# Side-Channel Attacks and New Principles in the Shuffle Model of Differential Privacy

Shaowei Wang, Jin Li, Changyu Dong, Jin Li, Zhili Zhou, *Senior Member, IEEE*, Di Wang, *Member, IEEE*, and Zikai Wen

***Abstract*—The shuffle model employs a shuffler to anonymize and permute user messages, thereby enhancing privacy/utility trade-offs compared to the local model. Ideally, it assumes perfect message anonymity protection against adversaries, allowing each user to hide among a large population. However, in contexts like mobile/edge networks or in scenarios where the shuffler is curious, this assumption is frequently unrealistic. In this study, we demonstrate the vulnerability of the shuffle model to communication side-channel attacks, which substantially compromise privacy amplification via shuffling. We categorize side-channel information in the shuffle model into three types: (i) in-out information, revealing the victim user's participation and timing, (ii) message-cardinality information, indicating the victim's message count, and (iii) message-length information, disclosing the victim's message length(s). Numerical results indicate these attacks increase privacy loss by 200% to 4100%, revealing secret value with probability more than 90%. After theoretically analyzing the remaining privacy amplification effects, we suggest several countermeasures and principles to alleviate degradation caused by these attacks: (a) appending padding bits to each message to counter message-length attacks, (b) maximizing query parallelization to elude in-out attacks and increase the population for privacy amplification, and (c) sending dummy messages to exchange communication costs for improved privacy amplification effects. The newly proposed paradigms and principles significantly save privacy budget in comparison to current models under attack.**

***Index Terms*—Differential privacy, shuffle model, side channels, privacy attacks.**

Received 14 April 2024; revised 3 December 2024 and 4 May 2025; accepted 7 May 2025. Date of publication 26 May 2025; date of current version 3 July 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62372120, Grant 62472116, Grant 62372125, Grant 62261160651, Grant 62102108, Grant U23A20307, and Grant U21A20463; in part by Guangdong Natural Science Funds for Distinguished Young Scholar under Grant 2023B1515020041; in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515030273 and Grant 2022A1515010061; in part by the Foundation of Yunnan Key Laboratory of Service Computing under Grant YNSC24115; and in part by the Science and Technology Projects in Guangzhou under Grant 2025A03J3182 and Grant 202201010194. The associate editor coordinating the review of this article and approving it for publication was Prof. Luisa Verdoliva. *(Corresponding authors: Zhili Zhou; Zikai Wen.)*

Shaowei Wang is with the School of Artificial Intelligence, Guangzhou University, Guangzhou 511370, China, and also with Guangdong Key Laboratory of Blockchain Security, Guangzhou University, Guangzhou 511370, China (e-mail: wangsw@gzhu.edu.cn).

Jin Li, Changyu Dong, Jin Li, and Zhili Zhou are with the School of Artificial Intelligence, Guangzhou University, Guangzhou 511370, China (e-mail: zhou_zhili@163.com).

Di Wang is with the Division of CEMSE, King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia.

Zikai Wen is with the School of Engineering and Technology, University of Washington, Tacoma, WA 98402 USA (e-mail: zkwen@uw.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TIFS.2025.3571661, provided by the authors.

Digital Object Identifier 10.1109/TIFS.2025.3571661

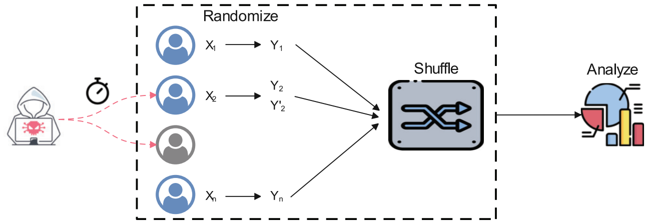

Fig. 1. An illustration of the shuffle model under communication side-channel attacks (the shuffler can be an honest-but-curious adversary).

## I. Introduction

THE shuffle model of differential privacy (DP) [1] has emerged as a compelling approach to data privacy protection, combining the benefits of the classical central model [2] (i.e., relatively high data utility) and the local model [3], [4], [5], [6], [7], [8], [9] (i.e., minimal trust in other parties). In the shuffle model, an intermediary shuffler anonymizes and randomly permutes messages from a user population before forwarding them to the server (e.g., data analysts). As the shuffling operation is data-agnostic and can be executed over ciphertext space, numerous parties can assume the shuffler's role in practice, including anonymous communication networks [10], secure hardware [11], [12], and other cryptographic tools [13]. Moreover, since potential privacy adversaries (e.g., the server) only observe anonymized and shuffled messages, each user can hide within a large population. Consequently, adding a minimal amount of noise to local messages sufficiently protects data privacy in the released view of shuffled messages. This phenomenon is known as *privacy amplification via shuffling* [1]. For example, in the single-message shuffle model where each user sends one message satisfying local $\epsilon_0$-differential privacy (LDP), the shuffled messages from $n$ users actually preserve $\tilde{O}(\sqrt{e^{\epsilon_0}/n})$-DP [14]; in the multi-message shuffle model where each user sends multiple messages, as long as these messages are carefully calibrated, the overall shuffled messages can achieve DP. Due to its potential to achieve excellent privacy-utility-efficiency trade-offs in decentralized settings, the shuffle model has been applied across various domains, including count/summation queries [15], [16], [17], [18], [19], [20], [21], [22], [23] and machine learning [24], [25], [26], [27], [28], [29], [30], and was deployed in Apple and Google's Exposure Notification Privacy-preserving Analytics [31].

Despite the success of the shuffle model in decentralized private data analysis, it relies on several unrealistic security assumptions in such environments. Ideally, privacy adversaries in the shuffle model are assumed to lack intermediate information about a victim user during protocol execution, having only access to the crowd's shuffled messages. We argue that this assumption is easily violated in mobile computing, edge computing, or cable networking environments where the shuffle model is applied (see Figure 1 for an illustration). For example, in wireless/cellular networks, privacy adversaries can precisely infer a victim user's communication activities at a negligible cost [32], [33], [34], [35], [36]. Adversaries can effortlessly deduce the victim's communication timing with the shuffler, potentially ascertain the message payload size, and infer the number of messages contributed by the victim at high precision (e.g., with over 90% accuracy as reported in [33] and [34]). *The situation is even worse when the intermediate shuffler is curious*: the shuffler observes almost certain side information about messages from each user (e.g., message cardinality, approximate message lengths, and the communication timing of a user), even if each message is encrypted by the public key of the server/analyzer.

### A. Our Contributions

In this work, we analyze the side-channel vulnerabilities of the shuffle DP model and propose mitigation strategies with formal guarantees. Despite many efforts in the literature on communication side-channel analyses (especially for anonymous channels [37], [38], [39]), we are the first to provide formal analyses in the differential privacy sense. Additionally, due to differences in privacy goals, we demonstrate that conventional countermeasures (e.g., sending dummy messages [40]) for anonymous networks can induce infinite DP loss or be inefficient in the shuffle model. The contributions of this work are as follows:

*1) Privacy Degradation Analyses Due to Side-Channel Information:* We demonstrate that side-channel information, easily inferred by privacy adversaries, significantly impacts privacy amplification effects. Intuitively, a victim user's communication timing reveals the specific task (e.g., the specific gradient descent iteration in federated learning) they participated in, allowing the victim to hide only among the sub-population of that particular task, which comprises only several tens or hundreds of users. If privacy adversaries know the length of the victim's contributed message, they can filter out many other messages with different lengths in the shuffled messages, substantially reducing the number of messages/users the victim can hide among. In certain multi-message protocols, the number of messages sent by a user can be directly related to the user's true value. Consequently, the knowledge of message cardinality by privacy adversaries completely undermines privacy amplification through shuffling. Numerical results on typical settings show that communication timing (termed as in-out attacks) and message-length attacks increase the victim user's differential privacy loss by 200% to 4100%, revealing the victim's secret value with probability more than 90%; for several state-of-the-art multi-message protocols, the message-cardinality attack increases privacy loss to $+\infty$, revealing the victim's secret value with probability more than 95%.

*2) Defending Against Side-Channel Attacks:* We conducted formal analyses of privacy degradation in widely-used shuffle models (e.g., shuffle-then-randomize model [1], [14], [41] and randomize-then-shuffle model [15], [24], [43]) under side-channel attacks and developed new shuffle models resistant to such attacks. In the proposed model, users make their own participation choices, reducing trust assumptions regarding the shuffler/server. To minimize privacy loss due to participation timing, the original data analysis task is first represented as a directed acyclic graph (DAG). Based on this, multiple independent queries are issued simultaneously, while sequential dependent queries are addressed using dummy messages from users, ensuring indistinguishable participation timings. To counteract privacy loss resulting from message payload size/length, each user's message is padded to a fixed length (as the traditonal defensive methods in the literature [44] for anonymou channels). To address severe privacy leaks from message cardinality where traditional defenses fail, domain transformation can be employed specific to the multi-message protocol. These techniques form new principles for the shuffle models under side-channel attacks. Furthermore, even in settings without attacks, our proposed multinomial-randomize-shuffle model provides much stronger privacy amplification effects than existing models.

We summarize the results of in-out/message-length attacks and defenses in Table I. The amplification population size denotes the number of users a victim can hide among - a critical parameter in privacy amplification via shuffling, where higher values indicate better amplification effects. The variables include: $n$, the full population size; $n_k$, the size of the population in the same $k$-th batch as the victim user; $n_{*,l}$, the size of the population with the same message length as the victim user; $n_{k,l}$, the size of the population with the same message length and in the same $k$-th batch as the victim user; and $n_{(m)}$, the size of the population that selected the same $m$-th bin as the victim user. The *normal* column denotes the amplification population size when there are no side-channel attacks; the *in-out attacks* column represents the amplification size under in-out attacks; the *in-out & length atk.* column signifies the amplification size under joint in-out and message-length attacks. The *adaptive query* column indicates whether the model supports sequentially adaptive queries. The *extra costs* column highlights the additional communication costs compared to the normal model.

### B. Organization

The remainder of this paper is organized as follows. Section II reviews related work on the shuffle model and side channels. Section III provides background knowledge. Section IV identifies several side-channel attacks and numerically analyzes their impact on privacy. Section V presents theoretical analyses of the privacy degradation impact of side-channel information. Section VI proposes defense methods. Section VII recaps new requirements for the shuffle model, and discusses the implication for the broader DP-Crypto systems. Finally, Section VIII concludes the paper.

## II. Related Work

### A. Side-Channel Attacks and Defenses

Side-channel attacks exploit unintended information leakage through a system's physical properties or observable character-

TABLE I

COMPARISON OF THE AMPLIFICATION POPULATION SIZE AND OTHER PROPERTIES OF VARIOUS SHUFFLE MODELS UNDER IN-OUT AND MESSAGE-LENGTH ATTACKS. CONVENTIONALLY, N $\geq n_{(m)} \gg ;n_k; \geq n_{k,l} \gg 1$, INDICATING DISPARATE AMPLIFICATION EFFECTS OF VARIOUS MODELS UNDER ATTACKS

| **shuffle models** | **normal setting** | **in-out attacks** | **in-out & length atk.** | **adaptive queries** | **extra costs** |
|---|---|---|---|---|---|
| shuffle-then-randomize<br>[1], [14], [41][Definition 5] | $n$ | 1 | 1 | ✓ | N.A. |
| divide-randomize-shuffle<br>[25]–[30] [Definition 6] | $n_k$ | $n_k$ | $n_{k,l}$ | ✓ | N.A. |
| subsample-randomize-shuffle<br>[24], [42] [Definition 7] | $n$ | $n_k$ | $n_{k,l}$ | ✓ | N.A. |
| parallel-randomize-shuffle<br>[1], [43][Algorithm 3] | $n$ | $n$ | $n_{*,l}$ | ✗ | N.A. |
| multinomial-randomize-shuffle<br>**this work** [Definition 11] | $n$ | $n_k$ | $n_{k,l}$ | ✓ | N.A. |
| rectified parallel-randomize-shuffle<br>**this work** [Algorithm 4] | $n$ | $n$ | $n$ | ✗ | padding bits per message |
| rectified multinomial-randomize-shuffle<br>**this work** [Algorithm 5] | $n$ | $n$ | $n$ | ✓ | padding bits per message<br>$K$-time dummy messages |
| rectified bin-randomize-shuffle<br>**this work** [Algorithm 6] | $n_{(m)}$ | $n_{(m)}$ | $n_{(m)}$ | ✓ | padding bits per message<br>few dummy messages |

istics, such as electromagnetic radiation, power consumption, and communication patterns.

*1) Side Channels of Anonymous Networks:* Anonymous networks [10] aim to protect users' identities and relationships between communicating parties. Researchers have investigated various side-channel attacks that target anonymous channels, including traffic analysis attacks [37], timing attacks [38], and intersection attacks [39]. These attacks exploit information leaks from communication patterns, message timings, or user behavior to undermine the anonymity guarantees provided by the channels. As countermeasures, researchers propose strategies like padding messages, traffic camouflage [44], sending dummy messages [40] and disturbing sending times [45] in anonymous networks.

This work presents the first side-channel attack/defense study to joint systems of cryptographic tools (i.e., the shuffling/anonymous network) and information-theoretic privacy tools. We provide formal privacy analyses of side-channel attacks in the shuffle model within the DP context. We also propose defense methods incorporating both classical countermeasures (e.g., padding messages and sending dummy messages [40]) and tailored approaches. Specifically, due to the differences in privacy goals, we demonstrate that certain classical countermeasures for anonymous networks can induce infinity DP loss (see Section VI-D) or being inefficient (see Section VI-C) in the shuffle model.

### *B. Shuffle Model of Differential Privacy*

The shuffle model of DP anonymizes user messages through an intermediate shuffler before sending them to a server for analytics. The model's foundation lies in privacy amplification analysis [15], [41], [43], ensuring the global privacy level of shuffled messages. Depending on the number of messages a user can send, the shuffle model can be categorized as either multi-message [21], [46] or single-message [1], [15], [41].

Various shuffle model variants exist in the literature. For convenient theoretical analysis of privacy amplification, studies such as [1], [14], and [41] utilize the shuffle-then-randomize model, in which user data are first shuffled and then fed into adaptive local randomizers sequentially. Practical approaches to support multiple adaptive queries in decentralized setting is to divide the user population into non-overlapping parts (e.g., in [25] and [27], referred to as the divide-randomize-shuffle model), or letting the shuffler to perform user subsampling (e.g., in [24] and [42], referred to as the subsample-randomize-shuffle model). Another approach is to let each user randomly select (at most) one query to check in [16], and then choose one user (from checked-in users) for each query slot.

We demonstrate that aforementioned shuffle models are all susceptible to side-channel attacks. Their privacy amplification degrades to the intra-batch level or even to the local level (i.e., no privacy amplification), refer to Table I for summarized results. Worse still, for several protocols in these shuffle models, such as SOTA multi-message summation protocols [17], [19], [46], [47], the differential privacy loss under side-channel attack increases to $+\infty$.

### *C. Security Attacks of Differential Privacy*

Differential privacy, as the de facto standard for data privacy, is widely adopted in industry for sensitive databases [48] and decentralized data collection/analyses [49]. Although DP provides rigorous data privacy in theory, its practical implementation may encounter unexpected security issues. For example, improper implementations of floating-point numbers for the prevalent Laplace/Gaussian mechanism compromise the intended DP [50], [51]. There can also be side-channel timing/global state attacks in central DP database engines [52]. Besides, a small portion of adversarial users may completely undermine the aggregation results of locally private protocols [53] and shuffle private protocols [18]. There also might be collusion between the shuffler and other parties [54]. To the best of our knowledge, this work is the first to study side-channel security issues of decentralized DP, and challenges the honest-and-not-curious assumption on the shuffler in the shuffle model. As comparison, in the local model of DP

TABLE II

LIST OF NOTATIONS

| Notation | Description |
|---|---|
| $[i]$ | $\{1, 2, ..., i\}$ |
| $[i:j]$ | $\{i, i+1, ..., j\}$ |
| $n$ | the number of users (data owners) |
| $K$ | the number of sequential queries |
| $U_k$ | the users participated in the $k$-th round query |
| $\mathbb{X}$ | the domain of input data |
| $\mathbb{Y}$ | the domain of a single message |
| $\mathbb{Z}$ | the domain of an analyzing algorithm |
| $\epsilon_0$ | the local privacy budget |
| $\delta$ | the failure/violating probability |
| $\mathcal{S}$ | the shuffling procedure |
| $\mathcal{R}$ | the randomization algorithm |
| $\mathcal{A}$ | the data analyzing algorithm |
| $io_v$ | the participation information of user $v$ |
| $num_v$ | the number of messages from user $v$ |
| $len(y)$ | the message length (bits) of message $y$ |

[3], where each user trusts no other party, there is limited vulnerability to side-channel attacks, as the plaintext data leaving the user/device is publishable, and thus side-channel information related to it is also publishable.

## III. Preliminaries

This section presents definitions of differential privacy and shuffle models. Notations are listed in Table II.

### A. Divergences and Differential Privacy

*Definition 1 (Hockey-stick divergence)* The Hockey-stick divergence between two random variables $P$ and $Q$ is:

$$D_{e^\epsilon}(P\|Q) = \int \max\{0, P(x) - e^\epsilon Q(x)\}\mathrm{d}x,$$

where $P$ and $Q$ denote both the random variables and their probability density functions.

Two variables $P$ and $Q$ are $(\epsilon, \delta)$-indistinguishable if $\max\{D_{e^\epsilon}(P\|Q),\ D_{e^\epsilon}(Q\|P)\} \leq \delta$. For two datasets of equal size that differ only by a single individual's data, they are referred to as *neighboring datasets*. Differential privacy limits the divergence of query results on neighboring datasets (see Definition 2). Similarly, in the local setting that accepts a single individual's data as input, we introduce the local $(\epsilon, \delta)$-differential privacy in Definition 3. When $\delta = 0$, the concept is abbreviated as $\epsilon$-LDP.

*Definition 2 (Differential privacy [2])* A protocol $\mathcal{R} : \mathbb{X}^n \mapsto \mathbb{Z}$ satisfies $(\epsilon, \delta)$-differential privacy if, for all neighboring datasets $X, X' \in \mathbb{X}^n$, $\mathcal{R}(X)$ and $\mathcal{R}(X')$ are $(\epsilon, \delta)$-indistinguishable.

*Definition 3 (Local differential privacy [3])* A protocol $\mathcal{R} : \mathbb{X} \mapsto \mathbb{Y}$ satisfies local $(\epsilon, \delta)$-differential privacy if, for all $x, x' \in \mathbb{X}$, $\mathcal{R}(x)$ and $\mathcal{R}(x')$ are $(\epsilon, \delta)$-indistinguishable.

### B. The Classic Shuffle Model

This part reviews the classic one-round shuffle model.

*1) Single-Message Shuffle Model:* Following the conventions of the randomize-then-shuffle model [15], [55], we define a single-message protocol $\mathcal{P}$ as a list of algorithms $\mathcal{P} = (\{\mathcal{R}_i\}_{i\in[n]}, \mathcal{A})$, where $\mathcal{R}_i : \mathbb{X} \to \mathbb{Y}$ is user $i$'s local randomizer, and $\mathcal{A} : \mathbb{Y}^n \to \mathbb{Z}$ is the analyzer on the data collector's side. The overall protocol implements a mechanism $\mathcal{P} : \mathbb{X}^n \to \mathbb{Z}$ as follows. User $i$ holds a data record $x_i$ and a local randomizer $\mathcal{R}_i$, then computes a message $y_i = \mathcal{R}_i(x_i)$. The messages $y_1, \ldots, y_n$ are shuffled and submitted to the analyzer. We denote the shuffling step as $\mathcal{S}(y_1, \ldots, y_n)$, where $\mathcal{S} : \mathbb{Y}^n \to \mathbb{Y}^n$ is a *shuffler* that applies a uniform-random permutation to its inputs. In summary, the output of $\mathcal{P}(x_1, \ldots, x_n)$ is represented by $\mathcal{A} \circ \mathcal{S} \circ \mathcal{R}_{[n]}(X) = \mathcal{A}(\mathcal{S}(\mathcal{R}_1(x_1), \ldots, \mathcal{R}_n(x_n)))$.

*2) Multi-Message Shuffle Model:* In contrast to sending a single message, the multi-message shuffle model allows each user to release multiple messages to the shuffler. The $\mathcal{R}_i : \mathbb{X} \to \mathbb{Y}^*$ is user $i$'s local randomizer. The output $\mathcal{P}(x_1, \ldots, x_n)$ of the overall protocol is $\mathcal{A} \circ \mathcal{S} \circ \mathcal{R}_{[n]}(X) = \mathcal{A}(\mathcal{S}(\mathcal{R}_1(x_1) \cup \mathcal{R}_2(x_2) \cup \cdots \cup \mathcal{R}_n(x_n)))$.

The shuffle model strives to ensure the privacy of $\mathcal{P}(x_1, \ldots, x_n)$ for any analyzer $\mathcal{A}$. Owing to the post-processing property of Hockey-stick divergence, guaranteeing that the shuffled messages $\mathcal{S} \circ \mathcal{R}_{[n]}(X)$ exhibit differential privacy suffices. We formally delineate differential privacy in the shuffle model in Definition 4.

*Definition 4 (DP in the shuffle model)* A protocol $\mathcal{P} = (\{\mathcal{R}_i\}_{i\in[n]}, \mathcal{A})$ satisfies $(\epsilon, \delta)$-DP in the shuffle model iff for all neighboring datasets $X$ and $X' \in \mathbb{X}^n$, the $\mathcal{S} \circ \mathcal{R}_{[n]}(X)$ and $\mathcal{S} \circ \mathcal{R}_{[n]}(X')$ are $(\epsilon, \delta)$-indistinguishable.

### C. Prevalent Variants of Shuffle Model

Several variants of shuffle model exist in the literature, depending on the organization of users to respond to multiple adaptive queries.

*1) Shuffle-Then-Randomize Model:* We revisit the ideal (single-message) shuffle model based on shuffle-then-randomize [1], [14], [41]. Given an input dataset $X = \{x_1, \ldots, x_n\}$, a uniform-random permutation $\pi : [n] \mapsto [n]$ is first applied to obtain $\mathcal{S}(X) = \{x_{\pi^{-1}(1)}, \ldots, x_{\pi^{-1}(n)}\}$, followed by a series of adaptive randomizers $\{\mathcal{R}_i\}_{i\in[n]}$. The $i$-th randomizer $\mathcal{R}_i$ takes the $i$-th datum $x_{\pi^{-1}(i)}$ in $\mathcal{S}(X)$ and the previous $i-1$ randomization results as input (see Definition 5). Since $\pi$ is not revealed to the server, each user's message can hide among all $n$ messages. Specifically, when every $\mathcal{R}_i$ satisfies $\epsilon_0$-LDP (not unnecessarily identical), the messages $\mathcal{R}_{[n]} \circ \mathcal{S}(X)$ satisfy $(\tilde{O}(e^{\epsilon_0/2}/\sqrt{n}), \delta)$-DP [14], [41]. This model serves as an ideal one that supports fully adaptive queries while providing the strongest privacy amplification effects (amplified by $n$ users).

*Definition 5* (Shuffle-then-randomize model [1], [14], [41]) Let $\mathcal{R}_i : \mathbb{Z}_0 \times \mathbb{Y}_1 \times \cdots \times \mathbb{Y}_{i-1} \times \mathbb{X} \to \mathbb{Y}_i$ for $i \in [n]$ be a sequence of algorithms, where $\mathbb{Z}_0$ denotes the range space of global information. A protocol $\mathcal{P}_{s\text{-}r} : \mathbb{Z}_0 \times \mathbb{X}^n \to \mathbb{Y}_0 \times \cdots \times \mathbb{Y}_n$ with global information $z_0$ in the shuffle-then-randomize model proceeds as follows: given a dataset $x_{[n]} \in \mathbb{X}^n$, it samples a uniform-random permutation $\pi$ and then sequentially computes $z_i = \mathcal{R}_i(z_{[0:i-1]}, x_{\pi^{-1}(i)})$ for $i \in [n]$ before outputting $z_{[0:n]}$.

*2) Divide-Randomize-Shuffle Model:* In decentralized settings, a more realistic approach is letting the analyzer divide users into multiple groups and employ each group for one query sequentially, see Definition 6. This model is adopted by many works for federated learning [27], and for bandit/reinforcement learning [25], [28], [29], [30]. Since the division is known to the analyzer (i.e., a potential adversary), a user $i \in U_k$ can only hide among users in the same group, thus privacy can only be amplified by $|U_k|$. This is considerably weaker than the ideal shuffle model.

*Definition 6 (Divide-randomize-shuffle model [25], [26])* Let $U_{[K]}$ denote a division of set $[n]$ such that $U_k \cap U_{k'} = \emptyset$ for all $k, k' \in [K]$ that $k \neq k'$, and $U_1 \cup \cdots \cup U_K = [n]$. Let $\mathcal{R}_{(k)} : \mathbb{Z}_0 \times \mathbb{Z}_1 \times \cdots \times \mathbb{Z}_{k-1} \times \mathbb{X} \to \mathbb{Y}_(k)$ denote the randomizer in the $k$-th round, where $\mathbb{Z}_k = \mathbb{Y}_{(k)}^{|U_k|}$ and $\mathbb{Y}_{(k)}$ is the range space of $\mathcal{R}_{(k)}$. The $z_{(0)} \in \mathbb{Z}_0$ is global information. A protocol $\mathcal{P}_{d\text{-}r\text{-}s} : \mathbb{Z}_0 \times \mathbb{X}^n \to \mathbb{Z}_0 \times \cdots \times \mathbb{Z}_K$ in the divide-randomize-shuffle model operates as follows: given a dataset $x_{[n]} \in \mathbb{X}^n$, it sequentially computes $z_{(i)} = \mathcal{S}(\mathcal{R}_{(k)}(z_{(0:k-1)}, x_i)_{i \in U_k})$ and outputs $z_{(0)}, z_{(1)}, \ldots, z_{(K)}$ along with $U_{[K]}$.

*3) Subsample-Randomize-Shuffle Model:* Rather than letting the analyzer (i.e., a potential adversary) to perform user division, a more practical method involves having the shuffler randomly sample users in each round independently, as illustrated in Definition 7. This approach is commonly employed for federated learning, as seen in [24] and [42].

*Definition 7* (Subsample-randomize-shuffle model [24], [42]) Let $\{U_k\}_{k \in [K]}$ denote a list of subsets that each has size $s$ and is uniform-randomly sampled from $[n]$ without replacement. Let $\mathcal{R}_{(k)} : \mathbb{Z}_0 \times \mathbb{Z}_1 \times \cdots \times \mathbb{Z}_{k-1} \times \mathbb{X} \to \mathbb{Y}_(k)$ denote the randomizer in the $k$-th, where $\mathbb{Z}_k = \mathbb{Y}_{(k)}^{|U_k|}$and $\mathbb{Y}_{(k)}$ is the range space of $\mathcal{R}_{(k)}$. The $z_{(0)} \in \mathbb{Z}_0$ is global information. A protocol $\mathcal{P}_{d\text{-}r\text{-}s} : \mathbb{Z}_0 \times \mathbb{X}^n \to \mathbb{Z}_0 \times \cdots \times \mathbb{Z}_K$ in the subsample-randomize-shuffle model operates as follows: given a dataset $x_{[n]} \in \mathbb{X}^n$, it independently samples $\{U_k\}_{k \in [K]}$ as described previously, then sequentially computes $z_{(i)} = \mathcal{S}(\mathcal{R}_{(k)}(z_{(0:k-1)}, x_i)_{i \in U_k})$ and outputs $z_{(0)}, \ldots, z_{(K)}$.

Given that only a relatively small batch of users, with a size of $s$, is randomly selected for each round, privacy is also amplified by subsampling [56]. Assuming that local randomizers satisfy $\epsilon_0$-LDP, and when combined with privacy amplification via shuffling and the advanced composition theorem of differential privacy, the overall privacy loss is $\left(\tilde{O}\left(\frac{s}{n}\sqrt{Ke^{\epsilon_0}/s}\right), \delta\right)$ [24], [42].

## IV. Side-Channel Attacks in the Shuffle Model

In this part, we show privacy amplification is severely damaged by side-channel knowledge, which can be readily divulged to privacy adversaries in decentralized settings.

### A. Threat Model of Side-Channel Attacks

We assume that the internal shuffling process $\mathcal{S}$ is perfectly secure, with adversaries only able to observe the victim users' communication activities (with the shuffler) and the output $\mathcal{P}(X)$ of the shuffle model. Typically, every user in the shuffle model encrypts their messages $\mathcal{R}_i(x_i)$ using the analyzer's public key before sending them to the shuffler. Privacy adversaries (including a potential adversary, the curious shuffler) do not have access to the values of these encrypted messages, but only to their side information, such as communication timing, the number of messages, and message length(s).

The aim of privacy adversaries is to deduce the secret $x_i$ of victim users, utilizing either the success probability of inferring $x_i$ or the differential privacy loss of $x_i$ as a metric.

### B. Categories of Side-Channel Information

In the shuffle models described above, we define the following three types of side-channel information about the victim user $v$ ($v \in [n]$):

I. (*In-out information*) In-out information indicates the victim's participation in multiple rounds of a shuffle protocol. We denote it as $io_v \in \{0, 1\}^K$, where the $k$-th value $io_v(k)$ is 1 if the victim participated in the $k$-th round, and 0 otherwise.

II. (*Message-cardinality information*) In a multi-message protocol, message-cardinality information indicates the number of messages from the victim. We denote this as $num(\mathcal{R}_v(x_v))$, representing the number of messages output by $\mathcal{R}_v(x_v)$.

III. (*Message-length information*) Message-length information indicates the length of the victim's message. We denote this as $len(\mathcal{R}_v(x_v))$, representing the number of bits in $\mathcal{R}_v(x_v)$. In a multi-message protocol, this represents the length of each message from $\mathcal{R}_v(x_v)$.

We argue that in decentralized settings, such as mobile computing and wireless networks, privacy adversaries can easily infer in-out, message-cardinality, and message-length information. The in-out information is strongly correlated with the communication patterns of the victim user, which are almost public information in wireless networks or cellular networks [57], [58]. Adversaries can sniff the number of packets sent from the victim to the shuffler through network activities and by examining packet headers, even when packets are delivered over the prevalent HTTPS [59]. Furthermore, the length of TCP/IP packets from the victim can be monitored by sniffing network traffic [60], [61], [62]. Relatively, the in-out information is easier to be inferred. The side-channel information pertaining to message number or length might often be imprecise. This lack of accuracy is predominantly due to the implementation of techniques such as packet padding and packet slicing by modern wireless or cellular networking protocols. Additionally, this imprecision is posed by modern encryption schemes, which encrypt payloads on a block-by-block basis. Notably, when the shuffler is a curious party, since the shuffler observes encrypted message(s) from each individuals, the leaked side-channel information can be quite precise, regardless of the communication media between each user and the shuffler.

Note that in-out information is especially vulnerable in interactive queries with substantial download overheads, such as federated machine learning. Users download the latest model parameters only when participating in a round. The correlated heavy-loaded downloading traffic, along with uploading traffic to the shuffler, significantly increases the risk of in-out information exposure.

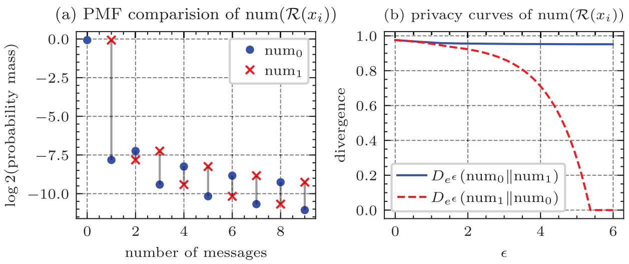

Fig. 2. The probability mass distributions of message numbers $num_0 = num(\mathcal{R}(0))$, $num_1 = num(\mathcal{R}(1))$ and privacy curves due to message-cardinality attack on [19].

### C. Privacy Under Message-Cardinality Attacks

Message-cardinality attacks occur in multi-message protocols, where the number of messages a user sends to the shuffler might depend on the actual value the user holds (e.g., in [17], [19], [27], [46], [47], and [63]). If message cardinality is observed by adversaries, privacy can be compromised.

**Algorithm 1** $\Delta$-Summation Randomizer [19]

```
1 if $x_i \neq 0$ then
2     Send $x_i$
3 Sample $z_i^{+1}, z_i^{-1} \sim \mathsf{NB}(1, e^{-(1-\gamma)\epsilon})_{/n}$
4 Send $z_i^{+1}$ copies of $+1$, and $z_i^{-1}$ copies of $-1$
5 for $\mathbf{s} \in S$ do
6     Sample $z_i^{\mathbf{s}} \sim \mathsf{NB}(3(1+\log(2/\delta)), e^{-0.1\min(1,\gamma\epsilon)/4})_{/n}$
7     for $m \in \mathbf{s}$ do
8         Send $z_i^{\mathbf{s}}$ copies of $m$
```

**Algorithm 2** $\Delta$-Summation Analyzer [19]

```
1 $T \leftarrow$ multiset of messages received
2 return $\sum_{y \in T} y$
```

As an example, we show a message-cardinality attack to the state-of-the-art $\Delta$-summation protocol [19]. In this protocol, each user holds an integer $x_i \in [0, \Delta]$, and the analyzer obtains the sum satisfying $(\epsilon, \delta)$-DP. The algorithms of the local randomizer and the analyzer are shown in Algorithm 1 and 2. The local randomizer randomizes the user's input in three steps: (1) each user $i$ sends $x_i$ if it is non-zero (line 1-2); (2) it sends unary noise messages (i.e. $z_i^{+1}$ copies of $+1$ and $z_i^{-1}$ copies of $-1$ following negative binomial distributions) whose sum is equal to the Discrete Laplace noise commonly used algorithms in the central DP model (line 3-4); (3) it defines a sub-collection $S$ of all multisets of $\{-\Delta, \cdots, +\Delta\}\backslash\{0\}$ whose sum of elements is equal to zero (e.g. $\{-1, -1, 2\}$), and for each multiset $s \in S$, $z_i^{\mathbf{s}}$ is sampled and $z_i^{\mathbf{s}}$ copies of every element in $s$ are sent (line 5-8). The analyzer simply sums up all messages.

For simplicity, let us consider the case $\Delta = 1$. Now the zero-sum messages collection $S$ simply contains one message set, i.e. $S = \{\{-1, 1\}\}$. Let $Z^1 \sim \mathsf{NB}(1, e^{-(1-\gamma)\epsilon})_{/n}$, $Z^2 \sim \mathsf{NB}(1, e^{-(1-\gamma)\epsilon})_{/n}$, and $Z^3 \sim \mathsf{NB}(3(1 + \log(2/\delta)), e^{-0.1\min(1,\gamma\epsilon)/4})_{/n}$, then when $x_i = 0$, the total number of messages is $num_0 = Z^1 + Z^2 + 2Z^3$ where $Z^1 + Z^3$ is the number of $+1$, and $Z^2 + Z^3$ is the number of $-1$. On the other hand, when $x_i = 1$, the total number of messages is $num_1 = 1 + Z^1 + Z^2 + 2Z^3$ because the local randomize also sends $x_i$ in addition to the noise messages. Our observation is that $num_0$ can be 0 and $num_1$ cannot, and in typical settings, $num_b = b$ with a large probability. In short, the number of messages exposes the user's private data value. To demonstrate this, in Figure 2(a), we plot the probability distributions of $num_0$, $num_1$ under the following parameters: $n = 10^4$, $\epsilon = 1.0$, $\delta = 0.01/n$, and $\gamma = 0.1$. As we can see in the figure, $\mathbb{P}[num_b = b] \geq 95.1\%$ for $b \in \{0, 1\}$. So if an adversary can observe the message cardinality, it can deduce almost certainly the user's private data value. A similar conclusion holds for multi-message protocols in [17], [27], [46], [47], and [63].

### D. Privacy Under In-Out Attacks

This section focuses on the privacy degradation of shuffle models under in-out attacks. We start with the ideal shuffle model: shuffle-then-randomize, and then give results about the prevalent subsample-randomize-shuffle model.

*1) Shuffle-Then-Randomize Model Under In-Out Attacks [No Privacy Amplification]:* The shuffle-then-randomize model, introduced by the seminal work [1] and refined by recent studies [14], [41], provides the best results to date in terms of privacy amplification. In this model, there are multiple rounds. In each round, the analyzer sends a query to the shuffler, who then chooses a user to answer the query according to a random permutation $\pi$. The $i$-th user answers the query in the $\pi(i)$-th round by sending the locally randomized data to the shuffler. The query and the local randomizer may vary in each round.

In this model, privacy amplification can be achieved because the users can hide themselves in a crowd while answering queries. However, if the adversary can observe a user participating in a particular round (i.e. an in-out attack), privacy is then reduced to whatever the local randomizer can provide and amplification is no longer achievable. We can demonstrate the difference through an example. Consider a scenario in which an analyzer queries a crowd of $n = 10{,}000$ users, and each of them holds a bit. In each round, the analyzer asks a user to submit her bit through the shuffler. Each user can use a local randomized response mechanism to perturb her bit before submitting it to the shuffler:

$$\mathcal{R}_i(x_i) = \begin{cases} x_i, & \text{with probability } 1-p; \\ 1-x_i, & \text{with probability p.} \end{cases}$$

where $p = \frac{1}{e^{\epsilon_0}+1}$ and $\epsilon_0$ is the local privacy budget. Suppose the global privacy goal is $(\epsilon = 0.2, \delta = 10^{-6})$-DP, then the local level only needs to be $\epsilon_0 = 2.81$ based on the analysis in [14]. An in-out attack means that the adversary can recover the user's data much easier: the $i$-th user submitted her data in the $\pi(i)$-th round, which is observed by the adversary. Since the local randomizer's budget is $\epsilon_0 = 2.81$, the probability $p \approx 0.057$. Hence, the adversary knows that with a high probability $1 - p = 0.943$, the $i$-th user's bit is the one submitted in the $\pi(i)$-th round.

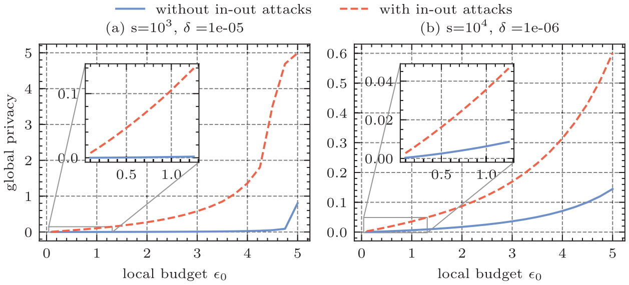

Fig. 3. The damage of in-out attack on privacy amplification of protocol [24] in subsample-randomize-shuffle model.

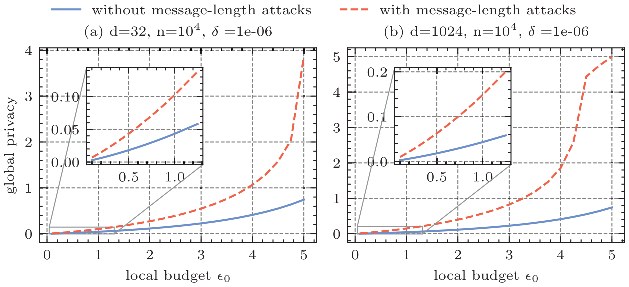

Fig. 4. The damage of message length attack on privacy amplification of the single-message protocol [1].

Similar attacks apply to binary randomized response used in [1] for streaming data aggregation, generalized randomized response used by [14], [41] for histogram estimation, or any other local randomizers in the shuffle-then-randomize model.

*2) Subsample-Randomize-Shuffle Model Under in-Out Attacks [Intra-Batch Amplification]:* In the subsample-randomize-shuffle model, a randomly sampled sub-population report their data in each round. Privacy amplification in this model relies on the fact that the analyzer does not know who participated in which round, hence can amplify privacy also by subsampling [56]. However, if an adversary possesses the victim user's in-out information, the victim's privacy will be weakened significantly. Assuming the victim participated in the $k_v$-th round, although the adversary cannot identify the exact message sent by the victim, due to the fact that $U_{k_v}$ is a smaller sub-population and is known by the adversary, privacy amplification via subsampling is no longer available.

To demonstrate, let us consider an example from [24]. In this example, subsample-randomize-shuffle is used for training neural networks through federated learning. To jointly train a classifier using MNIST dataset, each user holds a gradient vector $x_i \in \{-0.01, 0.01\}^d$, where $d = 13170$. In each round, $s = 1000$ users are selected from a total $n = 60000$ population. They sanitize their gradient vectors using randomized response on a randomly selected dimension with budget $\epsilon_0 = 2$. Without the in-out attack, a single round gradient aggregation consumes a DP budget (i.e. incurs a privacy loss) of $(0.0064, 10^{-5})$ (using the near-optimal shuffle amplification upper bound in [14, Theorem 3.1] and subsampling amplification bound in [24, Lemma 3]). However, when the adversary possesses the in-out information of a victim participated in a round, the actual privacy budget increases by over 41 times to $(0.27, 10^{-5})$ (using the shuffle amplification lower bound for $s = 1000$ users with randomized response in [43, Theorem 5.1]). Figure 3 shows more privacy loss comparisons with and without in-out attacks. In the worst-case, if a user $v$ is selected for $\Omega(K)$ rounds, the accumulated privacy loss is $\Omega(\sqrt{Ke^{\epsilon_0}/s})$, creating a gap of $s/n$ from the expected privacy loss without in-out attack.

## *E. Privacy Under Message-Length Attacks*

This section studies the impact of message-length leakage on privacy in the shuffle model, analyzing both local privacy and shuffle privacy amplification impacts.

*1) Privacy Amplification Degradation:* Intuitively, if privacy adversaries know the victim's message length $\mathcal{R}_i(x_i)$, they can filter out messages in $\mathcal{S} \circ \mathcal{R}_{[n]}(X)$ with unmatched lengths. Thus, the victim's message can only hide among a subset users with the same length. Specifically, message length information reduces the population size for privacy amplification to:

$$\#\{\mathcal{R}_i(x_i) \mid i \in [n_k]\backslash\{v\} \text{ and } len(\mathcal{R}_i(x_i)) = len(\mathcal{R}_v(x_v))\}. \quad (1)$$

For instance, the seminal work [1] on shuffle privacy amplification considers longitude data $x_i \in \{0, 1\}^d$ aggregation over a period of time $[1 : d]$. To avoid $\Theta(d)$ errors in estimators, a common practice is representing $x_i$ by its hierarchical residue and having users report one hierarchy level. Assuming periods time as $d = 2^H$, the $k$-th value in the $h$-th hierarchy is given by $V_{h,k} = x_i(k \cdot 2^h) - x_i((k-1) \cdot 2^h)$ (assumed $x_i(0) = 0$), where $h \in [0 : H]$, $k \in [1 : d/2^h]$. The study lets users uniformly select one hierarchy level $h \in [0 : H]$ and employ binary randomized response with full $\epsilon_0$ to sanitize ternary vector $V_{h,*} \in \{-1, 0, 1\}^{d/2^h}$, obtaining a message $Y_{h,*} \in \{-1, 1\}^{d/2^h}$. Depending on the hierarchy level $h$ selected, the message length of $Y_{h,*}$ varies significantly, ranging from 1 to $d$. Consequently, the victim can only hide among approximately $n/(\log_2 d + 1)$ users. In Figure 4, we compare remaining privacy protection under message-length attacks with claimed privacy levels in [1]. The actual privacy loss increases by over 200%, and the gap grows with $d$. Similar issues exist in other prevalent aggregation tasks (e.g., for range queries [64] and marginal queries [65]) employing sampling-based query selection or compressed binary randomized response [3]. In these protocols, while $len(\mathcal{R}_i(x_i))$ follows the same distribution $P_{len}$ for any input data $x_i$, message length may vary widely with high entropy. When adversaries possess the message length $len(\mathcal{R}_v(x_v)) = l$ of the victim user $v$, the user can only hide among a much smaller randomized sub-population (of size $|U_{*,l}| \approx 1 + (n-1) \cdot P_{len}[l]$).

*2) Local Privacy Loss:* When local randomizers' outputs are compressed (e.g., via list representation of sparse vector in RAPPOR [49], key-value data randomizer [66]) for transmission efficiency as in [27], the compressed message length might probabilistically reveal information about the message value, thus incur severe local privacy loss (see Appendix A).

*3) Joint In-Out and Message-Length Attack:* The timing of message transmission (in-out information) and the size of transmitted messages (message-length information) are often simultaneously leaked to adversaries. Consequently, the population a victim can hide among is further restricted to those sharing the same message length in the same round, and the privacy amplification effect under joint in-out and message-length attacks can be further diminished.

## V. Theoretical Analyses of Side-Channel Attacks

This section presents formal analyses of the detrimental impact of side-channel information on privacy guarantees in shuffle models. To facilitate theoretical characterization, we introduce some properties of distance measures employed in DP. The data processing inequality asserts that the privacy guarantee cannot be weakened by further analysis of a privatization mechanism's output.

*Definition 8 (Data processing inequality [67])* A distance measure $D : \Delta(\mathbb{T}) \times \Delta(\mathbb{T}) \rightarrow [0, \infty]$ on the probability distribution space satisfies data processing inequality if, for all distributions $P$ and $Q$ in $\Delta(\mathbb{T})$ and for all (possibly randomized) functions $g : \mathbb{T} \rightarrow \mathbb{T}'$,

$$D(g(P)\|g(Q)) \leq D(P\|Q).$$

In the shuffle model, a plethora of sources of randomness exists (e.g., randomness of users' participation choices and randomness of query results from previous rounds). We put forth two instrumental tools for scrutinizing the distance measures under intricate sources of randomness: separability property (refer to Definition 9), and conditioning increasing property (refer to Definition 10).

*Definition 9 (Separability property [67])* A distance measure $D : \Delta(\mathbb{T}) \times \Delta(\mathbb{T}) \rightarrow [0, \infty]$ on the space of probability distributions satisfies separability property if, for all distributions $P$ and $Q$ that are joint densities over $\mathbb{T} = \mathbb{T}_1 \times \mathbb{T}_2$ with the same marginal density with respect to $\mathbb{T}_1$, i.e. $P = P_{T_1} \cdot P_{T_2|T_1}$ and $Q = P_{T_1} \cdot Q_{T_2|T_1}$,

$$D(P\|Q) = \mathbb{E}_{t_1 \sim P_{T_1}}[D(P|t_1\|Q|t_1)],$$

where $P|t_1$ and $Q|t_1$ denote the conditional variables $P_{T_2|t_1}$ and $Q_{T_2|t_1}$, respectively.

*Definition 10 (Conditioning increasing property [67])* A distance measure $D : \Delta(\mathbb{T}) \times \Delta(\mathbb{T}) \rightarrow [0, \infty]$ satisfies condition increase property if, for all distributions $P$ and $Q$ over $\mathbb{T}$ that are generated by $P = \int P_{T|T_1} \mathrm{d}T_1$ and $Q = \int Q_{T|t_1} \mathrm{d}T_1$ with the same variable $T_1$,

$$D(P\|Q) \leq \mathbb{E}_{t_1 \sim P_{T_1}}[D(P|t_1\|Q|t_1)].$$

It is crucial to note that all $f$-divergence measures, including the Hockey-stick and Rényi divergences, adhere to these three properties [67, Proposition 7.1, Theorem 7.2]. When the distribution of the marginal/randomness source variable $T_1$ is the same for $P$ and $Q$, we use $D(P\|Q|t_1)$ to represent $D(P|t_1\|Q|t_1)$ for simplicity.

### A. Privacy Damage of In-Out Attacks

In Theorem 1, we formally demonstrate that the privacy of the shuffle-then-randomize model under in-out attack deteriorates to the local level (no privacy amplification). In Theorem 5 (see Appendix D), we reveal that the privacy of subsample-randomize-shuffle model under in-out attack degrades to the intra-batch level (amplification by $|U_k| = s$ users). Similar results applies to other shuffle model variants.

*Theorem 1 (Shuffle-then-randomize model under in-out attack)* Given two neighboring datasets $X = \{x_0, \ldots, x_v = a, \ldots, x_n\}$, $X' = \{x_0, \ldots, x_v = b, \ldots, x_n\} \in \mathbb{X}^n$, and a protocol $\mathcal{P}_{s\text{-}r}$ in the shuffle-then-randomize model. Let $io_v$ denote the in-out information about user $v$ that $io_v(k) = 1$, then for any distance measure $D$ that satisfies the data processing inequality and separability property:

$$\begin{aligned} &D(\mathcal{P}_{s\text{-}r}(X)\|\mathcal{P}_{s\text{-}r}(X')|io_v(k) = 1) \\ &\geq \min_{z_{[0:k-1]} \in \mathbb{Z}_0 \times \cdots \times \mathbb{Y}_{k-1}} D(\mathcal{R}_k(z_{[0:k-1]}, a)\|\mathcal{R}_k(z_{[0:k-1]}, b)). \end{aligned}$$

### B. Privacy Damage of Message-Length Attacks

Using divide-randomize-shuffle model as an example, we illustrate the destructive power of message-length attacks in Theorem 2. The victim user can only hide among users with the same message length in the same division. For the shuffle-then-randomize and subsample-randomize-shuffle models, the privacy amplification population under message-length attacks is limited to $U_{*,l}$, which represents users with identical message lengths across all rounds. When joint in-out and message-length attacks occur, the privacy amplification population further deteriorates to the same level as divide-randomize-shuffle model: $U_{k,l}$ (see Appendix F).

*Theorem 2: Divide-randomize-shuffle model under message-length attack:* Consider a protocol $\mathcal{P}_{d\text{-}r\text{-}s}$ and non-overlapping complete user divisions $\{U_k\}_{k\in[K]}$ in the divide-randomize-shuffle model, and two neighboring datasets $X = \{x_0, \ldots, x_v = a, \ldots, x_n\}$, $X' = \{x_0, \ldots, x_v = b, \ldots, x_n\} \in \mathbb{X}^n$ that differ at the $v$-th user data. Assuming $v \in U_k$ and $len(\mathcal{R}_{(k)}(z_{(0:k-1)}, a)) \overset{d}{=} len(\mathcal{R}_{(k)}(z_{(0:k-1)}, b))$, let $l$ denote the observed message-length information about user $i$. Define $U_{k,l}$ as the set of users with the same message length (i.e., $U_{k,l} = \{i \mid for\ i \in U_k\ and\ len(\mathcal{R}_{(k)}(x_i)) = l\}$), and let $S = \{X(i)\}_{i\in U_{k,l}}, S' = \{X'(i)\}_{i\in U_{k,l}} \in \mathbb{X}^{|U_{k,l}|}$ denote neighboring datasets w.r.t. $U_{k,l}$ then for any distance measure $D$ that satisfies the data processing inequality and the separability property:

$$\begin{aligned} &D(\mathcal{P}_{d\text{-}r\text{-}s}(X)\|\mathcal{P}_{d\text{-}r\text{-}s}(X')|U_{k,l}, len_v = l) \\ &\geq \min_{z_0} D(\mathcal{P}_{s\text{-}r}(S)\|\mathcal{P}_{s\text{-}r}(S')|z_0), \end{aligned}$$

where $\mathcal{R}_{[|U_{k,l}|]} = \mathcal{R}_{(k)}$ are local randomizers of a shuffle-then-randomize protocol $\mathcal{P}_{s\text{-}r}$ and $z_0$ is the global information.

When the condition that message lengths are distributionally equal given whether $x_v = a$ or $x_v = b$:

$$len(\mathcal{R}_{(k)}(z_{(0:k-1)}, a)) \overset{d}{=} len(\mathcal{R}_{(k)}(z_{(0:k-1)}, b))$$

does not hold in the theorem, there is also local privacy loss due to message length (see Section IV-E).

## VI. Defending Against Side-Channel Attacks

In this section, we present countermeasures for defending against side-channel attacks. These countermeasures give rise to new principles in the shuffle model that offer robustness

TABLE III

Amplified Privacy Level Comparison of MRS and Subsample-Randomize-Shuffle Model ($n$ = 60000, $\epsilon_0$ = 2, and Varying Number of Epochs $E = Ks/n$)

| | $\delta = 10^{-5}$ | | | |
|---|---|---|---|---|
| | $E = 1$ | $E = 10$ | $E = 100$ | $E = 500$ |
| MRS | **0.0357** | **0.126** | **0.444** | **1.070** |
| s=$10^2$ | 0.0393 | 0.143 | 0.566 | 1.584 |
| s=$10^3$ | 0.0367 | 0.129 | 0.458 | 1.132 |
| | $\delta = 10^{-8}$ | | | |
| | $E = 1$ | $E = 10$ | $E = 100$ | $E = 500$ |
| MRS | **0.0571** | **0.190** | **0.635** | **1.485** |
| s=$10^2$ | 0.0629 | 0.211 | 0.769 | 2.037 |
| s=$10^3$ | 0.0593 | 0.194 | 0.651 | 1.551 |

to potential attacks and yield stronger privacy amplification effects. Prior to discussing the specifics, we introduce a novel model: the multinomial-randomize-shuffle (MRS) model, which possesses several advantages in decentralized settings and serves as a foundation for defense against side-channel attacks. For other variants of the shuffle model, the defense techniques are essentially the same.

### *A. Multinomial-Randomize-Shuffle Model*

In this model, each user randomly selects one query $k_i$ from $[K]$ according to a distribution $P_K \in \Delta_K$ and then responds with $\mathcal{R}_{k_i}(z_{(0:k_i-1)}, x_i)$ in the $k_i$-th round. The shuffler uniformly permutes messages received in the $k_i$-th round and releases them to the analyzer. The query selection of each user is private and is not exposed other parties. Consequently, this model has similar privacy amplification effects as the ideal model (the privacy amplification population is $n$, see Theorem 4, proved in Appendix B).

*Definition 11 (Multinomial-randomize-shuffle model)* Let $P_K$ denote a public distribution over $[K]$, and let $k_1, \ldots, k_n$ denote $n$ independent samples following $P_K$. Define $U_k = \{i \mid for\ i \in [n]\ and\ k_i = k\}$ as the $k$-subgroup of users. Let $\mathcal{R}_{(k)} : \mathbb{Z}_0 \times \cdots \times \mathbb{Z}_{k-1} \times \mathbb{X} \to \mathbb{Y}_{(}k)$ denote the randomizer in the $k$-th round, where $\mathbb{Z}_k = \mathbb{Y}^*_{(k)}$ and $\mathbb{Y}_{(k)}$ is the range space of $\mathcal{R}_{(k)}$. The $z_{(0)} \in \mathbb{Z}_0$ is global information. A protocol $\mathcal{P}_{m\text{-}r\text{-}s} : \mathbb{Z}_0 \times \mathbb{X}^n \to \mathbb{Z}_0 \times \cdots \times \mathbb{Z}_K$ in the MRS model proceeds as follows: given a dataset $x_{[1:n]} \in \mathbb{X}^n$, it samples $k_1, \ldots, k_n \sim P_K$ to obtain $\{U_k\}_{k\in[K]}$, then sequentially compute $z_{(i)} = \mathcal{S}(\{\mathcal{R}_{(k)}(z_{(0:k-1)}, x_i)\}_{i\in U_k})$, finally outputs $z_{(0)}, \ldots, z_{(K)}$ and the distribution $P_K$.

The MRS model has some resemblance to the random check-in approach [16]. However, the random check-in involves complex dummy and uniform selection operations on received messages in each round, and only provides privacy amplified by $K$ users, rendering it significantly weaker than the privacy guarantees of MRS. In benign settings, MRS is also preferable to the divide-randomize-shuffle model as it offers greater potential for privacy amplification, yielding asymptotic savings of $(1 - \sqrt{1/K}) \cdot 100\%$ in budgets when all local randomizers comply with LDP. The MRS exhibits similar asymptotic privacy consumption behavior as the subsample-randomize-shuffle model [24], [42]. Assuming there are $K \approx \frac{n}{s}$ queries (i.e., one epoch in federated learning), both models consume $(\tilde{O}(\sqrt{e^{\epsilon_0}/n}), \delta)$-DP. Since the MRS ensures each user participates in exactly one query during one epoch while the subsample-randomize-shuffle model has higher sampling variance, the constant factor in privacy loss of MRS is slightly smaller. Specifically, under the same settings as [24] and [42]: $n$ = 60000, $s$ = 1000, $\epsilon_0$ = 2, and $K = n/s$ = 60, the subsample-randomize-shuffle model consumes privacy of $(0.0367, 10^{-5})$-DP (using the near-optimal shuffle amplification [41] and tight numerical composition with subsampling [68]), while the MRS consumes only $(0.0357, 10^{-5})$-DP (using [41] as well). More results are listed in Table III, the MRS saves 2%-50% budget than subsampling. Moreover, the subsample-randomize-shuffle relies on additional trust in the shuffler for subsampling, whereas the MRS grants control to the user.

### *B. Defend Against Message-Length Attack*

In this subsection, we present a countermeasure to defend against potential message-length attacks. Recall that message-length information possessed by adversaries threatens both local privacy and privacy amplification. To avoid local privacy loss, it is necessary to ensure that the message-length distribution is independent of the true value $x_i$; to avoid degradation in privacy amplification as shown in Equation 1, it is necessary to ensure that the message-length is always the same across all users. The latter requirement is stricter as it demands identical concrete message-lengths for all users with various randomizers and true values.

*1) Padding Messages:* We employ a straightforward approach, message padding (e.g., as [69] for anonymous channels), to defend against message-length attacks. Specifically, we let $len_{max}$ denote the maximum possible message length outputted from the original local randomizers across all users:

$$len_{max} = \max_{i\in[n]} \sup_{x\in\mathbb{X}} len(\mathcal{R}_i(x)).$$

Then, for all messages released from each user, the message payload is padded to $len_{max}$ bits.

After implementing the padding, the message length becomes global information shared among all parties. Consequently, possessing the victim's message length information no longer provides an advantage for privacy attacks.

### *C. Defend Against In-Out Attacks*

In this subsection, we present several strategies for defending against in-out attacks. For non-adaptive queries in the single-message shuffle model, there is a simple and efficient strategy: *parallelizing queries*; for adaptive queries in the single-message shuffle model, users could sacrifice communication overheads to regain privacy amplification from in-out attacks by sending *dummy messages* (a common approach in the anonymous channel literature [40]).

*1) Non-Adaptive Queries in the Single-Message Shuffle Model:* Many data analysis tasks involve multiple non-adaptive estimation queries. In the local model of DP, a common practice for achieving better utility (compared to dividing the privacy budget $\epsilon_0$) is to separate the entire user population into multiple non-overlapping subsets and assign each subset to accomplish one query with the full budget $\epsilon_0$. For example, this approach is used in heavy hitter estimation [70], marginal queries [65], and machine learning [71]. This complies with the parallel composition theorem of differential privacy in the central model [2]. In the shuffle model, an (almost) equivalent approach is to have each user randomly choose one query among all $K$ queries with a fixed probability distribution $P_K \in \Delta_K$, and contribute to the chosen query with the full budget [1], [43]. We illustrate this approach in Algorithm 3. Since all mechanisms $M_k$ ($k \in [K]$) are $\epsilon_0$-LDP, the overall algorithm is $\epsilon_0$-LDP.

**Algorithm 3** Parallel Local Randomizer [1], [43]

```
Params : A distribution $P_K : [K] \mapsto [0, 1]$, base randomizers
    $\{\mathcal{M}_k : \mathbb{X} \mapsto \mathbb{Y}_k\}_{k\in[K]}$ each satisfies $\epsilon_0$-LDP.
Input: An input $x \in \mathbb{X}$.
Output: An output that satisfies $\epsilon_0$-LDP.
sample $k \sim P_K$
$y \leftarrow \mathcal{M}_k(x)$
return $y$
```

Given that every user follows the same distribution $P_K$, this implies all users are adopting an identical randomization algorithm, which ensures that privacy amplification via shuffling still holds. Denote Algorithm 3 as $\mathcal{R}$, one straightforward conclusion is that:

$$D(\mathcal{S} \circ \mathcal{R}(X) \parallel \mathcal{S} \circ \mathcal{R}(X')) \leq D(\mathcal{P}_{s\text{-}r}(X) \| \mathcal{P}_{s\text{-}r}(X')),$$

where the shuffle-then-randomize model employs $\mathcal{R}$ as the local randomizers.

*2) Parallel as a Cost-Free Defense to in-Out Attacks:* We emphasize that parallel local randomizers are naturally immune to in-out attacks in the shuffle model. As all $K$ queries are performed together in one single round (with the parallel randomizer in Algorithm 3), the in-out information $io_v = \{1\}$ is trivial, thus maintaining the population size for shuffle privacy amplification at $n$. We note that parallelizing queries does not require the shuffler/server to process all messages from users within a specific short period of time but rather serves to obfuscate which specific query the victim participated in. The message sending and receiving time window can span a wide range, and the shuffler/server could process these messages in the virtual timestamp that corresponds to this time window.

**Algorithm 4** Rectified Parallel Local Randomizer

```
Params : A distribution $P_K : [K] \mapsto [0, 1]$, base randomizers
    $\{\mathcal{M}_k : \mathbb{X} \mapsto \mathbb{Y}_k\}_{k\in[K]}$ each satisfies $\epsilon_0$-LDP, the maximum
    possible length $len_{max}$.
Input: An input $x \in \mathbb{X}$.
Output: An output that satisfies $\epsilon_0$-LDP.
sample $k \sim P_K$
$y \leftarrow \mathcal{M}_k(x)$
add padding bits toy to form a $len_{max}$-length bit vector $y'$
    encrypts $y'$ with the
public key of the analyzer and get$Enc_a(y')$
return $Enc_a(y')$
```

*3) Defend Joint In-Out & Message-Length Attacks:* We note that message-length information must be protected in the parallel local randomizer, as each base randomizer $\mathcal{M}_k$ often has a different output space $\mathbb{Y}_k$, and the message length leaks information about each query the victim participated in. We present an implementation that resists joint in-out and message-length attacks in Algorithm 4, where messages are padded. Now for the privacy properties of Algorithm 4, since it can be considered as post-processing upon Algorithm 3 and both in-out & message-length information are trivial, it enjoys the same local and shuffle privacy guarantees as the shuffle-then-randomize model (with identical local randomizers).

**Algorithm 5** Rectified MRS Model

```
Params: A probability distribution $P_K : [K] \mapsto [0, 1]$,
        adaptive local randomizers $\{\mathcal{R}_{(k)}\}_{k\in[K]}$, the
        maximum possible length $len_{max}$, global
        information $z_{(0)}$.
Input: Inputs $x_1, \ldots, x_n \in \mathcal{X}$ from $n$ users.
Output: The querying results of $K$ adaptive queries.
1  ▷ Sample participation choices on the user side
2  for users $i \in [n]$ do
3      sample $k_i \sim P_K$
4  ▷ Run randomization & shuffling
5  for $k \in [K]$ do
6      for users $i \in [n]$ do
7          ▷ Randomize, pad, and encrypt on the user side
8          if $k = k_i$ then
9              $y_{i,k} \leftarrow \mathcal{R}_{(k)}(z_{(0:k-1)}, x_i)$
10         else
11             let $y_{i,k}$ be an empty message
12         pad $y_{i,k}$ to form a $len_{max}$-length bit vector
           $y'_{i,k}$
13         encrypts $y'_{i,k}$ with the analyzer's public key
           to get $Enc_a(y'_{i,k})$
14     ▷ Uniform-randomly permute on the shuffler side
15     $E_k = \mathcal{S}(Enc_a(y'_{1,k}), \ldots, Enc_a(y'_{n,k}))$
16     ▷ Decrypt and analyze on the server side
17     $Y'_k = \{Dec_a(ciphertext) \mid cihpertext \in E_k\}$
18     $Y_k = \{y \mid y \in Y'_k \; and \; y \; is \; not \; empty\}$
19     $z_{(k)} = \mathcal{A}_k(Y_k)$
20 return $z_{(0)}, \ldots, z_{(K)}$
```

*4) Adaptive Queries in Single-Message Shuffle Model:* For adaptive queries, parallelizing queries is not applicable since the $k$-th query relies on previous querying results $z_{(0)}, \ldots, z_{(k-1)}$. We propose contributing dummy messages at every round based on the MRS model (see Algorithm 5), at the cost of $(K-1)$-times more communication overhead. We then show how to trade-off communication and amplification.

*5) Sending Dummy Messages:* To obscure the information about which round a user participated in and prevent adversaries from obtaining meaningful in-out information, we let every user contribute message(s) at all $K$ rounds. In the true participating round $k_i$ (sampled as in the normal MRS model), user $i$ contributes a true message $\mathcal{R}(k_i)(z_{(0:k-1)}, x_i)$ (line 9); in the other $K-1$ rounds, user $i$ contributes a dummy/empty message (line 11). To further prevent adversaries from launching message-length attacks, all messages are padded (line 12) and encrypted before transmission.

Since every user participates in all rounds, the in-out information $io_v = [1, \ldots, 1]$ becomes trivial in the new model presented in Algorithm 5; since all messages have the same length, there is no degradation due to side-channel message-length information. Moreover, the observed shuffled messages $E_1, \ldots, E_K$ can be viewed as adding padded empty messages at every round to the outputs of $\mathcal{P}_{m\text{-}r\text{-}s}$ according to the total number of users $n$ (a public piece of information). Therefore, the new model resists in-out & message-length attacks and has the same privacy amplification effects as the normal MRS model in Theorem 4.

A practical issue in the new model is that the total messages grows to $K$ times when compared to the original model. This might pose significant communication burdens on users with scarce resources (e.g., in mobile devices). To balance communication costs and privacy amplification, we propose a flexible, generalized participation model extending the multinomial approach.

*6) Bin Participation Paradigm:* The model first defines a series of bar points $\{b_0, b_1, \ldots, b_M\}$ in $[K+1]$. Specifically, $b_0 \equiv 1$, $b_M \equiv K+1$, and $b_m < b_{m+1}$ for $m \in [M-1]$. The model then defines a set $Q = \{Q_{(1)}, \ldots, Q_{(M)}\}$ of non-overlapping, consecutive, and complete subsets (bins) of $[K]$ where $Q_{(m)} = [b_{m-1}, b_m - 1]$ for $m \in [M]$. It is obvious that $Q_{(m)} \neq \emptyset$, $Q_{(m)} \cap Q_{(m')} = \emptyset$ holds for all $m, m' \in [M]$ when $m \neq m'$, and $Q_{(1)} \cup Q_{(2)} \cup \cdots \cup Q_{(M)} = [K]$. For each $Q_{(m)} \in Q$, it is associated with a probability distribution $P_{(m)} : Q_{(m)} \mapsto [0,1] \in \Delta_{|Q_{(m)}|}$. In the binned participating model, each user $i \in [n]$ select $m_i \in [M]$ with an *arbitrary rule*. For example, if a user becomes available online after round $k$, the user might select a bin $Q_{(m_i)}$ such that $Q_{(m_i)} \subseteq [k+1 : K]$; the user might also arbitrary-randomly select a bin $Q_{(m_i)}$ from $Q$. After choosing the bin $Q_{(m_i)}$, the user $i$ samples a true participation round $k_i$ from $Q_{(m_i)}$ according to $P_{(m_i)}$. The corresponding shuffle model with the binned participation paradigm is termed the bin-randomize-shuffle model. Specifically, *when $Q = [n]$, the bin-randomize-shuffle model is equivalent to the MRS model.*

We let $U_{(m)} \subseteq [n]$ denote the users that selected bin $Q_{(m)}$ in the bin-randomize-shuffle model, $U_k \subseteq [n]$ denote the users that selected query $k$, and let $l_{(k)}$ denote the bin where $k$ belongs to (i.e., $k \in Q_{(l_{(k)})}$). Then, by combining the defensive techniques in the MRS model, we show that user $i \in U_{(m)}$ can hide among $U_{(m)}$, even under in-out and message-length attacks. We present the overall procedure in Algorithm 6, and formally state the privacy amplification guarantee in Theorem 3 (see Appendix G for proof). Since $U_{(m)}$ lies between $U_k$ and $[n]$, the privacy amplification and communication costs can be flexibly traded off.

**Algorithm 6** Rectified Bin-Randomize-Shuffle Model

```
Params: Participation bins $Q_{(1:M)}$, probability
        distributions $P_{(m)} : Q_{(m)} \mapsto [0,1]$ for $m \in [M]$,
        adaptive local randomizers $\{\mathcal{R}_{(k)}\}_{k \in [K]}$, the
        maximum possible length $len_{max}$, global
        information $z_{(0)}$.
Input: Inputs $x_1, \ldots, x_n \in \mathcal{X}$ from $n$ users.
Output: The querying results of $K$ adaptive queries.
▷ Compute participation choices on the user side
for users $i \in [n]$ do
    choose $m_i \in [M]$ with an arbitrary (personalized) rule
    $k_i \sim P_{(m_i)}$
▷ Run randomization & shuffling
for $k \in [K]$ do
    for users $i \in [n]$ do
        ▷ Randomize, pad, and encrypt on the user side
        if $k \in Q_{(m_i)}$ then
            if $k = k_i$ then
                $y_{i,k} \leftarrow \mathcal{R}_{(k)}(z_{(0:k-1)}, x_i)$
            else
                let $y_{i,k}$ be an empty message
            pad $y_{i,k}$ to form a $len_{max}$-length bit vector $y'_{i,k}$
            encrypts $y'_{i,k}$ with the analyzer's public key to get $Enc_a(y'_{i,k})$
    ▷ Uniform-randomly permute on the shuffler side
    $E_k = \mathcal{S}(\{Enc_a(y'_{i,k})\}_{i \in U_{l_{(k)}}})$
    ▷ Decrypt and analyze on the server side
    $Y'_k = \{Dec_a(ciphertext) \mid cihpertext \in E_k\}$
    $Y_k = \{y \mid y \in Y'_k \text{ and } y \text{ is not empty}\}$
    $z_{(k)} = \mathcal{A}_k(Y_k)$
return $z_{(0)}, \ldots, z_{(K)}$
```

*Theorem 3 (Rectified bin-randomize-shuffle model)* Given a protocol $\mathcal{P}_{b\text{-}r\text{-}s}$, non-overlapping complete bins $Q = \{Q_{(1)}, \ldots, Q_{(m)}\}$, and non-overlapping complete user divisions $\{U_{(m)}\}_{m \in [M]}$ in the binned-randomize-shuffle model (as in Algorithm 6), and two neighboring datasets $X = \{x_0, \ldots, x_v = a, \ldots, x_n\}$, $X' = \{x_0, \ldots, x_v = b, \ldots, x_n\} \in \mathbb{X}^n$. Let $l$ denote the message-length information about user $v$ that $v \in U_k \subseteq U_{(m)}$, and let $S = \{X(i')\}_{i' \in U_{(m)}}, S' = \{X'(i')\}_{i' \in U_{(m)}} \in \mathbb{X}^{|U_{(m)}|}$ denote neighboring datasets w.r.t. $U_{(m)}$ then for any distance measure $D$ that satisfies the data processing inequality and separability property:

$$D(\mathcal{P}_{b\text{-}r\text{-}s}(X) \| \mathcal{P}_{b\text{-}r\text{-}s}(X') | U_{(m)}, len_v = l) \le \max_{z_0} D(\mathcal{P}_{m\text{-}r\text{-}s}(S) \| \mathcal{P}_{m\text{-}r\text{-}s}(S') | z_0),$$

where $\{\mathcal{R}_{(k')}\}_{k' \in Q_m}$ are the local randomizers of a multinomial-randomize-shuffle protocol $\mathcal{P}_{m\text{-}r\text{-}s}$ that has $|Q_m|$ rounds, $z_0$ is the global information in $\mathcal{P}_{m\text{-}r\text{-}s}$ and $P_{(m)}$ is the query selection distribution of the $m$-th bin.

### *D. Defend Against Message-Cardinality Attacks*

Section IV-C highlights that message-cardinality attacks may result in significant privacy loss for some SOTA protocols [17], [19], [46], [47]. In this section, we demonstrate the difficulty and feasibility in securing the $\Delta$-summation protocol [19] against message-cardinality attacks.

Recall that in the $\Delta$-summation protocol (see Algorithm 1 and 2) with $\Delta = 1$, if a user has $x_i = 1$, they send a 1 and a random number of $-1, 1$ messages to the shuffler. Conversely, if a user has $x_i = 0$, they send only a random number of $-1, 1$ messages following the same distribution as when $x_i = 1$. A straightforward remedy would involve sending an extra 0 (or any other stub/dummy message) to the shuffler when $x_i = 0$ to ensure identical message cardinality distributions for both $x_i = 1$ and $x_i = 0$. However, while this addresses the message-cardinality issue, the number of 0s or stubs observed by the analyzer (i.e., potential adversaries) in shuffled messages directly reveals the number of users with 0, leading to an infinite differential privacy loss.

One feasible approach is transforming the $\Delta$-summation problem into a $(\Delta + 1)$-summation problem while utilizing the original protocol as a base. In this approach, given user data $x_i \in [\Delta]$, we first transform it to $x_i + 1 \in [\Delta + 1]$. These transformed values are then fed into a $(\Delta + 1)$-summation protocol from [19]. After retrieving the noisy summation $s$ from the protocol, we obtain the final unbiased estimator of $\sum_{i\in[n]} x_i$ by subtracting $n$ from $s$. By incorporating the transformation step where we only input $x_i + 1$, which is always greater than 0, the local randomizer in Algorithm 1 consistently sends $x_i + 1$, effectively eliminating the message-cardinality issue. Since the user population size $n$ is a publicly known parameter and each user data point increases by 1, we can still recover the unbiased noisy summation. The only additional cost incurred is the increased message complexity associated with a $(\Delta + 1)$-summation protocol when compared to a $\Delta$-summation protocol.

It is important to note that this defense approach cannot be generalized to other vulnerable protocols in [17], [46], and [47] that only take binary inputs.

## VII. Discussions

In this section, we summary new privacy requirements and novel principles for the shuffle model, and discuss their implications to more broader DP-Cryptography systems.

### A. Curious Shuffle Model

Keep in mind that privacy attackers can easily carry out side-channel attacks on the shuffle model. A typical example of such attackers is an honest-but-*curious* shuffler. If we let $\texttt{SideInfo}(i)$ represent the information that privacy attackers have about a user $i \in [n]$, we can then define a side-channel-resistant version of DP in the shuffle model like this:

*Definition 12 (DP in the curious shuffle model)* A protocol $\mathcal{P} = (\{\mathcal{R}_i\}_{i\in[n]}, \mathcal{A})$ satisfies $(\epsilon, \delta)$-differential privacy in the curious shuffle model iff for all neighboring datasets $X$ and $X' \in \mathbb{X}^n$, the $(\mathcal{S} \circ \mathcal{R}_{[n]}(X), \texttt{SideInfo}(i)_{i\in[n]})$ and $(\mathcal{S} \circ \mathcal{R}_{[n]}(X'), \texttt{SideInfo}(i)_{i\in[n]})$ are $(\epsilon, \delta)$-indistinguishable.

Usually, $\texttt{SideInfo}(i)$ includes $io(i), num(\mathcal{R}_i)$ and $len(\mathcal{R}_i)$ as defined in Section IV-B. In this case, our rectified proposals in Section VI maintain privacy amplification in the curious shuffle model.

### B. New Principles in the Shuffle Model

To maximize privacy amplification effects and minimize side-channel risks, we summarize new principles for the (curious) shuffle model:

*1) (a) Pad Every Message:* As demonstrated in Section IV-E, message-length information can significantly compromise privacy. A simple but effective solution is to pad each message to at least the maximum possible message length [69], $len_{max}$. Consequently, the message length becomes a public information in the system, thus avoid privacy degradation even when adversaries possess message length information.

*2) (b) Parallelize Queries:* When handling multiple queries with no sequential dependence, the best practice in the shuffle model is to pack them into one parallel query (see Section VI-C1). This allows users to benefit from privacy amplification over the full population without sending dummy messages.

*3) (c) Contribute Dummy Messages:* As for adaptive queries, existing shuffle models fail to fully amplify privacy across rounds when there are side-channel attacks. An effective mitigation approach is to have each user contribute extra dummy messages [40] to the shuffler at every round (as shown in Algorithm 5), so that users can hide among the overall population. Alternatively, users may adopt the bin-randomize-shuffle model (see Algorithm 6) to flexibly control the trade-off between communication overheads, privacy amplification effects, and utility.

Combining these principles enables users to achieve optimized privacy-utility-communication trade-offs.

### C. New Challenges in DP-Crypto Systems

Stemming from this study, while countermeasures from the cryptography literature (e.g., padding bits and sending dummy messages) offer valuable insights, they are not universally applicable to defending against side-channel attacks in the shuffle DP model (see Section VI-D) and may be inefficient (see Section VI-C), due to differences in functionalities and privacy goals. Our research pivots to a crucial frontier: the synthesis of information-theoretic privacy tools and cryptographic methods (e.g., anonymous channels, oblivious shuffling, order preserving encryption, and approximate homomorphic encryption). This merger seeks not only to balance privacy and utility/efficiency but also to illuminate and address novel attack vectors. These attacks, within the DP context, beckon comprehensive analyses and the genesis of innovative defenses.

## VIII. Conclusion

This study presents a novel identification of communication side-channel attacks associated with the shuffle model of differential privacy. We classify these attacks into three categories: in-out, message-length, and message-cardinality attacks. Additionally, we empirically and theoretically investigate the resulting degradation of privacy due to these attacks. Our findings reveal that these attacks cause a significant or even infinity increase in privacy loss, potentially nullifying the benefits of privacy amplification through shuffling. To counteract these vulnerabilities, we propose two new variants of the shuffle model: the MRS model and the bin-randomize-shuffle model. We introduce new principles within these models, such as parallelizing queries, padding messages, and sending dummy messages. As a result, the privacy amplification effects are redeemed with minimal additional costs.

Furthermore, these new models are applicable not only for defending against attacks but also in benign environments, resulting in significantly stronger privacy amplification effects compared to existing models.

## References

[1] Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, K. Talwar, and A. Thakurta, "Amplification by shuffling: From local to central differential privacy via anonymity," in *Proc. SODA*, Jan. 2019, pp. 2468–2479.

[2] C. Dwork, "Differential privacy," in *Proc. ICALP*, Jan. 2006, pp. 1–12.

[3] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Proc. 51st Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Oct. 2013, pp. 1592–1592.

[4] S. Wang et al., "Local differential private data aggregation for discrete distribution estimation," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 9, pp. 2046–2059, Sep. 2019.

[5] S. Wang, L. Huang, Y. Nie, P. Wang, H. Xu, and W. Yang, "PrivSet: Set-valued data analyses with locale differential privacy," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2018, pp. 1088–1096.

[6] S. Wang, Y. Qian, J. Du, W. Yang, L. Huang, and H. Xu, "Set-valued data publication with local privacy: Tight error bounds and efficient mechanisms," *Proc. VLDB Endowment*, vol. 13, no. 8, pp. 1234–1247, Apr. 2020.

[7] S. Wang, Y. Peng, K. Chen, and W. Yang, "Optimal locally private data stream analytics," in *Proc. IEEE Conf. Comput. Commun.*, May 2024, pp. 31–40.

[8] S. Wang, X. Luo, Y. Qian, J. Du, W. Lin, and W. Yang, "Analyzing preference data with local privacy: Optimal utility and enhanced robustness," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 8, pp. 7753–7767, 2023.

[9] S. Wang et al., "Locally private set-valued data analyses: Distribution and heavy hitters estimation," *IEEE Trans. Mobile Comput.*, vol. 23, no. 8, pp. 8050–8065, Aug. 2024.

[10] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proc. USENIX Secur. Symp.*, vol. 4, 2004, pp. 303–320.

[11] V. Costan and S. Devadas, "Intel SGX explained," *Cryptol. ePrint Arch.*, vol. 2016, p. 86, Jan. 2016.

[12] A. Bittau et al., "Prochlo: Strong privacy for analytics in the crowd," in *Proc. 26th Symp. Operating Syst. Princ.*, Oct. 2017, pp. 441–459.

[13] I. Abraham, B. Pinkas, and A. Yanai, "Blinder—Scalable, robust anonymous committed broadcast," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2020, pp. 1233–1252.

[14] V. Feldman, A. McMillan, and K. Talwar, "Stronger privacy amplification by shuffling for Rényi and approximate differential privacy," in *Proc. SODA*, Jan. 2023, pp. 4966–4981.

[15] B. Balle, J. Bell, A. Gascón, and K. Nissim, "The privacy blanket of the shuffle model," in *Proc. CRYPTO*, Jan. 2019, pp. 638–667.

[16] B. Balle, P. Kairouz, H. B. McMahan, O. Thakkar, and A. Thakurta, "Privacy amplification via random check-ins," in *Proc. NeurIPS*, Jan. 2020, pp. 4623–4634.

[17] B. Ghazi, R. Kumar, P. Manurangsi, and R. Pagh, "Private counting from anonymous messages: Near-optimal accuracy with vanishing communication overhead," in *Proc. ICML*, Jan. 2021, pp. 3505–3514.

[18] V. Balcer, A. Cheu, M. Joseph, and J. Mao, "Connecting robust shuffle privacy and pan-privacy," in *Proc. SODA*, Jan. 2021, pp. 2384–2403.

[19] B. Ghazi, R. Kumar, P. Manurangsi, R. Pagh, and A. Sinha, "Differentially private aggregation in the shuffle model: Almost central accuracy in almost a single message," in *Proc. ICML*, Jul. 2021, pp. 3692–3701.

[20] S. Wang, J. Li, Y. Qian, J. Du, W. Lin, and W. Yang, "Hiding numerical vectors in local private and shuffled messages," in *Proc. 30th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 3706–3712.

[21] A. Cheu and M. Zhilyaev, "Differentially private histograms in the shuffle model from fake users," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2022, pp. 440–457.

[22] S. Wang et al., "Shuffle differential private data aggregation for random population," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 5, pp. 1667–1681, May 2023.

[23] S. Wang et al., "Differentially private numerical vector analyses in the local and shuffle model," *IEEE Trans. Dependable Secure Comput.*, vol. 22, no. 1, pp. 1–15, Jan. 2025.

[24] A. M. Girgis, D. Data, S. Diggavi, P. Kairouz, and A. T. Suresh, "Shuffled model of differential privacy in federated learning," in *Proc. AISTATS*, Mar. 2021, pp. 2521–2529.

[25] J. M. Tenenbaum, H. Kaplan, Y. Mansour, and U. Stemmer, "Differentially private multi-armed bandits in the shuffle model," in *Proc. NeurIPS*, Jan. 2021, pp. 24956–24967.

[26] A. Lowy and M. Razaviyayn, "Private federated learning without a trusted server: Optimal algorithms for convex losses," 2021, *arXiv:2106.09779*.

[27] A. Cheu, M. Joseph, J. Mao, and B. Peng, "Shuffle private stochastic convex optimization," in *Proc. ICLR*, Jan. 2021, pp. 1–28.

[28] S. R. Chowdhury and X. Zhou, "Shuffle private linear contextual bandits," in *Proc. ICML*, Jan. 2022, pp. 3984–4009.

[29] E. Garcelon, K. Chaudhuri, V. Perchet, and M. Pirotta, "Privacy amplification via shuffling for linear contextual bandits," in *Proc. ALT*, Jan. 2021, pp. 381–407.

[30] F. Li, X. Zhou, and B. Ji, "Differentially private linear bandits with partial distributed feedback," in *Proc. 20th Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, Sep. 2022, pp. 41–48.

[31] Apple Google.(2021). *Exposure Notification Privacy-Preserving Analytics White Paper*. [Online]. Available: https://covid19-static.cdn-apple.com/applications/covid19/current/static/contact-tracing/pdf/ENPA_White_Paper.pdf

[32] Y.-S. Shiu, S. Y. Chang, H.-C. Wu, S. C.-H. Huang, and H.-H. Chen, "Physical layer security in wireless networks: A tutorial," *IEEE Wireless Commun.*, vol. 18, no. 2, pp. 66–74, Apr. 2011.

[33] Y. Wan, K. Xu, F. Wang, and G. Xue, "Characterizing and mining traffic patterns of IoT devices in edge networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 1, pp. 89–101, Jan. 2021.

[34] S. Feghhi and D. J. Leith, "A Web traffic analysis attack using only timing information," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 8, pp. 1747–1759, Aug. 2016.

[35] J. Deng, R. Han, and S. Mishra, "Countermeasures against traffic analysis attacks in wireless sensor networks," in *Proc. 1st Int. Conf. Secur. Privacy Emerg. Areas Commun. Netw. (SECURECOMM)*, Sep. 2005, pp. 113–126.

[36] F. Zhang, W. He, and X. Liu, "Defending against traffic analysis in wireless networks through traffic reshaping," in *Proc. 31st Int. Conf. Distrib. Comput. Syst.*, Jun. 2011, pp. 593–602.

[37] S. J. Murdoch and G. Danezis, "Low-cost traffic analysis of tor," in *Proc. IEEE Symp. Secur. Privacy (S&P)*, May 2005, pp. 183–195.

[38] L. Overlier and P. Syverson, "Locating hidden servers," in *Proc. IEEE Symp. Secur. Privacy (S&P)*, May 2006, pp. 15–pp.

[39] G. Danezis and A. Serjantov, "Statistical disclosure or intersection attacks on anonymity systems," in *Proc. Inf. Hiding, 6th Int. Workshop*, Jan. 2004, pp. 293–308.

[40] O. Berthold and H. Langos, "Dummy traffic against long term intersection attacks," in *Proc. Privacy Enhancing Technol., 2nd Int. Workshop*, Jan. 2003, pp. 110–128.

[41] V. Feldman, A. McMillan, and K. Talwar, "Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling," in *Proc. IEEE 62nd Annu. Symp. Found. Comput. Sci. (FOCS)*, Feb. 2022, pp. 954–964.

[42] A. M. Girgis, D. Data, and S. Diggavi, "Renyi differential privacy of the subsampled shuffle model in distributed learning," in *Proc. NeurIPS*, Jan. 2021, pp. 29181–29192.

[43] S. Wang et al., "Privacy amplification via shuffling: Unified, simplified, and tightened," *Proc. VLDB Endowment*, vol. 17, no. 8, pp. 1870–1883, Apr. 2024.

[44] Z. Weinberg et al., "StegoTorus: A camouflage proxy for the tor anonymity system," in *Proc. ACM Conf. Comput. Commun. Secur.*, Oct. 2012, pp. 109–120.

[45] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Commun. ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[46] B. Ghazi, N. Golowich, R. Kumar, P. Manurangsi, R. Pagh, and A. Velingker, "Pure differentially private summation from anonymous messages," in *Proc. 1st Conf. Inf.-Theoretic Cryptography (ITC)*, Jan. 2020, pp. 15:1–15:23.

[47] V. Balcer and A. Cheu, "Separating local & shuffled differential privacy via histograms," in *Proc. 1st Conf. Inf.-Theoretic Cryptography*, Jan. 2020, pp. 1–14.

[48] J. M. Abowd, "The U.S. census bureau adopts differential privacy," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, p. 2867.

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

[50] I. Mironov, "On significance of the least significant bits for differential privacy," in *Proc. ACM Conf. Comput. Commun. Secur.*, Oct. 2012, pp. 650–661.

[51] J. Jin, E. McMurtry, B. I. P. Rubinstein, and O. Ohrimenko, "Are we there yet? Timing and floating-point attacks on differential privacy systems," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2022, pp. 473–488.

[52] A. Haeberlen, B. C. Pierce, and A. Narayan, "Differential privacy under fire," in *Proc. 20th USENIX Secur. Symp. (USENIX Secur.)*, Aug. 2011, pp. 1–33.

[53] A. Cheu, A. Smith, and J. Ullman, "Manipulation attacks in local differential privacy," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2021, pp. 883–900.

[54] T. Wang et al., "Improving utility and security of the shuffler-based differential privacy," *Proc. VLDB Endowment*, vol. 13, no. 13, pp. 3545–3558, Sep. 2020.

[55] A. Cheu, A. Smith, J. Ullman, D. Zeber, and M. Zhilyaev, "Distributed differential privacy via shuffling," in *Proc. EUROCRYPT*, Jan. 2019, pp. 375–403.

[56] B. Balle, G. Barthe, and M. Gaboardi, "Privacy amplification by subsampling: Tight analyses via couplings and divergences," in *Proc. NeurIPS*, Jan. 2018, pp. 6280–6290.

[57] Q. Wang, A. Yahyavi, B. Kemme, and W. He, "I know what you did on your smartphone: Inferring app usage over encrypted data traffic," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Sep. 2015, pp. 433–441.

[58] J. S. Atkinson, J. E. Mitchell, M. Rio, and G. Matich, "Your WiFi is leaking: What do your mobile apps gossip about you?," *Future Gener. Comput. Syst.*, vol. 80, pp. 546–557, Mar. 2018.

[59] E. Papadogiannaki and S. Ioannidis, "A survey on encrypted network traffic analysis applications, techniques, and countermeasures," *ACM Comput. Surveys*, vol. 54, no. 6, pp. 1–35, Jul. 2022.

[60] D. Naboulsi, M. Fiore, S. Ribot, and R. Stanica, "Large-scale mobile traffic analysis: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 124–161, 1st Quart., 2016.

[61] J. Ren, A. Rao, M. Lindorfer, A. Legout, and D. Choffnes, "ReCon: Revealing and controlling PII leaks in mobile network traffic," in *Proc. 14th Annu. Int. Conf. Mobile Syst., Appl., Services*, Jun. 2016, pp. 361–374.

[62] M. Conti, Q. Q. Li, A. Maragno, and R. Spolaor, "The dark side(-channel) of mobile devices: A survey on network traffic analysis," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2658–2713, 4th Quart., 2018.

[63] L. Chen, B. Ghazi, R. Kumar, and P. Manurangsi, "On distributed differential privacy and counting distinct elements," in *Proc. 12th Innov. Theor. Comput. Sci. Conf. (ITCS)*, Jan. 2020, pp. 56:1–56:18.

[64] G. Cormode, T. Kulkarni, and D. Srivastava, "Answering range queries under local differential privacy," *Proc. VLDB Endowment*, vol. 12, no. 10, pp. 1126–1138, Jun. 2019.

[65] G. Cormode, T. Kulkarni, and D. Srivastava, "Marginal release under local differential privacy," in *Proc. Int. Conf. Manage. Data*, May 2018, pp. 131–146.

[66] X. Gu, M. Li, Y. Cheng, L. Xiong, and Y. Cao, "PCKV: Locally differentially private correlated key-value data collection with optimized utility," in *Proc. USENIX Secur.*, Jan. 2019, pp. 967–984.

[67] Y. Polyanskiy and Y. Wu, *Information Theory: From Coding to Learning*. Cambridge, U.K.: Cambridge Univ. Press, 2022. [Online]. Available: https://people.lids.mit.edu/yp/homepage/data/itbook-export.pdf

[68] A. Koskela, M. A. Heikkilä, and A. Honkela, "Numerical accounting in the shuffle model of differential privacy," *Trans. Mach. Learn. Res.*, 2023.

[69] S. Yu, G. Zhao, W. Dou, and S. James, "Predicted packet padding for anonymous Web browsing against traffic analysis attacks," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 4, pp. 1381–1393, Aug. 2012.

[70] R. Bassily, K. Nissim, U. Stemmer, and A. Thakurta, "Practical locally private heavy hitters," in *Proc. NeurIPS*, vol. 30, Jul. 2017, pp. 2288–2296.

[71] S. Fletcher and M. Z. Islam, "Decision tree classification with differential privacy: A survey," *ACM Comput. Surv.*, vol. 52, no. 4, pp. 1–33, Jul. 2020.

**Shaowei Wang** received the Ph.D. degree in computer science from the University of Science and Technology of China (USTC) in 2019. He is an Associate Professor with the Institute of Artificial Intelligence, Guangzhou University. He has published more than 40 papers on top-tier conferences and journals, such as NeurIPS, VLDB, INFOCOM, ICDE, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING. His research interests are data privacy, federated learning, and recommendation systems.

**Jin Li** is currently pursuing the master's degree with Guangzhou University. His research interests are data privacy and federated learning.

**Changyu Dong** received the Ph.D. degree from Imperial College London. He is currently a Professor at the Institute of Artificial Intelligence, Guangzhou University. He has authored more than 50 publications in international journals and conferences. His research interests include applied cryptography, data privacy, AI security, and blockchain. His recent work focuses mostly on designing practical secure computation protocols. The application domains include secure cloud computing and privacy-preserving data mining.

**Jin Li** received the Ph.D. degree in information security from Sun Yat-sen University, Guangzhou, China, in 2007. He is currently a Professor and the Vice Dean of the School of Computer Science, Guangzhou University. He has published more than 100 papers in top-tier international conferences and journals, including IEEE INFOCOM, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and ESORICS. His work has been cited more than 10000 times at Google Scholar and the H-Index is 34. His research interests include design of secure protocols in cloud computing and cryptographic protocols. He received the NSFC Outstanding Youth Foundation in 2017.

**Zhili Zhou** (Senior Member, IEEE) received the Ph.D. degree in computer application from the School of Information Science and Engineering, Hunan University, in 2014. He is currently a Professor with the Institute of Artificial Intelligence, Guangzhou University. His current research interests include multimedia security, artificial intelligence security, and information hiding. He received the ACM Rising Star Award and Guangdong Natural Science Funds for Distinguished Young Scholar. He was as a "World's Top 2% Scientists" by Stanford University and Elsevier from 2019 to 2022.

**Di Wang** (Member, IEEE) received the Ph.D. degree in computer science from The State University of New York (SUNY), Buffalo. He is currently an Assistant Professor of computer science and a Faculty Member of statistics with the Division of Computer, Electrical and Mathematical Sciences and Engineering (CEMSE), King Abdullah University of Science and Technology (KAUST). His research interests include trustworthy machine learning, machine learning theory, and AI for science.

**Zikai Wen** received the Ph.D. degree in computer science from Cornell University in 2021. He is an Assistant Professor of Information Technology at the School of Engineering & Technology, University of Washington Tacoma. His research in human-computer interaction (HCI) focuses on usable privacy and security, emphasizing the importance of user autonomy, and informed decision-making. His work appears in leading conferences and journals, including ACM CHI and IEEE S&P. He has received several academic honors, notably an AI 2000 Most Influential Scholar Award Honorable Mention (2024).