ell=1}^{n-1} \sum_{x : u \in \mathcal{U}^-_{\ell,x}} \left(\bar{\lambda}_{x,\ell} - \bar{\lambda}_{x,\ell-1}\right). \tag{B.5}$$

Assume without loss of generality that $u = u_{1,\alpha_1} = \cdots = u_{n,\alpha_n}$. Then, the left-hand side of (B.5) can be written as

$$\begin{aligned}\sum_{x=1}^{n} \max\left\{p\left(X = x | U = u\right) - \bar{\lambda}_{x,n-1}, 0\right\} &= \sum_{x : \alpha_x \geq n-1} p\left(X = x | U = u\right) - \bar{\lambda}_{x,n-1} \\ &= \sum_{x : \alpha_x \geq n-1} \left(\bar{\lambda}_{x,\alpha_x} - \bar{\lambda}_{x,n-1}\right),\end{aligned}$$

and the right-hand side of (B.5) can be written as

$$\begin{aligned}\sum_{\ell=1}^{n-1} \sum_{x : u \in \mathcal{U}^-_{\ell,x}} \bar{\lambda}_{x,\ell} - \bar{\lambda}_{x,\ell-1} &= \sum_{x : \alpha_x \leq n-2} \sum_{\ell = \alpha_x + 1}^{n-1} \bar{\lambda}_{x,\ell} - \bar{\lambda}_{x,\ell-1} \\ &= \sum_{x : \alpha_x \leq n-2} \left(\bar{\lambda}_{x,n-1} - \bar{\lambda}_{x,\alpha_x}\right).\end{aligned}$$

Since

$$\sum_{x : \alpha_x \geq n-1} \left(\bar{\lambda}_{x,\alpha_x} - \bar{\lambda}_{x,n-1}\right) - \sum_{x : \alpha_x \leq n-2} \left(\bar{\lambda}_{x,n-1} - \bar{\lambda}_{x,\alpha_x}\right)$$

$$= \sum_{x=1}^{n} \left(\bar{\lambda}_{x,\alpha_x} - \bar{\lambda}_{x,n-1} \right) = 1 - \sum_{x=1}^{n} \bar{\lambda}_{x,n-1} = \bar{\theta}_n, \tag{B.6}$$

we can see that (B.5) is established if and only if $\bar{\theta}_n \geq 0$, which is given by Proposition 3. This completes the proof of Proposition 4.

Remark 9. To benefit the following proof, we give an immediate implication of (B.6) here. As described, the right-hand side of (B.5) is the total values assigned for $\ell = 1, \ldots, n-1$ and the left-hand side of (B.5) is the initialization of the matrix M, so the remaining values will be assigned for $\ell = n$ as described in (3.48) and (3.49). As such, we know from (B.6) that

$$\sum_{x:(z,x,u)\in\mathcal{F}_n} g(z,x,u) = \bar{\theta}_n, \tag{B.7}$$

for any $u \in \{1, \ldots, m\}$ and $z = \{1, \ldots, n\}$.

B.6 Proof of Proposition 5

Before proving the proposition, we provide some observations of $\mathcal{F}_{\ell,x,k}$ for some x, k and $\ell = 1, \ldots, n-1$ by examining (3.45). Let $\mathbb{1}\{\cdot\}$ be the indicator function.

- If $(\bar{z}, \bar{x}, \bar{u}) \in \mathcal{F}_{\ell,x,k}$ for some ℓ, x and k, then $\bar{z} = z_{\ell,x,k}$ is uniquely determined, i.e., if $(\bar{z}, \bar{x}, \bar{u}) \in \mathcal{F}_{\ell,x,k}$, then
$$\left(z', x', u'\right) \notin \mathcal{F}_{\ell,x,k}, \forall z' \neq \bar{z}. \tag{B.8}$$

- For any $(\bar{z}, \bar{x}, \bar{u}) \in \mathcal{F}_{\ell,x,k}$, $|\bar{z}| = \ell$, and $\bar{x} \in \bar{z}$.

- The cardinality of each $\mathcal{F}_{\ell,x,k}$ is $|\mathcal{F}_{\ell,x,k}| = m$. In particular, all tuples $(\bar{z}, \bar{x}, \bar{u}) \in \mathcal{F}_{\ell,x,k}$ have distinct values of $\bar{u}$. In other words, let $(\bar{z}, \cdot, \cdot) \in \mathcal{F}_{\ell,x,k}$ denote that there exists some $\bar{x}, \bar{u}$ such that $(\bar{z}, \bar{x}, \bar{u}) \in \mathcal{F}_{\ell,x,k}$, and then
$$\sum_{x'=1}^{n} \mathbb{1}\left\{\left(\bar{z}, x', u'\right) \in \mathcal{F}_{\ell,x,k}\right\} = \mathbb{1}\left\{(\bar{z}, \cdot, \cdot) \in \mathcal{F}_{\ell,x,k}\right\}, \tag{B.9}$$
for any $u' \in \{1, \ldots, m\}$.

1. The first statement is straightforward. Suppose that $(\bar{z}, \bar{x}, \bar{u}) \in \mathcal{F}_\ell$ for some ℓ. If $\ell = 1, \ldots, n-1$, we know from (3.46) that $\mathcal{F}_\ell$ is the union of $\mathcal{F}_{\ell,x,k}$. For each $\mathcal{F}_{\ell,x,k}$, we know that $\bar{x} \in \bar{z}$ for any $(\bar{z}, \bar{x}, \bar{u}) \in \mathcal{F}_{\ell,x,k}$. If $\ell = n$, $\bar{z} = \mathcal{N}$ for all $(\bar{z}, \bar{x}, \bar{u}) \in \mathcal{F}_n$, so $\bar{x} \in \bar{z}$.

2. For the second statement, when $|\bar{z}| = \ell \in \{1, \ldots, n-1\}$,

$$\begin{aligned}\sum_{x:(\bar{z},x,\bar{u})\in\mathcal{F}} g(\bar{z}, x, \bar{u}) &= \sum_{x:(\bar{z},x,\bar{u})\in\mathcal{F}_\ell} g(\bar{z}, x, \bar{u}) \\ &\overset{\text{(a)}}{=} \sum_{x} \sum_{x'=1}^{n} \sum_{k:(\bar{z},x,\bar{u})\in\mathcal{F}_{\ell,x',k}} \nu_{\ell,x',k} \\ &\overset{\text{(b)}}{=} \sum_{x'=1}^{n} \sum_{k=1}^{c_{\ell,x'}} \nu_{\ell,x',k} \cdot \mathbb{1}\left\{(\bar{z}, \cdot, \cdot) \in \mathcal{F}_{\ell,x',k}\right\}, \end{aligned} \tag{B.10}$$

   where (a) follows from (3.47) and (b) follows from (B.9).

   When $|\bar{z}| = \ell = n$, i.e., $\bar{z} = \{1, \ldots, n\}$, we have

$$\sum_{x:(\bar{z},x,\bar{u})\in\mathcal{F}} g(\bar{z}, x, \bar{u}) = \sum_{x:(\bar{z},x,\bar{u})\in\mathcal{F}_n} g(\bar{z}, x, \bar{u}) \overset{\text{(c)}}{=} \bar{\theta}_n, \tag{B.11}$$

   where (c) follows from (B.7).

   Since both the right-hand sides of (B.10) and (B.11) are independent of $\bar{u}$, for any given $\bar{z}$, $\bar{u}$ and $\bar{u}'$

$$\sum_{x:(\bar{z},x,\bar{u})\in\mathcal{F}} g(\bar{z}, x, \bar{u}) = \sum_{x:(\bar{z},x,\bar{u}')\in\mathcal{F}} g(\bar{z}, x, \bar{u}').$$

3. As for the third statement, for any given $\bar{x}$ and $\bar{u}$,

$$\begin{aligned}\sum_{z:(z,\bar{x},\bar{u})\in\mathcal{F}} g(z, \bar{x}, \bar{u}) &= \sum_{\ell=1}^{n} \sum_{z:(z,\bar{x},\bar{u})\in\mathcal{F}_\ell} g(z, \bar{x}, \bar{u}) \\ &= \sum_{\ell=1}^{n-1} \sum_{z:(z,\bar{x},\bar{u})\in\mathcal{F}_\ell} g(z, \bar{x}, \bar{u}) + \sum_{z:(z,\bar{x},\bar{u})\in\mathcal{F}_n} g(z, \bar{x}, \bar{u}). \end{aligned} \tag{B.12}$$

   For the first term of (B.12), we have

$$\sum_{\ell=1}^{n-1} \sum_{z:(z,\bar{x},\bar{u})\in\mathcal{F}_\ell} g(z, \bar{x}, \bar{u}) \overset{\text{(a)}}{=} \sum_{\ell=1}^{n-1} \sum_{z} \sum_{x=1}^{n} \sum_{k:(z,\bar{x},\bar{u})\in\mathcal{F}_{\ell,x,k}} \nu_{\ell,x,k}$$

$$\overset{\text{(b)}}{=} \sum_{\ell=1}^{n-1} \sum_{x=1}^{n} \sum_{k:(\cdot,\bar{x},\bar{u})\in\mathcal{F}_{\ell,x,k}} \nu_{\ell,x,k}, \tag{B.13}$$

where (a) follows by substituting (3.47), and (b) follows from (B.8).

By examining $\mathcal{F}_{\ell,x,k}$, we can see two disjoint subsets,

$$\mathcal{F}^{-}_{\ell,x,k} := \{(\bar{z},\bar{x},\bar{u}) : \bar{z} = z_{\ell,x,k}, \bar{x} = \zeta_{\ell,x,k}(i), \bar{u} = u_{x,i} \in \mathcal{U}^{-}_{\ell,x}\},$$

and

$$\mathcal{F}^{+}_{\ell,x,k} := \left\{(\bar{z},\bar{x},\bar{u}) : \bar{z} = z_{\ell,x,k}, \bar{x} = x, \bar{u} \in \mathcal{U}^{+}_{\ell,x}\right\},$$

For a fixed $\bar{u}$, assume that $\bar{u} = u_{1,\alpha_1} = \cdots = u_{n,\alpha_n}$. Then, we write (B.13) as

$$\begin{aligned}
\sum_{\ell=1}^{n-1} \sum_{z:(z,\bar{x},\bar{u})\in\mathcal{F}_\ell} g\left(z,\bar{x},\bar{u}\right) &= \sum_{\ell=1}^{n-1}\sum_{x=1}^{n} \sum_{k:(\cdot,\bar{x},\bar{u})\in\mathcal{F}_{\ell,x,k}} \nu_{\ell,x,k} \\
&= \sum_{\ell=1}^{n-1} \sum_{x:\alpha_x\le\ell-1} \sum_{k:\bar{x}=\zeta_{\ell,x,k}(\alpha_x)} \nu_{\ell,x,k} + \sum_{\ell=1}^{n-1} \sum_{k:\alpha_{\bar{x}}\ge\ell} \nu_{\ell,\bar{x},k} \\
&= \sum_{\ell=1}^{n-1} \sum_{x:\alpha_x\le\ell-1} \sum_{k:\bar{x}=\zeta_{\ell,x,k}(\alpha_x)} \nu_{\ell,x,k} + \sum_{\ell=1}^{\min\{n-1,\alpha_{\bar{x}}\}} \sum_{k} \nu_{\ell,\bar{x},k}.
\end{aligned} \tag{B.14}$$

From (3.42), we know that $\sum_k \nu_{\ell,\bar{x},k} = \bar{\lambda}_{\bar{x},\ell} - \bar{\lambda}_{\bar{x},\ell-1}$, and hence the second term of (B.14) can be written as

$$\sum_{\ell=1}^{\min\{n-1,\alpha_{\bar{x}}\}} \sum_{k} \nu_{\ell,\bar{x},k} = \bar{\lambda}_{\bar{x},\min\{n-1,\alpha_{\bar{x}}\}}. \tag{B.15}$$

For the first term of (B.14), we know from (3.41) that

$$\sum_{k:\zeta_{\ell,x,k}(\alpha_x)=\bar{x}} \nu_{\ell,x,k} = v_{\ell,x,\alpha_x,j},$$

and $\bar{x} = e_{\ell,x,i,j}$ for some j, where $v_{\ell,x,\alpha_x,j}$ and $e_{\ell,x,i,j}$ are defined in (3.36). Then, we know

from (3.39) that

$$\sum_{k:\zeta_{\ell,x,k}(\alpha_x)=\bar{x}} \nu_{\ell,x,k} = v_{\ell,x,\alpha_x,j} = M^-_{u_{x,\alpha_x},\bar{x}} = M^-_{\bar{u},\bar{x}},$$

i.e., the value subtracted from $M_{\bar{u},\bar{x}}$ for given ℓ and x. Thus, we have

$$\sum_{\ell=1}^{n-1} \sum_{x:\alpha_x\leq\ell-1} \sum_{k:\bar{x}=\zeta_{\ell,x,k}(\alpha_x)} \nu_{\ell,x,k} = \sum_{\ell=1}^{n-1} \sum_{x:\alpha_x\leq\ell-1} M^-_{\bar{u},\bar{x}} = \sum_{\ell=1}^{n-1} \sum_{x:\bar{u}\in\mathcal{U}^-_{\ell,x}} M^-_{\bar{u},\bar{x}}, \tag{B.16}$$

i.e., all values subtracted from $M_{\bar{u},\bar{x}}$ for $\ell = 1,\ldots,n-1$ and all x. By substituting (B.16) and (B.15) in (B.14), we have

$$\sum_{\ell=1}^{n-1} \sum_{z:(z,\bar{x},\bar{u})\in\mathcal{F}_\ell} g\left(z,\bar{x},\bar{u}\right) = \sum_{\ell=1}^{n-1} \sum_{x:\bar{u}\in\mathcal{U}^-_{\ell,x}} M^-_{\bar{u},\bar{x}} + \bar{\lambda}_{\bar{x},\min\{n-1,\alpha_{\bar{x}}\}}. \tag{B.17}$$

Then, substituting (B.17) in (B.12), we have

$$\begin{aligned}\sum_{z:(z,\bar{x},\bar{u})\in\mathcal{F}} g\left(z,\bar{x},\bar{u}\right) &= \sum_{\ell=1}^{n-1} \sum_{z:(z,\bar{x},\bar{u})\in\mathcal{F}_\ell} g\left(z,\bar{x},\bar{u}\right) + \sum_{z:(z,\bar{x},\bar{u})\in\mathcal{F}_n} g\left(z,\bar{x},\bar{u}\right) \\ &= \sum_{\ell=1}^{n-1} \sum_{x:\bar{u}\in\mathcal{U}^-_{\ell,x}} M^-_{\bar{u},\bar{x}} + \bar{\lambda}_{\bar{x},\min\{n-1,\alpha_{\bar{x}}\}} + \sum_{z:(z,\bar{x},\bar{u})\in\mathcal{F}_n} g\left(z,\bar{x},\bar{u}\right).\end{aligned}$$

Recalling that we assign all the remaining values $M_{\bar{u},\bar{x}}$ to $g\left(z,\bar{x},\bar{u}\right)$ in (3.48) and (3.49) when $\ell = n$, we know that

$$\sum_{\ell=1}^{n-1} \sum_{x:\bar{u}\in\mathcal{U}^-_{\ell,x}} M^-_{\bar{u},\bar{x}} + \sum_{z:(z,\bar{x},\bar{u})\in\mathcal{F}_n} g\left(z,\bar{x},\bar{u}\right) = \max\left\{p\left(X=\bar{x}|U=\bar{u}\right) - \bar{\lambda}_{\bar{x},n-1}, 0\right\},$$

i.e., the initial value of $M_{\bar{u},\bar{x}}$ defined in (3.35). Therefore,

$$\begin{aligned}\sum_{z:(z,\bar{x},\bar{u})\in\mathcal{F}} g\left(z,\bar{x},\bar{u}\right) &= \sum_{\ell=1}^{n-1} \sum_{x:\bar{u}\in\mathcal{U}^-_{\ell,x}} M^-_{\bar{u},\bar{x}} + \bar{\lambda}_{\bar{x},\min\{n-1,\alpha_{\bar{x}}\}} + \sum_{z:(z,\bar{x},\bar{u})\in\mathcal{F}_n} g\left(z,\bar{x},\bar{u}\right) \\ &= \max\left\{p\left(X=\bar{x}|U=\bar{u}\right) - \bar{\lambda}_{\bar{x},n-1}, 0\right\} + \bar{\lambda}_{\bar{x},\min\{n-1,\alpha_{\bar{x}}\}} \\ &= \max\left\{p\left(\bar{x}|\bar{u}\right) - \bar{\lambda}_{\bar{x},n-1}, 0\right\} + \min\left\{\bar{\lambda}_{\bar{x},n-1}, p\left(\bar{x}|\bar{u}\right)\right\}\end{aligned}$$

$$= p\left(\bar{x}|\bar{u}\right),$$

which completes the proof.

B.7 Proof of Proposition 6

The proof is quite straightforward from previous intermediate steps. As we know that $p\left(z|u\right) = p\left(z|u'\right)$ for any $u, u' \in \mathcal{N}^2$ and $z \in \mathcal{Z}$ from Proposition 5, for any given $\ell \in \{1, \ldots, n-1\}$, we have

$$\begin{aligned}
\sum_{z:|z|=\ell} p\left(z\right) = \sum_{z:|z|=\ell} p\left(z|u\right) &= \sum_{z:|z|=\ell} \sum_{x:(z,x,u)\in\mathcal{F}} g\left(z,x,u\right) \\
&\overset{\text{(a)}}{=} \sum_{z:|z|=\ell} \sum_{x'=1}^{n} \sum_{k=1}^{c_{\ell,x'}} \nu_{\ell,x',k} \cdot \mathbb{1}\left\{(z,\cdot,\cdot) \in \mathcal{F}_{\ell,x',k}\right\} \\
&= \sum_{x'=1}^{n} \sum_{k=1}^{c_{\ell,x'}} \nu_{\ell,x',k} \cdot \sum_{z:|z|=\ell} \mathbb{1}\left\{(z,\cdot,\cdot) \in \mathcal{F}_{\ell,x',k}\right\} \\
&\overset{\text{(b)}}{=} \sum_{x'=1}^{n} \sum_{k=1}^{c_{\ell,x'}} \nu_{\ell,x',k} \overset{\text{(c)}}{=} \sum_{x'=1}^{n} \bar{\lambda}_{x',\ell} - \bar{\lambda}_{x',\ell-1} = \bar{\theta}_\ell,
\end{aligned}$$

where (a) follows from (B.10), (b) follows from (B.8), and (c) follows from (3.42). For $\ell = n$, we have

$$\sum_{z:|z|=n} p\left(z\right) = 1 - \sum_{\ell=1}^{n-1} \sum_{z:|z|=\ell} p\left(z\right) = 1 - \sum_{\ell=1}^{n-1} \bar{\theta}_\ell = \bar{\theta}_n$$

by definition, then $p\left(|Z| = \ell\right) = \bar{\theta}_\ell$, for all $\ell = 1, \ldots, n$.

B.8 Proof of Lemma 4

Consider

$$\begin{aligned}
\max_{u\in\mathcal{N}^2} p\left(x|u\right) &\overset{\text{(a)}}{=} \max_{u\in\mathcal{N}} \sum_{y:x\in y} p\left(x,y|u\right) \\
&= \max_{u\in\mathcal{N}^2} \sum_{y:x\in y} p\left(y|u\right) p\left(x|y,u\right) \\
&\overset{\text{(b)}}{=} \max_{u\in\mathcal{N}^2} \sum_{y:x\in y} p\left(y\right) p\left(x|y,u\right)
\end{aligned}$$

$$\leq \sum_{y:x\in y} p\left(y\right) \max_{u\in\mathcal{N}^2} p\left(x|y,u\right)$$
$$\leq \sum_{y:x\in y} p\left(y\right),$$

where (a) follows from $p(x,y|u) = 0$ for $x \notin y$, and (b) follows because Y is independent of U. Thus, we obtain that

$$\sum_{x\in\mathcal{N}} \max_{u\in\mathcal{N}^2} p\left(x|u\right) \leq \sum_{x\in\mathcal{N}} \sum_{y:x\in y} p\left(y\right) = \sum_{y\in\mathscr{P}(\mathcal{N})} \sum_{x:x\in y} p\left(y\right)$$
$$= \sum_{y\in\mathscr{P}(\mathcal{N})} p\left(y\right) \sum_{x:x\in y} 1 = \mathbb{E}\left[|Y|\right],$$

which completes the proof.

APPENDIX C

PRIVATE MUTI-GROUP AGGREGATION

C.1 The Q&A Scheme

C.1.1 Proof of Theorem 5

We separate the proof into three parts starting with communication, then privacy, and finally with the accuracy.

1. **Communication:** The user sends the server the index of a column of the query matrix. Since the query matrix has dimension $k \times 2m$, the user sends $\log(2m)$ bits to the server.

2. **Privacy:** From Definition 4.2, the privacy of user i is

$$\begin{aligned} e^{\epsilon_{\text{QA}}} &= \max\left\{ \max_{\substack{g,g'\in\mathcal{G},g\neq g',\\ a\in[2m],q\in\mathcal{Q}}} \frac{\Pr\left(A_i=a\middle|G_i=g,Q_i=q\right)}{\Pr\left(A_i=a\middle|G_i=g',Q_i=q\right)}, 1 \right\} \\ &= \max_{\substack{g,g'\in\mathcal{G},g\neq g',\\ a\in[2m],q\in\mathcal{Q}}} \frac{\Pr\left(A_i=a\middle|G_i=g,Q_i=q\right)}{\Pr\left(A_i=a\middle|G_i=g',Q_i=q\right)}, \end{aligned} \tag{C.1}$$

where $\mathcal{Q}$ is as defined in (4.10). Notice that if $g = g'$, the ratio of probabilities is equal to 1, and if $g \neq g'$, the maximum of the ratio of probabilities is greater than or equal to 1. Consider

$$\begin{aligned} &\Pr\left(A_i=a\middle|G_i=g,Q_i=q\right) \\ &\overset{\text{(a)}}{=} \sum_{v\in\mathcal{V}} \Pr(V_i=v\middle|G_i=g) \sum_{\mathring{v}\in\mathcal{V}} \Pr\left(\mathring{V}_i=\mathring{v}\middle|V_i=v\right) \Pr\left(A_i=a\middle|G_i=g,Q_i=q,\mathring{V}_i=\mathring{v}\right) \\ &\overset{\text{(b)}}{=} \sum_{v\in\mathcal{V}} p_g(v) \Pr\left(\mathring{V}_i=v^*\middle|V_i=v\right) \\ &\overset{\text{(c)}}{=} (1-\lambda)p_g(v^*) + \tfrac{\lambda}{2m-1}(1-p_g(v^*)), \end{aligned} \tag{C.2}$$

where (a) follows from the law of total probability and the random variable relationships. As for (b), it follows from definition $p_g(v) := \Pr(V_i = v|G_i = g)$, and noticing that given a user's randomized value $\mathring{V}_i$, his group G_i, and assigned query Q_i, the user's answer A_i is deterministic.

So, the probability $\Pr\left(A_i = a \middle| G_i = g, Q_i = q, \mathring{V}_i = \mathring{v}\right) = 1$ only for one realization of $\mathring{V}_i$ which we denote by $v^* = q(g, a)$, otherwise

$$\Pr\left(A_i = a \middle| G_i = g, Q_i = q, \mathring{V}_i = \mathring{v}\right) = 0.$$

Finally, (c) follows from (4.11). Substituting (C.2) in (C.1), we obtain

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

where we replaced $v^*, v'^* \in \mathcal{V}$ by $v, v' \in \mathcal{V}$.

3. **Accuracy:** We start by finding probabilities relating to the user's assigned queries. User i is assigned query $Q_i = q$, which is chosen uniformly at random from the set $\mathcal{Q}$ defined in (4.10). Therefore, for fixed row j and column a, the probability $\Pr(Q_i(j, a) = v) = \frac{1}{2m}$ for all $v \in \mathcal{V}$. Note that if user i's answer is A_i, and given his assigned query, the server maps the user's answer into the vector $Q_i(:, A_i)$. Given user i's group $G_i = g$ and group $V_i = v$, we find the distribution of $Q_i(j, A_i)$ for all $j \in [k]$.

That is, for all $j \neq g$, $j, g \in \mathcal{G}$, and $v, v' \in \mathcal{V}$, we have

$$\Pr\left(Q_i(j, A_i) = v' \middle| G_i = g, V_i = v\right) = \frac{1}{2m},$$

Otherwise, for all $j = g$, $j, g \in \mathcal{G}$, and $v, v' \in \mathcal{V}$,

$$\Pr\left(Q_i(g, A_i) = v' \middle| G_i = g, V_i = v\right) = \begin{cases} 1 - \lambda & \text{for } v' = v \\ \frac{\lambda}{2m-1} & \text{for } v' \in \mathcal{V} - \{v\}. \end{cases} \tag{C.3}$$

For all $i \in [n]$, we introduce the auxiliary random variables X_i and Y_i for ease of notation. For all $i \in [n]$, user i's group G_i and his value V_i are random variables as described in Section 4.2. We define an auxiliary random variable X_i that functions as an indicator for both the user's group and value. More precisely, X_i is a random k dimensional vector (where k is the number of groups), such that $X_i(j) = 0$ if $j \neq G_i$ and $X_i(j) = V_i$ if $j = G_i$.

Then one readily obtains, for all $j \in [k]$,

$$\Pr(X_i(j) = v) = \begin{cases} \theta_j p_j(v) & \text{for } v \in \mathcal{V}, \\ 1 - \theta_j & \text{for } v = 0, \end{cases} \tag{C.4}$$

and,

$$\mathbb{E}\left[X_i(j)\right] = \sum_{v \in \mathcal{V}} v p_j(v) \theta_j = \mathbb{E}[V_i | G_i = j]\theta_j. \tag{C.5}$$

Since the X_i's are i.i.d. for all $i \in [n]$, we have

$$\mathbb{E}\left[\left(\sum_{i \in [n]} X_i(j)\right)^2\right] = n\mathbb{E}[V_1^2 | G_1 = j]\theta_j + (n^2 - n)\mathbb{E}[V_1 | G_1 = j]^2 \theta_j^2. \tag{C.6}$$

For every user $i \in [n]$, we define an auxiliary random variable $Y_i = Q_i(:, A_i)$, which is a k dimensional random vector. Given user i's group $G_i = g_i$ and his value $V_i = v_i$, the $g_i{}^{\text{th}}$ coordinate of the vector Y_i contains user i's randomized value. All the other coordinates of the vector Y_i are randomly chosen from the alphabet $\mathcal{V}$. More precisely,

$$\Pr(Y_i(j) = v | G_i = g_i, V_i = v_i) = \begin{cases} 1 - \lambda & \text{for } v = v_i \text{ and } j = g_i, \\ \frac{\lambda}{2m-1} & \text{for } v \in \mathcal{V} - \{v_i\} \text{ and } j = g_i, \\ \frac{1}{2m} & \text{for } v \in \mathcal{V} \text{ and } j \neq g_i, \end{cases} \tag{C.7}$$

where $\lambda \in \left[0, 1 - \frac{1}{2m}\right)$. Then, following from (C.7), we obtain,

$$\Pr(Y_i(j) = v) = \theta_j \left((1 - \lambda)p_j(v) + \frac{(1 - p_j(v))\lambda}{2m - 1}\right) + \frac{(1 - \theta_j)}{2m}.$$

Then,

$$\mathbb{E}\left[Y_i(j)\right] = \frac{2m - 2m\lambda - 1}{2m - 1}\mathbb{E}[V_1 | G_1 = j]\theta_j, \tag{C.8}$$

Since $Y_1, \ldots, Y_n$ are i.i.d., we have

$$\mathbb{E}\left[\left(\sum_{i\in[n]} Y_i(j)\right)^2\right] = \frac{n(1-\theta_j)}{2m} + \frac{n\lambda\theta_j}{2m-1}\sum_{v\in\mathcal{V}} v^2 + (n^2-n)\left(\frac{2m-2m\lambda-1}{2m-1}\theta_j\mathbb{E}[V_1|G_1=j]\right)^2 + n\mathbb{E}[V_1^2|G_1=j]\theta_j\left(\frac{2m-2m\lambda-1}{2m-1}\right). \quad \text{(C.9)}$$

Note that $\sum_{v\in\mathcal{V}} v^2 = \frac{1}{3}m(m+1)(2m+1)$.

One readily obtains $\mathbb{E}\left[\hat{\mathbf{S}}_{\text{QA}} - \mathbf{S}\right] = 0$ by substituting (C.5) and (C.8), and observing that $X_1, \ldots, X_n$ are i.i.d. and $Y_1, \ldots, Y_n$ are i.i.d.. Then the estimator $\hat{\mathbf{S}}_{\text{QA}}$ is unbiased.

Next we calculate the expectation $\mathbb{E}\left[\sum_{i=1}^n Y_i(j)\sum_{l=1}^n X_l(j)\right]$ which will be helpful later in the proof. Notice that given user i's group, $G_i = g_i$, and value, $V_i = v_i$, the product $X_i(g_i)Y_i(g_i)$ is equal to v_i^2 with probability λ, and equal to $v_i v$ with probability $\frac{1-\lambda}{2m-1}$ for all $v \in \mathcal{V} - \{v_i\}$. And since $X_i(j) = 0$ for all $j \neq g_i$, then $X_i(j)Y_i(j) = 0$ for all $j \neq g_i$. Following these observations,

$$\mathbb{E}[X_i(j)Y_i(j)] = \frac{2m-2m\lambda-1}{2m-1}\theta_j\mathbb{E}[V_i^2|G_i=g], \quad \text{(C.10)}$$

which follows from $\sum_{v'\in\mathcal{V}-\{v\}} v' = -v$. Then,

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i=1}^n Y_i(j)\sum_{l=1}^n X_l(j)\right] \\
&\overset{\text{(a)}}{=} \sum_{i,l\in[n],i\neq l} \mathbb{E}\left[Y_i(j)\right]\mathbb{E}\left[X_l(j)\right] + \sum_{i=1}^n \mathbb{E}\left[Y_i(j)X_i(j)\right] \\
&\overset{\text{(b)}}{=} \frac{2m-2m\lambda-1}{2m-1}\Big[(n^2-n)\theta_j^2\mathbb{E}[V_1|G_1=j]^2 + n\theta_j\mathbb{E}[V_1^2|G_1=j]\Big]. \quad \text{(C.11)}
\end{aligned}$$

We have that (a) follows from that fact that if $i \neq l$, then $Y_i(j)$ is independent of $X_l(j)$. And (b) follows from substituting (C.5), (C.8), and (C.10). Then,

$$\mathcal{E}_{\text{QA}} = \frac{2m\lambda}{n(2m-2m\lambda-1)}\mathbb{E}[V_1^2] + \frac{(4m^2-1)(m+1)}{6n(2m-2m\lambda-1)}\left(\frac{(2m-1)k}{2m-2m\lambda-1} - 1\right), \quad \text{(C.12)}$$

follows from substituting (C.6), (C.9), and (C.11) in (4.1).

Now we have the exact expression of $\mathcal{E}_{\text{QA}}$ in terms of λ. To characterize the accuracy vs. privacy

trade-off, one might be interested in the expression of $\mathcal{E}_{\text{QA}}$ in terms of the privacy parameter ϵ. We can get a loose upper bound on $\mathcal{E}_{\text{QA}}$, as a function of ϵ, by substituting $\lambda = (2m-1)/(2m-1+e^{\epsilon})$, from Remark 5, in (C.12), and we get $\mathcal{E}_{\text{QA}} = \mathcal{O}\left(\frac{km^4}{n}\right)$.

C.1.2 On the Choice of Parameters for the Q&A scheme

We consider the class of schemes in which the user is assigned a random query matrix and in turn sends the server an answer, corresponding to a column of this matrix, based on his group and randomized value.

On the Choice of Queries

In the Q&A scheme we considered queries chosen uniformly at random from the set (4.10). Below we discuss variations on this set (4.10), and some of their disadvantages.

- Suppose the rows do not have all possible values from the set $\mathcal{V}$. Then, if the user's value does not exist in the corresponding row (group), he is not able to answer the query. For example, consider an assigned query matrix $\left[\begin{smallmatrix} -1 & -1 \\ +1 & -1 \end{smallmatrix}\right]$. This is asking a user of group 1 (row 1): "Which of these is your value: -1 or -1?". If the user has value $+1$, he cannot answer the question.

- If the rows have extra values that are not in $\mathcal{V}$, i.e., the set of possible user values, this can lead to breaking the privacy requirement because the server knows that no user will have that value. To clarify, consider the example of 2 groups and 2 values, i.e., $\mathcal{V} = \{-1, +1\}$. Let's assume the user is in group 1, has value $+1$, and is assigned the query matrix $\left[\begin{smallmatrix} +1 & -1 & +3 \\ +3 & -1 & +1 \end{smallmatrix}\right]$. If he sends the server the answer corresponding to column 1, the server knows with certainty that the user is in group 1, since he cannot assume the value $+3$. On the other hand, if both the additional numbers are in the same column, e.g., $\left[\begin{smallmatrix} +1 & -1 & +3 \\ +1 & -1 & +3 \end{smallmatrix}\right]$, then this column can effectively be removed because no user will answer it.

- One could also restrict to a subset of (4.10). Depending on the choice of subset this could perform equivalently or result in a biased estimator. For example, consider the case of 2 groups and 2 values. The query set defined in (4.10) for this case is

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

– We note that if a query is a result of permuting the columns of another query, then they convey the same information. For example, the following two queries, $\left[\begin{smallmatrix} -1 & +1 \\ +1 & -1 \end{smallmatrix}\right]$ and $\left[\begin{smallmatrix} +1 & -1 \\ -1 & +1 \end{smallmatrix}\right]$, are equivalent. This is because the server's interpretation of the user's answer will be the same irrespective of which of these two queries is assigned. Therefore, restricting to a subset $\mathcal{Q}' \subseteq \mathcal{Q}$, such that,

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

results in a scheme equivalent to Q&A.

– On the other hand, consider restricting to a subset $\mathcal{Q}' \subseteq \mathcal{Q}$, such that,

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

where the queries are chosen uniformly at random from $\mathcal{Q}'$. Then, the noise term in (4.6) will not have zero mean, and thus the estimate will be biased.

Therefore, a reduction to a subset of $\mathcal{Q}$ needs to be done carefully, and can complicate the definition in (4.10).

Randomizing the User's Value

One could consider a more general scheme in which the probability assigned to each possible random value ($v \in \mathcal{V}$ and $v \neq v_i$) is arbitrarily different. However, this can lead to a biased estimator, that cannot be unbiased without the knowledge of the parameters $p_g(v)$ and θ_g (which the server and user do not know).

Let us elaborate by considering the following modification to (11). Given his true value v, the user first chooses a randomized value $\mathring{v}$ according to the distribution

$$\Pr(\mathring{V}_i = \mathring{v} | V_i = v) = \begin{cases} \ell & \text{for } \mathring{v} = v \\ \ell_{v,\mathring{v}} & \text{for } \mathring{v} \in \mathcal{V} - \{v\}, \end{cases} \tag{C.13}$$

such that $\sum_{\mathring{v} \in \mathcal{V}, \mathring{v} \neq v} \ell_{v,\mathring{v}} = 1 - \ell$ for all $v \in \mathcal{V}$. Note that the ℓ and $\ell_{v,\mathring{v}}$ have to be constants, and

not a function of the problem parameters $p_g(v)$ or θ_g. Otherwise, because the user does not know these problem parameters, he would not be able to perform the randomization.

Following the same procedure described in the Accuracy discussion of Appendix C.1.1, we define the k-dimensional random vector $\widetilde{Y}_i$ whose coordinates are randomly chosen from the alphabet $\mathcal{V}$, such that,

$$\Pr(\widetilde{Y}_i(j) = v | G_i = g_i, V_i = v_i) = \begin{cases} \ell & \text{for } v = v_i \text{ and } j = g_i, \\ \ell_{v,\mathring{v}} & \text{for } v \in \mathcal{V} - \{v_i\} \text{ and } j = g_i, \\ \frac{1}{2m} & \text{for } v \in \mathcal{V} \text{ and } j \neq g_i. \end{cases}$$

Then, the expected value of $\widetilde{Y}_i(j)$, for all $i \in [n]$ and $j \in [k]$,

$$\mathbb{E}\left[\widetilde{Y}_i(j)\right] = \ell \mathbb{E}[V_i | G_i = j]\theta_j + \theta_j \sum_{v,\mathring{v} \in \mathcal{V}, v \neq \mathring{v}} \mathring{v} p_j(v) \ell_{v,\mathring{v}}. \tag{C.14}$$

Therefore, following from (C.5) and (C.14), we have

$$\begin{aligned} \mathbb{E}\left[\sum_{i\in[n]} \widetilde{Y}_i - \sum_{i\in[n]} X_i\right] &= \sum_{i\in[n]} \mathbb{E}[\widetilde{Y}_i - X_i] \\ &= n(\ell - 1)\mathbb{E}\left[\sum_{i\in[n]} X_i\right] + n\theta_g \sum_{v,\mathring{v}\in\mathcal{V}, v\neq\mathring{v}} \mathring{v} p_g(v) \ell_{v,\mathring{v}}. \end{aligned}$$

Notice that the server does not know the parameters $p_g(v)$ and θ_g for $g \in [k]$ and $v \in \mathcal{V}$. Therefore, the server, in general, cannot unbias the estimator. However, the Q&A scheme considers a special case of (C.13), where we have $\ell_{v,\mathring{v}} = \frac{\lambda}{2m-1}$ for all $v, \mathring{v} \in \mathcal{V}$. Therefore, we unbias the estimator by multiplying the aggregate by the constant $\frac{2m-1}{2m-2m\lambda-1}$, as we did in (4.12).

Moreover, there exists other special cases where the server can unbias the estimator without the knowledge of $p_g(v)$ and θ_g, for example if $\sum_{\mathring{v}\in\mathcal{V}, \mathring{v}\neq v} \mathring{v}\ell_{v,\mathring{v}} = 0$ for all $v \in \mathcal{V}$. We chose the first method for the Q&A scheme, as it simplifies its presentation, and is an adaptation of randomized response from the literature. The analysis of other methods, including their privacy analysis, remains open.

On an additional note, in general, the randomization of the user's value could be done such

that the incorrect values are chosen based on the mean square error of the scheme. However, this error is a function of the parameters θ_g and $p_g(v)$ for $g \in [k]$ and $v \in \mathcal{V}$ (see (4.4)). Therefore, implementing the scheme would require the user (and the server) to know these parameters, which is inconsistent with our model. Thus, the Q&A scheme does not depend on this error.

C.2 The Randomized Group (RG) Scheme

In Section C.2.1 of this appendix we prove Theorem 6, and in Section C.2.2 we prove Corollary 5.

C.2.1 Proof of Theorem 6

We separate the proof into three parts starting with communication, then privacy, and finally the accuracy.

1. **Communication**: Each user i sends the server an answer a_i, which is a 2 dimensional vector. The first coordinate has information about the user's group, i.e., $\mathring{g} \in \mathcal{G}$, and the second coordinate has information about the user's value, i.e., $\mathring{v} \in \mathcal{V}$. Therefore, to represent the user's answer, a_i, we need $\log(|\mathcal{V}|) + \log(|\mathcal{G}|) = 1 + \log(m) + \log(k)$ bits.

2. **Privacy**: To prove (4.19), we consider user i and look at the distribution $\Pr(A_i = a|G_i = g) = \Pr\left(\mathring{G}_i = g', \mathring{V}_i = v|G_i = g\right)$, for all $v \in \mathcal{V}$ and $g, g' \in \mathcal{G}$. We separately consider the two cases of $g' = g$ and $g' \neq g$ mirroring the two cases described in Section 4.5. For all $v \in \mathcal{V}$ and $g, g' \in \mathcal{G}$ such that $g' = g$,

$$\Pr\left(\mathring{G}_i = g, \mathring{V}_i = v|G_i = g\right) = \left((1 - \lambda_{vl})p_g(v) + \frac{(1 - p_g(v))\lambda_{vl}}{2m - 1}\right)(1 - \lambda_{gr}), \tag{C.15}$$

which follows from (4.16), (4.18), and

$$\Pr\left(\mathring{V}_i = v|\mathring{G}_i = g, G_i = g\right) = (1 - \lambda_{vl})p_g(v) + \frac{(1 - p_g(v))\lambda_{vl}}{2m - 1}$$

However, for all $v \in \mathcal{V}$ and $g, g' \in \mathcal{G}$ such that $g' \neq g$,

$$\Pr\left(\mathring{G}_i = g', \mathring{V}_i = v|G_i = g\right) = \frac{\lambda_{gr}}{2m(k - 1)}, \tag{C.16}$$

which follows from (4.16) and (4.17).

From Definition 5, we drop Q from the conditioning because there are no queries assigned to the users in this scheme, also the conditioning on P and Θ is implicit. Therefore, by substituting (C.15) and (C.16) in (4.2), we get

$$e^{\epsilon_{\text{RG}}} = \max\Bigg\{\rho \max_{g\in\mathcal{G},v\in\mathcal{V}} p_g(v)(2m(1-\lambda_{vl})-1)+\lambda_{vl}, \\ \left(\rho \min_{g\in\mathcal{G},v\in\mathcal{V}} p_g(v)(2m(1-\lambda_{vl})-1)+\lambda_{vl}\right)^{-1}\Bigg\},$$

where $\rho = \frac{2m(k-1)(1-\lambda_{gr})}{\lambda_{gr}}$.

3. **Accuracy**: For all $i \in [n]$, user i sends the server the answer $A_i = \left(\mathring{G}_i, \mathring{V}_i\right)$, where the user's randomized group, $\mathring{G}_i$, is described in (4.16), and his randomized value, $\mathring{V}_i$, is described in equations (4.18) and (4.17).

We define an auxiliary random variable Z_i that functions as an indicator for both user i's randomized group and randomized value. More precisely, Z_i is a random k dimensional vector (where k is the number of groups), such that $Z_i(j) = V_i$ if $j = \mathring{G}_i$ and $Z_i(j) = 0$ otherwise, i.e., $j \neq G_i$. For all $j \in [k]$ and $v \in \mathcal{V}$, one readily obtains

$$\Pr(Z_i(j)=v) = \frac{(1-\theta_g)\lambda_{gr}}{2m(k-1)} \\ + \theta_g(1-\lambda_{gr})\left((1-\lambda_{vl})p_j(v) + \frac{(1-p_j(v))\lambda_{vl}}{2m-1}\right). \quad \text{(C.17)}$$

Since $Z_1, Z_2, \ldots, Z_n$ are i.i.d., then following from (C.17) for all $j \in [k]$ and $i \in [n]$ the expectation

$$\mathbb{E}\left[\sum_{i\in[n]} Z_i(j)\right] = \frac{n\theta_g(1-\lambda_{gr})(2m(1-\lambda_{vl})-1)}{2m-1}\mathbb{E}[V_1|G_1=j]. \quad \text{(C.18)}$$

Then, $\mathbb{E}\left[\hat{\mathbf{S}}_{\text{RG}} - \hat{\mathbf{S}}_{\text{QA}}\right] = 0$, which follows from (C.5) and (C.18). Thus, the estimator of the RG scheme is unbiased. Moreover,

$$\mathbb{E}\left[\left(\frac{2m-1}{(1-\lambda_{gr})(2m(1-\lambda_{vl})-1)}\sum_{i=1}^{n} Z_i(j)\right)^2\right]$$
$$= \frac{n(4m^2-1)[2m\theta_g(k-1)(1-\lambda_{gr})\lambda_{vl}+\lambda_{gr}(2m-1)(1-\theta_g)]}{6(k-1)(1-\lambda_{gr})^2(2m(1-\lambda_{vl})-1)^2(m+1)^{-1}}$$
$$+\frac{n\theta_g(2m-1)}{(1-\lambda_{gr})(2m(1-\lambda_{vl})-1)}\mathbb{E}[V_1^2|G_1=j]$$
$$+(n^2-n)\theta_g^2\mathbb{E}[V_1|G_1=j]^2. \quad \text{(C.19)}$$

Consider the random variables X_i, for all $i \in [n]$, described in (C.4). For all $i \in [n]$ and $j \in [k]$, notice that the product $X_i(j)Z_i(j)$ can take on one of these values:

$$X_i(j)Z_i(j) = \begin{cases} v^2 & \forall v \in \mathcal{V}, \text{ if } X_i(j) = Z_i(j) = v, \\ vv' & \forall v, v' \in \mathcal{V}, v' \neq v, \text{ if } X_i(j) = v, \text{ and } Z_i(j) = v', \\ 0 & \forall v, v' \in \mathcal{V}, \text{ if } X_i(j) = 0 \text{ or } Z_i(j) = 0, \end{cases}$$

Then, we can use this to find the expectation

$$\mathbb{E}\left[X_i(j)Z_i(j)\right] = \frac{2m(1-\lambda_{vr})-1}{2m-1}\mathbb{E}\left[V_i^2|G_i=j\right]\theta_j(1-\lambda_{gr}). \quad \text{(C.20)}$$

Moreover, since $Z_1, \ldots, Z_n$ are i.i.d., $X_1, \ldots, X_n$ are i.i.d., and Z_i is independent of X_ℓ if $i \neq \ell$,

$$\mathbb{E}\left[\sum_{i\in[n]} Z_i(j)\sum_{\ell\in[n]} X_\ell(j)\right]$$
$$= \sum_{i\in[n]}\mathbb{E}\left[Z_i(j)X_i(j)\right] + (n^2-n)\sum_{i,\ell\in[n],i\neq\ell}\mathbb{E}\left[Z_i(j)\right]\left[X_\ell(j)\right]$$
$$= \frac{(n^2-n)(1-\lambda_{gr})(2m(1-\lambda_{vl})-1)}{2m-1}\mathbb{E}[V_1|G_1=j]^2\theta_g^2$$
$$+n(1-\lambda_{gr})(1-\lambda_{vl})\mathbb{E}[V_1^2|G_1=j]\theta_g], \quad \text{(C.21)}$$

which follows from the substitution of (C.5), (C.18), and (C.20).

Notice that $\hat{\mathbf{S}}_{\text{RG}}(j) = \frac{2m-1}{(1-\lambda_{gr})(2m(1-\lambda_{vl})-1)}\sum_{i=1}^{n} Z_i(j)$, and $\mathbf{S}(j) = \sum_{i=1}^{n} X_i(j)$. This implies

in

$$\mathcal{E}_{\text{RG}} = \frac{2m\lambda_{vl}(1-\lambda_{gr}) + \lambda_{gr}(2m-1)}{n(1-\lambda_{gr})(2m(1-\lambda_{vl})-1)} \left(\mathbb{E}[V_1^2] + \frac{(4m^2-1)(m+1)}{6(1-\lambda_{gr})(2m(1-\lambda_{vl})-1)} \right), \tag{C.22}$$

which follows from substituting (C.6), (C.19), (C.21) in (4.1), and noting that $\sum_{v\in\mathcal{V}} v^2 = \frac{1}{3}m(m+1)(2m+1)$.

This proves how we obtained the expression of $\mathcal{E}_{\text{RG}}$ as a function of λ_{gr} and λ_{vl}. Moreover, one could be interested in the error as a function of a given required privacy $\epsilon > 0$. We give an upper bound of $\mathcal{E}_{\text{RG}}$ as a function of ϵ. We substitute λ_{gr} and λ_{vl} that minimize the error from Corollary 5, in (C.22). Then, the error $\mathcal{E}_{\text{RG}}$ is upper bounded by $\mathcal{O}\left(\frac{m^4k^2}{ne^\epsilon}\right)$.

C.2.2 Proof of Corollary 5

We first assume that $p_{\max} > \frac{1}{2m}$ and $p_{\min} < \frac{1}{2m}$, and consider the special case of $p_{\max} = p_{\min} = \frac{1}{2m}$ separately in the end. For ease of notation define

$$\begin{aligned} &f(\lambda_{gr}, \lambda_{vl}) = \text{MSE}(\hat{\mathbf{S}}_{\text{RG}}) \\ &= \frac{2m\lambda_{vl}(1-\lambda_{gr}) + \lambda_{gr}(2m-1)}{(1-\lambda_{gr})(2m(1-\lambda_{vl})-1)} \left(\mathbb{E}[V_1^2] + \frac{(4m^2-1)(m+1)}{6(1-\lambda_{gr})(2m(1-\lambda_{vl})-1)} \right) n \end{aligned}$$

which follows directly from (4.20). To minimize the error of the RG scheme we solve the following optimization problem,

$$\begin{aligned} \underset{\lambda_{vl},\lambda_{gr}}{\text{minimize}} \quad & f(\lambda_{gr}, \lambda_{vl}) \\ \text{subject to} \quad & e^\epsilon = \max\left\{ \frac{2m(k-1)(1-\lambda_{gr})(p_{\max}(2m(1-\lambda_{vl})-1)+\lambda_{vl})}{\lambda_{gr}(2m-1)}, \right. \\ & \qquad \left. \frac{\lambda_{gr}(2m-1)}{2m(k-1)(1-\lambda_{gr})(p_{\min}(2m(1-\lambda_{vl})-1)+\lambda_{vl})} \right\}, \\ & 0 \le \lambda_{vl} < \tfrac{2m-1}{2m}, 0 < \lambda_{gr} < 1. \end{aligned} \tag{C.23}$$

To solve it, we consider two optimization problems. Consider this first optimization problem, assume its optimal value is attained, and let $\lambda_{gr}^{(1)}$ and $\lambda_{vl}^{(1)}$ be its optimal points,

$$\begin{aligned}
\underset{\lambda_{vl},\lambda_{gr}}{\text{minimize}} \quad & f(\lambda_{gr},\lambda_{vl}) \\
\text{subject to} \quad & e^{\epsilon} = \tfrac{2m(k-1)(1-\lambda_{gr})(p_{\max}(2m(1-\lambda_{vl})-1)+\lambda_{vl})}{\lambda_{gr}(2m-1)}, \\
& \tfrac{2m(k-1)(1-\lambda_{gr})(p_{\max}(2m(1-\lambda_{vl})-1)+\lambda_{vl})}{\lambda_{gr}(2m-1)} \\
& \qquad\qquad \geq \tfrac{\lambda_{gr}(2m-1)}{2m(k-1)(1-\lambda_{gr})(p_{\min}(2m(1-\lambda_{vl})-1)+\lambda_{vl})}, \\
& 0 \leq \lambda_{vl} < \tfrac{2m-1}{2m}, 0 < \lambda_{gr} < 1.
\end{aligned} \tag{C.24}$$

Consider this second optimization problem, assume its optimal value is attained, and let $\lambda_{gr}^{(2)}$ and $\lambda_{vl}^{(2)}$ be its optimal points,

$$\begin{aligned}
\underset{\lambda_{vl},\lambda_{gr}}{\text{minimize}} \quad & f(\lambda_{gr},\lambda_{vl}) \\
\text{subject to} \quad & e^{\epsilon} = \tfrac{\lambda_{gr}(2m-1)}{2m(k-1)(1-\lambda_{gr})(p_{\min}(2m(1-\lambda_{vl})-1)+\lambda_{vl})}, \\
& \tfrac{\lambda_{gr}(2m-1)}{2m(k-1)(1-\lambda_{gr})(p_{\min}(2m(1-\lambda_{vl})-1)+\lambda_{vl})} \\
& \qquad\qquad \geq \tfrac{2m(k-1)(1-\lambda_{gr})(p_{\max}(2m(1-\lambda_{vl})-1)+\lambda_{vl})}{\lambda_{gr}(2m-1)}, \\
& 0 \leq \lambda_{vl} < 1 - \frac{1}{2m}, 0 < \lambda_{gr} < 1.
\end{aligned} \tag{C.25}$$

Then, the solution of (C.23) is $\min\left\{f\left(\lambda_{gr}^{(1)},\lambda_{vl}^{(1)}\right), f\left(\lambda_{gr}^{(2)},\lambda_{vl}^{(2)}\right)\right\}$. Therefore, to solve (C.23), we first solve (C.24) and (C.25).

- *Solution of* (C.24)*:* Since $0 \leq \lambda_{vl} < \frac{2m-1}{2m}$, we have the following two cases.

  - If $e^{2\epsilon} < \frac{p_{\max}}{p_{\min}}$, from the first condition directly follows

$$\lambda_{gr}^{(1)} = \tfrac{2m(k-1)(p_{\max}(2m(1-\lambda_{vl})-1)+\lambda_{vl})}{2m(k-1)(p_{\max}(2m(1-\lambda_{vl})-1)+\lambda_{vl})+e^{\epsilon}}. \tag{C.26}$$

    Since $e^{2\epsilon} < \frac{p_{\max}}{p_{\min}}$, then $(2m-1)(p_{\max} - p_{\min}e^{2\epsilon}) > 0$, and by substituting (C.26) in the conditions of (C.24), we get

$$0 < \tfrac{(2m-1)p_{\max}-(2m-1)p_{\min}e^{2\epsilon}}{(1-2mp_{\min})e^{2\epsilon}+2mp_{\max}-1} \leq \lambda_{vl} < \tfrac{1}{2}.$$

Since $f(\lambda_{gr}^{(1)}, \lambda_{vl})$ is increasing in λ_{vl}, then its minimum is at achieved at the boundary of the domain, i.e.,

$$\lambda_{vl}^{(1)} = \frac{(2m-1)p_{\max}-(2m-1)p_{\min}e^{2\epsilon}}{(1-2mp_{\min})e^{2\epsilon}+2mp_{\max}-1}.$$

Then substituting this back in (C.26), we get

$$\lambda_{gr}^{(1)} = \frac{2m(k-1)(p_{\max}-p_{\min})e^{\epsilon}}{2m(k-1)(p_{\max}-p_{\min})e^{\epsilon}+(1-2p_{\min})e^{2\epsilon}+2mp_{\max}-1}.$$

- If $e^{2\epsilon} \geq \frac{p_{\max}}{p_{\min}}$, then $(2m-1)(p_{\max} - p_{\min}e^{2\epsilon}) \leq 0$, and

$$\frac{(2m-1)p_{\max}-(2m-1)p_{\min}e^{2\epsilon}}{(1-2mp_{\min})e^{2\epsilon}+2mp_{\max}-1} \leq 0 \leq \lambda_{vl} < \frac{1}{2}.$$

Similarly, since $f(\lambda_{gr}^{(1)}, \lambda_{vl})$ is increasing in λ_{vl}, then $\lambda_{vl}^{(1)} = 0$. And we have that, $\lambda_{gr}^{(1)} = \frac{2m(k-1)p_{\max}}{2m(k-1)p_{\max}+e^{\epsilon}}$,

- *Solution of* (C.25)*:* The solution of (C.25) follows similarly as that of (C.24), and we have the following two cases.

- If $e^{2\epsilon} < \frac{p_{\max}}{p_{\min}}$, $\lambda_{vl}^{(2)} = \frac{(2m-1)p_{\max}-(2m-1)p_{\min}e^{2\epsilon}}{(1-2mp_{\min})e^{2\epsilon}+2mp_{\max}-1}$, and

$$\lambda_{gr}^{(2)} = \frac{2m(k-1)(p_{\max}-p_{\min})e^{\epsilon}}{2m(k-1)(p_{\max}-p_{\min})e^{\epsilon}+(1-2p_{\min})e^{2\epsilon}+2mp_{\max}-1}.$$

- If $e^{2\epsilon} \geq \frac{p_{\max}}{p_{\min}}$, then $\lambda_{vl}^{(2)} = 0$, and $\lambda_{gr}^{(2)} = \frac{2m(k-1)p_{\min}e^{\epsilon}}{2m(k-1)p_{\min}e^{\epsilon}+1}$.

- *Combining the two solutions:* We also have to look at the two cases separately as follows.

- If $e^{2\epsilon} < \frac{p_{\max}}{p_{\min}}$, the solution is straightforward. The optimal points for (C.23) are $\lambda_{vl}^{*} = \frac{(2m-1)p_{\max}-(2m-1)p_{\min}e^{2\epsilon}}{(1-2mp_{\min})e^{2\epsilon}+2mp_{\max}-1}$, and

$$\lambda_{gr}^{*} = \frac{2m(k-1)(p_{\max}-p_{\min})e^{\epsilon}}{2m(k-1)(p_{\max}-p_{\min})e^{\epsilon}+(1-2p_{\min})e^{2\epsilon}+2mp_{\max}-1}.$$

- If $e^{2\epsilon} \geq \frac{p_{\max}}{p_{\min}}$, one readily obtains $f\left(\lambda_{vl}^{(1)}, \lambda_{gr}^{(1)}\right) \leq f\left(\lambda_{vl}^{(2)}, \lambda_{gr}^{(2)}\right)$. Therefore, for this case, the optimal points for (C.23) are $\lambda_{vl}^{*} = 0$ and $\lambda_{gr}^{*} = \frac{2m(k-1)p_{\max}}{2m(k-1)p_{\max}+e^{\epsilon}}$.

This completes the proof for $p_{\max} > \frac{1}{2m}$. If $p_{\max} = p_{\min} = \frac{1}{2m}$, then the second condition of (C.23) reduces to

$$e^{\epsilon} = \max\left\{\frac{(k-1)(1-\lambda_{gr})}{\lambda_{gr}}, \frac{\lambda_{gr}}{(k-1)(1-\lambda_{gr})}\right\}.$$

i.e., e^{ϵ} is not a function of λ_{vl}. Therefore, for this case, the optimal points for (C.23) can be readily obtained such that $\lambda_{vl}^* = 0$ and $\lambda_{gr}^* = \frac{k-1}{k-1+e^{\epsilon}}$. Combining all the described cases completes the proof.

Remark 10. In Corollary 5, we minimize the relative error subject to a fixed privacy parameter ϵ. Because of the monotonicity of the relative error as a function of ϵ, an increase in privacy, i.e., smaller ϵ, cannot decrease the error. Thus, minimizing the error subject to

$$e^{\epsilon} \leq \max\left\{\frac{2m(k-1)(1-\lambda_{gr})(p_{\max}(2m(1-\lambda_{vl})-1)+\lambda_{vl})}{\lambda_{gr}(2m-1)}, \frac{\lambda_{gr}(2m-1)}{2m(k-1)(1-\lambda_{gr})(p_{\min}(2m(1-\lambda_{vl})-1)+\lambda_{vl})}\right\},$$

is equivalent to solving the optimization (C.23).

C.3 Proof of Theorem 7

We start by sketching the proof of (i) in Theorem 7.

- For $k = 2$, $m = 1$, and $p_1(v) \neq p_2(v')$ or $p_1(v) = p_2(v) = 0.5$ for all $v, v' \in \mathcal{V} = \{-1, 1\}$, we can easily find the exact value of λ that satisfies (4.3); therefore, we can find the expression for the error of the Q&A scheme $\mathcal{E}_{\text{QA}}(\epsilon, b)$. Moreover, the minimum error of the RG scheme $\mathcal{E}_{\text{RG}}(\epsilon, b)$ follows from Corollary 5. We find that the limit of the difference of the errors, $\mathcal{E}_{\text{RG}}(\epsilon, b) - \mathcal{E}_{\text{QA}}(\epsilon, b)$, as ϵ goes to zero, is positive.

- For $k > 2$ and $m = 1$ or $k \geq 2$ and $m > 1$, the minimum error of the RG scheme $\mathcal{E}_{\text{RG}}(\epsilon, b)$ follows from Corollary 5. From Remark 5, to guarantee a required privacy ϵ, we can choose any $\lambda \geq \frac{(2m-1)p_{\max} - p_{\min}e^{\epsilon}}{2m(p_{\max} - p_{\min}e^{\epsilon}) + e^{\epsilon} - 1}$. We use this λ to bound the error of the Q&A scheme. Finally, we find that the bound on the limit of the difference of the errors, $\mathcal{E}_{\text{RG}}(\epsilon, b) - \mathcal{E}_{\text{QA}}(\epsilon, b)$, as ϵ goes to zero, is positive.

Now we prove (ii) of Theorem 7 by showing that there exists an $\epsilon_\ell > 0$, such that for all $\epsilon > \epsilon_\ell$, we have $\mathcal{E}_{\text{QA}}(\epsilon, b) > \mathcal{E}_{\text{RG}}(\epsilon, b)$. We first consider the Q&A scheme. From Remark 5, there exits an

$\epsilon_0 > 0$, such that for all $\epsilon > \epsilon_0$, the parameter $\lambda = 0$ guarantees privacy level ϵ_0. And the error of the Q&A scheme, as defined in (4.21), for $\lambda = 0$, i.e., all $\epsilon > \epsilon_0$, is

$$\mathcal{E}_{\text{QA}}(\epsilon, b) = \frac{1}{6b} \log(2m)(2m+1)(m+1)\,(k-1) > 0.$$

Let $\epsilon_\ell > \epsilon_0 > \sqrt{\frac{p_{\max}}{p_{\min}}}$, then from Corollary 5, the parameters $\lambda^*_{vl} = 0$ and

$$\lambda^*_{gr} = \frac{2m(k-1)p_{\max}}{2m(k-1)p_{\max} + e^{\epsilon_\ell}}$$

minimize the error of the RG scheme. Thus, there exists ϵ_ℓ, such that for all $\epsilon > \epsilon_\ell$,

$$\mathcal{E}_{\text{QA}}(\epsilon_\ell, b) - \mathcal{E}_{\text{RG}}(\epsilon_\ell, b) = \frac{\log(2m)(2m+1)(m+1)\,(k-1)}{6b},$$

which is greater than zero and this completes the proof.

ACKNOWLEDGMENT OF PREVIOUS PUBLICATIONS

P1 C. Naim, F. Ye, S. El Rouayheb, “ON-OFF Privacy with Correlated Requests”, IEEE International Symposium on Information Theory (ISIT), 2019.

P2 F. Ye, C. Naim, S. El Rouayheb, “Preserving ON-OFF Privacy for Past and Future Requests”, Information Theory Workshop (ITW), 2019.

P3 F. Ye, C. Naim, S. El Rouayheb, “ON-OFF Privacy Against Correlation Over Time", IEEE Transactions on Information Forensics and Security, 2021.

P4 C. Naim, R.G.L. D’Oliveira, S. El Rouayheb, “Private Multi-Group Aggregation", IEEE International Symposium on Information Theory (ISIT), 2021.

P5 F. Ye, C. Naim, S. El Rouayheb, “ON-OFF Privacy in the Presence of Correlation", IEEE Transactions on Information Theory, 2021.

P6 C. Naim, R.G.L. D’Oliveira, S. El Rouayheb, “Private Multi-Group Aggregation", IEEE Journal on Selected Areas in Communication, 2022.

REFERENCES

[1] L. Sweeney, "K-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, Oct. 2002.

[2] C. Dwork, "Differential privacy: A survey of results," in *International conference on theory and applications of models of computation*, Apr. 2008, pp. 1–19.

[3] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," *Journal of the ACM*, vol. 45, no. 6, pp. 965–981, Nov. 1998.

[4] K. Bonawitz *et al.*, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, Oct. 2017, pp. 1175–1191.

[5] Differential Privacy Team, "Learning with privacy at scale differential," *Apple*, Dec. 2017.

[6] "Regulation (EU) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/EC (general data protection regulation)," *Official Journal L119*, pp. 1–88, May 2016.

[7] *The california consumer privacy act of 2018*, https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375, Jun. 2018.

[8] 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.

[9] C. Naim, F. Ye, and S. El Rouayheb, "ON-OFF privacy with correlated requests," in *IEEE International Symposium on Information Theory (ISIT)*, Jul. 2019.

[10] F. Ye, C. Naim, and S. El Rouayheb, "ON-OFF privacy in the presence of correlation," *IEEE Transactions on Information Theory*, vol. 67, no. 11, pp. 7438–7457, Nov. 2021.

[11] F. Ye, C. Naim, and S. El Rouayheb, "Preserving ON-OFF privacy for past and future requests," *IEEE Information Theory Workshop (ITW)*, Aug. 2019.

[12] F. Ye, C. Naim, and S. El Rouayheb, "ON-OFF privacy against correlation over time," *IEEE Transactions on Information Forensics and Security*, vol. 16, no. 1, pp. 2104–2117, 2021.

[13] C. Naim, R. G. L. D'Oliveira, and S. El Rouayheb, "Private multi-group aggregation," in *IEEE International Symposium on Information Theory (ISIT)*, Jul. 2021.

[14] C. Naim, R. G. L. D'Oliveira, and S. El Rouayheb, "Private multi-group aggregation," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 3, pp. 800–814, Mar. 2022.

[15] J. So, B. Güler, and A. S. Avestimehr, "Turbo-aggregate: Breaking the quadratic aggregation barrier in secure federated learning," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 1, pp. 479–489, Mar. 2021.

[16] S. Kadhe, N. Rajaraman, O. O. Koyluoglu, and K. Ramchandran, "FastSecAgg: Scalable secure aggregation for privacy-preserving federated learning," in *ICML Workshop on Federated Learning for User Privacy and Data Confidentiality*, Jul. 2020.

[17] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1142–1154, Feb. 2022.

[18] T.-H. H. Chan, E. Shi, and D. Song, "Optimal lower bound for differentially private multi-party aggregation," in *Proceedings of the 20th Annual European Conference on Algorithms*, Sep. 2012, pp. 277–288.

[19] 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)*, Jun. 2020.

[20] 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, Oct. 2017.

[21] S. Truex *et al.*, "A hybrid approach to privacy-preserving federated learning," in *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, Nov. 2019, pp. 1–11.

[22] E. Shi, T.-H. H. Chan, E. Rieffel, R. Chow, and D. Song, "Privacy-preserving aggregation of time-series data," in *Network & Distributed System Security Symposium (NDSS)*, Feb. 2011.

[23] 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*, Oct. 2013, pp. 429–438.

[24] S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *Journal of the American Statistical Association*, vol. 60, pp. 63–69, Mar. 1965.

[25] L. Sankar, S. R. Rajagopalan, and H. V. Poor, "Utility-privacy tradeoffs in databases: An information-theoretic approach," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 838–852, Mar. 2013.

[26] Y. Gertner, Y. Ishai, E. Kushilevitz, and T. Malkin, "Protecting data privacy in private information retrieval schemes," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, May 1998, pp. 151–160.

[27] M. Bezzi, "An information theoretic approach for privacy metrics," *Transactions on Data Privacy*, vol. 3, no. 3, pp. 199–215, Dec. 2010.

[28] E. Nekouei, T. Tanaka, M. Skoglund, and K. H. Johansson, "Information-theoretic approaches to privacy in estimation and control," *Annual Reviews in Control*, vol. 47, pp. 412–422, Jan. 2019.

[29] F. du Pin Calmon and N. Fawaz, "Privacy against statistical inference," in *2012 50th annual Allerton conference on communication, control, and computing (Allerton)*, Oct. 2012, pp. 1401–1408.

[30] S. Kadhe, B. Garcia, A. Heidarzadeh, S. El Rouayheb, and A. Sprintson, "Private information retrieval with side information: The single server case," *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Oct. 2017.

[31] S. Li and M. Gastpar, "Single-server multi-message private information retrieval with side information," in *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Oct. 2018, pp. 173–179.

[32] N. B. Shah, K. V. Rashmi, and K. Ramchandran, "One extra bit of download ensures perfectly private information retrieval," in *IEEE International Symposium on Information Theory (ISIT)*, Jun. 2014.

[33] H. Sun and S. A. Jafar, "The capacity of private information retrieval," *IEEE Transaction on Information Theory*, vol. 63, no. 7, pp. 4075–4088, Mar. 2017.

[34] R. Tajeddine, O. W. Gnilke, and S. El Rouayheb, "Private information retrieval from MDS coded data in distributed storage systems," *IEEE Transactions on Information Theory*, vol. 64, no. 11, pp. 7081–7093, Nov. 2018.

[35] R.Freij-Hollanti, O. W. Gnilke, C. Hollanti, and D. A. Karpuk, "Private information retrieval from coded databases with colluding servers," *SIAM Journal on Applied Algebra and Geometry*, vol. 1, no. 1, pp. 647–664, 2017.

[36] K. Banawan and S. Ulukus, "The capacity of private information retrieval from coded databases," *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1945–1956, Mar. 2018.

[37] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: Architecture and algorithms," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 1–18, 2007.

[38] R. Shokri, C. Troncoso, C. Diaz, J. Freudiger, and J.-P. Hubaux, "Unraveling an old cloak: K-anonymity for location privacy," in *Proceedings of the 9th Annual ACM Workshop on Privacy in the Electronic Society*, 2010, pp. 115–118.

[39] J. Hua, W. Tong, F. Xu, and S. Zhong, "A geo-indistinguishable location perturbation mechanism for location-based services supporting frequent queries," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1155–1168, 2018.

[40] K. Chatzikokolakis, C. Palamidessi, and M. Stronati, "A predictive differentially-private mechanism for mobility traces," in *International Symposium on Privacy Enhancing Technologies Symposium*, 2014, pp. 21–41.

[41] Y. Xiao and L. Xiong, "Protecting locations with differential privacy under temporal correlations," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1298–1309.

[42] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux, "Quantifying location privacy," in *2011 IEEE Symposium on Security and Privacy*, 2011, pp. 247–262.

[43] R. Shokri, G. Theodorakopoulos, and C. Troncoso, "Privacy games along location traces: A game-theoretic framework for optimizing location privacy," *ACM Transactions on Privacy Security*, vol. 19, no. 4, 2017.

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

[45] W. Zhang, M. Li, R. Tandon, and H. Li, "Online location trace privacy: An information theoretic approach," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 1, pp. 235–250, 2019.

[46] F. Ye, H. Cho, and S. El Rouayheb, "Mechanisms for hiding sensitive genotypes with information-theoretic privacy," in *2020 IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 902–907.

[47] P. M. Vaidya, "Speeding-up linear programming using fast matrix multiplication," in *30th Annual Symposium on Foundations of Computer Science*, Oct. 1989, pp. 332–337.

[48] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows*. John Wiley & Sons, 2011.

[49] 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.

[50] M. Abadi *et al.*, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[51] 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.

[52] M. Kim, O. Günlü, and R. F. Schaefer, "Federated learning with local differential privacy: Trade-offs between privacy, utility, and communication," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 2650–2654.

[53] J. Konečný and P. Richtárik, "Randomized distributed mean estimation: Accuracy vs. communication," *Frontiers in Applied Mathematics and Statistics*, vol. 4, p. 62, 2018.

[54] A. T. Suresh, F. X. Yu, S. Kumar, and H. B. McMahan, "Distributed mean estimation with limited communication," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[55] L. P. Barnes, Y. Han, and A. Özgür, "Lower bounds for learning distributions under communication constraints via fisher information," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 9583–9612, Jan. 2020.

[56] I. Diakonikolas, E. Grigorescu, J. Li, A. Natarajan, K. Onak, and L. Schmidt, "Communication-efficient distributed learning of discrete distributions," in *Advances in Neural Information Processing Systems*, 2017.

[57] 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.

[58] M. Ye and A. Barg, "Optimal schemes for discrete distribution estimation under local differential privacy," in *IEEE International Symposium on Information Theory (ISIT)*, 2017.

[59] I. Diakonikolas, M. Hardt, and L. Schmidt, "Differentially private learning of structured discrete distributions," in *Advances in Neural Information Processing Systems 28*, 2015.

[60] J. Acharya, Z. Sun, and H. Zhang, "Hadamard response: Estimating distributions privately, efficiently, and with little communication," in *The 22nd International Conference on Artificial Intelligence and Statistics*, 2019.

[61] S. Wang *et al.*, "Mutual information optimally local private discrete distribution estimation," *arXiv:1607.08025*, Jul. 2016.

[62] Ú. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: Randomized aggregatable privacy-preserving ordinal response," in *Proceedings of the 21st ACM Conference on Computer and Communications Security*, 2014.

[63] J. Acharya and Z. Sun, "Communication complexity in locally private distribution estimation and heavy hitters," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

[64] W. Zhu, P. Kairouz, H. Sun, B. McMahan, and W. Li, "Federated heavy hitters with differential privacy," in *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

[65] W.-N. Chen, P. Kairouz, and A. Özgür, "Breaking the communication-privacy-accuracy trilemma," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Dec. 2020.

[66] C. Niu *et al.*, "Billion-scale federated learning on mobile clients: A submodel design with tunable privacy," ser. MobiCom '20, Apr. 2020.

[67] M. Kim and J. Lee, "Information-theoretic privacy in federated submodel learning," *ICT Express*, Feb. 2022.

[68] Z. Jia and S. A. Jafar, "X-secure T-private federated submodel learning," in *ICC 2021 - IEEE International Conference on Communications*, Jun. 2021, pp. 1–6.