Card-Based Cryptography Meets Differential Privacy

Reo Eriguchi
National Institute of Advanced Industrial Science and Technology, Tokyo, Japan

Kazumasa Shinagawa
Ibaraki University, Japan
National Institute of Advanced Industrial Science and Technology, Tokyo, Japan

Takao Murakami
The Institute of Statistical Mathematics, Tachikawa, Japan
National Institute of Advanced Industrial Science and Technology, Tokyo, Japan

Abstract

Card-based cryptography studies the problem of implementing cryptographic algorithms in a visual way using physical cards to demonstrate their security properties for those who are unfamiliar with cryptography. In this paper, we initiate the study of card-based implementations of differentially private mechanisms, which are a standard privacy-enhancing technique to publish statistics of databases while protecting the privacy of any particular individual. We start with giving the definition of differential privacy of card-based protocols. As a feasibility result, we present three kinds of protocols using standard binary cards for computing the sum of parties' binary inputs, $f(x_1, \ldots, x_n) = \sum_{i=1}^{n} x_i$ for $x_i \in \{0, 1\}$, under differential privacy. Our first protocol follows the framework of output perturbation, which provides differential privacy by adding noise to exact aggregation results. The protocol needs only two shuffles, and the overheads in the number of cards and the error bound are independent of the number n of parties. Our second and third protocols are based on Randomized Response, which adds noise to each input before aggregation. Compared to the first protocol, they improve the overheads in the number of cards and the error bound in terms of differential privacy parameters at the cost of incurring a multiplicative factor of n. To address a technical challenge of generating non-uniform noise using a finite number of cards, we propose a novel differentially private mechanism based on the hypergeometric distribution, which we believe may be of independent interest beyond applications to card-based cryptography.

2012 ACM Subject Classification Security and privacy → Information-theoretic techniques

Keywords and phrases Card-based cryptography, Differential privacy, Secure computation

Digital Object Identifier 10.4230/LIPIcs.FUN.2024.12

Funding *Reo Eriguchi*: This work was supported in part by JST CREST Grant Number MJCR22M1 and JST AIP Acceleration Research JPMJCR22U5.
Kazumasa Shinagawa: This work was supported in part by JSPS KAKENHI 21K17702 and 23H00479, and JST CREST Grant Number MJCR22M1.
Takao Murakami: This work was supported in part by JSPS KAKENHI JP22H00521.

1 Introduction

With the rapid development of cryptography, various kinds of cryptographic primitives have been proposed and allowed secure data processing on sensitive data. However, most of these primitives are supposed to be implemented by computers and, as such, often lead to complicated algorithm design. As a result, there remains a gap in non-experts' understanding of the security properties, which may prevent active social implementations.

12th International Conference on Fun with Algorithms (FUN 2024).
Editors: Andrei Z. Broder and Tami Tamir; Article No. 12; pp. 12:1–12:20
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

To address this problem, *card-based cryptography* [7, 6] studies the problem of implementing cryptographic algorithms in a visual way using physical cards and demonstrates their security properties for those who are unfamiliar with cryptography. So far, many card-based cryptographic protocols have been proposed to implement secure multiparty computation (e.g., [6, 16, 19, 22]) and zero-knowledge proofs [18, 11].

Recently, the concept of *differential privacy* [10] has been attracting a lot of attention as the gold standard for rigorous privacy guarantees. Differential privacy is a mathematical concept introduced in [8, 9] to quantify the privacy loss associated with any publication of statistics of databases. For example, consider the simplest task of computing the sum of n parties' private inputs. If the exact aggregation result is published, an adversary colluding with $n-1$ parties can deduce the input of the remaining party from the result, which in principle cannot be prevented only by secure computation techniques. Differentially private mechanisms make results untraceable back to individuals by perturbing them with the addition of noise. Due to its strong privacy and robustness guarantees, many differentially private mechanisms have been proposed and deployed in privacy-preserving data analysis of, e.g., users' location information [2, 26] and social network data [21].

We note that differentially private mechanisms were previously supposed to be implemented using computers in the literature. This seems to be in part because the mechanisms usually need complicated processes to generate noise drawn from non-uniform probability distributions (e.g., Laplace or Bernoulli distribution [10]). Towards the further deployment of privacy-enhancing techniques, it is important to demonstrate differentially private mechanisms in an easier-to-understand way. However, the problem of implementing differentially private mechanisms using cards has never been considered prior to this work.

1.1 Our Results

In this paper, we initiate the study of card-based implementations of differentially private mechanisms. We start with giving the definition of differential privacy of card-based protocols. Our definition is inspired by the framework of [3] defining differential privacy of (non-card-based) distributed protocols. As a feasibility result, we present three kinds of protocols using standard binary cards for computing the sum of parties' binary inputs, $f(x_1, \ldots, x_n) = \sum_{i=1}^{n} x_i$ for $x_i \in \{0, 1\}$, under differential privacy. Computing the binary sum function is one of the most fundamental problems within the context of differential privacy [3, 4, 10, 24].

Our first protocol is based on output perturbation, which provides differential privacy by adding noise to exact aggregation results. The protocol needs only two shuffles, and the overheads in the number of cards and a bound on the mean squared error (MSE) are independent of the number n of inputs. A technical challenge is how to generate non-uniform noise using a finite number of cards. Along the way, we propose a novel differentially private mechanism based on the hypergeometric distribution, which we believe may be of independent interest beyond applications to card-based cryptography.

Our second and third protocols are based on input perturbation, which adds noise to each input before aggregation. Compared to the above protocol, they improve the overheads in the number of cards and the error bound in terms of differential privacy parameters at the cost of incurring a multiplicative factor of n. Our third protocol even reduces the number of shuffles to one. While not apparent in asymptotic notations, we empirically show that the second protocol ensures a smaller number of cards and a smaller MSE in concrete parameter settings. The detailed comparison is shown in Table 1 and Section 6.

We note that our first protocol can be described in the traditional model of card-based protocols introduced in [16]. On the other hand, our second and third protocols assume that parties apply private reveals to cards, which is not allowed in the model of [16]. While

Table 1 Comparison of our card-based protocols.

Protocol	MSE	# cards	# shuffles
$\Pi^{\mathrm{HG}}_{k,\ell}$ (Section 4)	$O(\epsilon^{-4}\ln\delta^{-1})$	$n + O(\epsilon^{-5}\ln\delta^{-1})$	2
$\Pi^{\mathrm{RR}}_{k,\ell}$ (Section 5.1)	$O(\epsilon^{-2}n)$	$O(\epsilon^{-1}n)$	n
$\Pi^{\mathrm{RR}'}_{k,\ell}$ (Section 5.2)	$O(\epsilon^{-2}n)$	$O(\epsilon^{-1}n)$	1

ϵ, δ denote differential privacy parameters and n denotes the number of parties (see Section 2 for the definition). We show the asymptotic performance when ϵ tends to 0 and use the approximation $\mathrm{e}^{\epsilon} \approx 1 + \epsilon + \epsilon^2/2$.

such private operations could be easily implemented in practice and are assumed by a prior work [13], constructing protocols without any private operations has been considered as theoretically important in the literature. For that, we show that private operations in our second and third protocols can be removed at the cost of doubling the number of cards and requiring n more shuffles.

1.2 Overview of Our Techniques

We here provide an overview of our protocols. More detailed descriptions and security proofs are given in the following sections.

Our protocols assume standard binary cards with $\heartsuit$ and $\clubsuit$. A framework to guarantee differential privacy is roughly categorized into output perturbation and input perturbation: The former first computes an exact result privately and then perturbs it with the addition of noise; The latter first perturbs private inputs and then computes a target function on the noisy values.

1.2.1 Our Protocol Based on the Hypergeometric Distribution

Our first protocol is based on output perturbation. The private computation of the sum $f(\boldsymbol{x}) = \sum_{i\in[n]} x_i$ of binary inputs is straightforward: If parties submit face-down cards following the encoding rule $\heartsuit = 1$, $\clubsuit = 0$, then the number of $\heartsuit$s in a resulting sequence of cards is equal to $f(\boldsymbol{x})$. A main technical challenge is thus how to implement the addition of noise providing differential privacy using cards. One of the most common choices for the noise distribution is the binomial distribution $\mathrm{Bin}(k, 1/2)$ with the number of trials k and the success probability $1/2$ [1]. A naïve card-based implementation of binomial distributions would be that for each $i = 1, 2, \ldots, k$, parties uniformly permutes a pair of two cards with $\heartsuit$ and $\clubsuit$, and adds one of them to the above sequence in a face-down manner. Although it indeed generates binomial samples, this naïve implementation needs a large number of shuffle operations proportional to k. A state-of-the-art analysis [1] shows that k should be chosen as $k = \Omega(\epsilon^{-2}\ln\delta^{-1})$ to guarantee (ϵ, δ)-differential privacy. Concretely, k should be larger than 2000 for $\epsilon = 0.5$ and $\delta = 10^{-6}$, and even larger than 20000 for $\epsilon = 0.1$ and $\delta = 10^{-6}$.

To reduce the number of shuffles, we prepare a supplementary sequence of randomly shuffled cards containing equal numbers of $\heartsuit$s and $\clubsuit$s, and choose k cards from it without replacement. Intuitively, if the number of $\heartsuit$s in the sequence is sufficiently larger than k, then the number of $\heartsuit$s in the k draws approximately follows the binomial distribution $\mathrm{Bin}(k, 1/2)$. Since we sample cards without replacement, our method requires only a single shuffle to prepare the supplementary sequence for generating noise. It is important to note that the number of $\heartsuit$s in the sequence is actually a finite value. A technical challenge is thus that the noise does not exactly follow the binomial distribution but follows the *hypergeometric distribution*, which precisely describes the distribution of the number of $\heartsuit$s in

k draws without replacement from a sequence of cards containing equal numbers of ♡s and ♣s[1]. We present for the first time the differential privacy guarantee of a mechanism adding noise drawn from the hypergeometric distribution, and also present a utility guarantee in terms of the mean squared error (MSE). We believe that differentially private mechanisms based on the hypergeometric distribution may be of independent interest beyond applications to card-based cryptography.

1.2.2 Our Protocols Based on Randomized Response

Our second and third protocols are based on input perturbation. Specifically, we focus on a traditional mechanism called *Randomized Response* [10, 24], which guarantees differential privacy by having parties flip their input bits with a probability $p = 1/(\mathrm{e}^{\epsilon} + 1)$. A technical challenge here is how to implement biased coins using cards.

Our first realization is a direct implementation of the above procedure: We prepare n supplementary sequences of ℓ cards each consisting of randomly permuted k ♡s and $\ell - k$ ♣s such that $p \approx k/\ell$ and let the i-th party privately open a card in the i-th sequence and flip his input if and only if he draws ♡. In Section 5.1, we carefully analyze the impact of the finite approximation of the probability p on differential privacy, which is not a straightforward problem as there are known attacks on naïve implementations of algorithms assuming real arithmetic [14].

A possible drawback of the above implementation is that the number of shuffle operations grows linearly in the number n of parties since n supplementary sequences should be independently prepared. To reduce the number of shuffles, we propose an alternative implementation: We prepare *one* supplementary sequence consisting of randomly permuted k ♡s and $\ell - k$ ♣s such that $p \approx k/\ell$ and let the i-th party privately open the i-th card in the sequence and flip his input if and only if he draws ♡. This method allows us to prepare the supplementary sequence with only a single shuffle. On the other hand, a more careful analysis of privacy and utility is necessary since the states of cards drawn by parties are no more independent. Note that this kind of challenge has not been encountered in the prior computer-based implementations of Randomized Response or its variants [24, 10, 23] where parties can locally generate independent randomness.

Finally, both of the above implementations require parties to apply private reveals to cards. While such private operations could be easily implemented in practice, it has also been considered as theoretically important to construct protocols without any private operations (i.e., those following the traditional model of card-based protocols [16]). We also show that private operations can be removed by emulating the local computations done by parties with card-based secure computation protocols. In the above implementations, parties need to privately compute the XOR of their inputs and the states of cards drawn from supplementary sequences. We emulate these computations with an efficient card-based XOR protocol without any private operations [17]. As a result, we obtain variant protocols removing private operations at the cost of doubling the number of cards and requiring n more shuffles. Note that our first protocol based on the hypergeometric distribution can be described following the model of [16] as it assumes no private operation.

[1] The hypergeometric distribution can be defined in a more general setting where a sequence contains different numbers of ♡s and ♣s.

2 Preliminaries

Notations. For $n \in \mathbb{N}$, define $[n] = \{i \in \mathbb{Z} : 1 \leq i \leq n\}$. If a random variable z follows a probability distribution $\mathcal{D}$, we write $z \leftarrow \mathcal{D}$. Let $\ln x$ denote the base-e logarithm of x, where e is the Napiers constant.

2.1 Card-based Protocols

Card. In this paper, we use *binary cards* whose front sides are either $\boxed{\clubsuit}$ or $\boxed{\heartsuit}$ and back sides are both $\boxed{?}$. We assume that two cards with the same symbol are indistinguishable. We use the encoding $\clubsuit = 0$ and $\heartsuit = 1$ throughout the paper except in Section 7.

Shuffle. A *shuffle* is an operation that applies a random permutation to a sequence of face-down cards, where the permutation is chosen by some probability distribution. It is assumed that no party guesses which permutation is chosen from the shuffle.

A *complete shuffle* is a shuffle that applies a uniformly random permutation to a sequence of face-down cards, which is denoted by $[\cdot]$. For example, a complete shuffle for a sequence of three cards results in one of the six sequences each with probability $1/6$ as follows:

$$\left[\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\overset{3}{\boxed{?}}\right] \to \overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\overset{3}{\boxed{?}} \text{ or } \overset{1}{\boxed{?}}\overset{3}{\boxed{?}}\overset{2}{\boxed{?}} \text{ or } \overset{2}{\boxed{?}}\overset{3}{\boxed{?}}\overset{1}{\boxed{?}} \text{ or } \overset{2}{\boxed{?}}\overset{1}{\boxed{?}}\overset{3}{\boxed{?}} \text{ or } \overset{3}{\boxed{?}}\overset{1}{\boxed{?}}\overset{2}{\boxed{?}} \text{ or } \overset{3}{\boxed{?}}\overset{2}{\boxed{?}}\overset{1}{\boxed{?}}.$$

A *pile-scramble shuffle* [12] is a shuffle that divides a sequence of cards into multiple *piles* of the same number of cards and applies a random permutation to a sequence of piles, which is denoted by $[\cdot|\cdots|\cdot]$. For example, a pile-scramble shuffle for a sequence of three piles each having two cards results in one of the six sequences each with probability $1/6$ as follows:

$$\left[\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\,\middle|\,\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\,\middle|\,\overset{5}{\boxed{?}}\overset{6}{\boxed{?}}\right] \to \left\{\begin{array}{l} \overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\;\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\;\overset{5}{\boxed{?}}\overset{6}{\boxed{?}} \\ \overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\;\overset{5}{\boxed{?}}\overset{6}{\boxed{?}}\;\overset{3}{\boxed{?}}\overset{4}{\boxed{?}} \\ \overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\;\overset{5}{\boxed{?}}\overset{6}{\boxed{?}}\;\overset{1}{\boxed{?}}\overset{2}{\boxed{?}} \\ \overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\;\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\;\overset{5}{\boxed{?}}\overset{6}{\boxed{?}} \\ \overset{5}{\boxed{?}}\overset{6}{\boxed{?}}\;\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\;\overset{3}{\boxed{?}}\overset{4}{\boxed{?}} \\ \overset{5}{\boxed{?}}\overset{6}{\boxed{?}}\;\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\;\overset{1}{\boxed{?}}\overset{2}{\boxed{?}} \end{array}\right. .$$

Protocol. Suppose that there are n parties each having an input $x_i \in D$, where the input domain D is a finite set. A card-based protocol consists of three phases: the setup phase, the computation phase, and the output phase. In the setup phase, *supplementary cards* are prepared. Using shuffles, they are drawn from a probability distribution independent of parties' inputs. Here, the front sides of them are hidden from all parties. In the computation phase, parties repeat one of the following operations:

- Input: Each party submits a face-down card according to his/her input. They are called *main cards*. If necessary, it is allowed to perform *private reveals* for a subset of the supplementary cards (e.g., [13, 20, 25]), where a designated party privately reads the symbol of a face-down card.
- Shuffle: A random permutation is applied to the current sequence of cards consisting of the main cards and the supplementary cards. It is assumed that no party guesses which permutation is chosen from the shuffle.
- Insertion: Some of the supplementary cards are inserted to the main cards.

In the output phase, parties open all cards in the current sequence and determine the output value. Note that if parties do not perform private reveals, then a protocol can be described in the traditional model of card-based protocols [16].

We evaluate the space complexity of a card-based protocol Π by the total number of cards used to execute Π, which we denote by $\#\mathsf{Card}(\Pi)$. We also evaluate the computational complexity of Π by the total number of shuffle operations since shuffling is the most costly operation in practice [15]. We denote it by $\#\mathsf{Shuffle}(\Pi)$.

2.2 Differential Privacy

Following the terminology in [3], we say that two n-dimensional vectors $\boldsymbol{x} = (x_i)_{i\in[n]}$, $\boldsymbol{x}' = (x'_i)_{i\in[n]}$ are T*-neighboring* for a subset $T \subseteq [n]$ if $x_i = x'_i$ for any $i \in T$ and $x_i \neq x'_i$ for at most one $i \notin T$. If $\boldsymbol{x}$ and $\boldsymbol{x}'$ are $\emptyset$-neighboring, we simply say that they are neighboring. For a finite set D, we define the *sensitivity* of a function $f : D^n \to \mathbb{Z}$ as

$$\Delta = \max_{\substack{\boldsymbol{x},\boldsymbol{x}'\in D^n:\\ \text{neighboring}}} |f(\boldsymbol{x}) - f(\boldsymbol{x}')|.$$

We say that two probability distributions $\mathcal{D}_1, \mathcal{D}_2$ over a set U are (ϵ, δ)*-DP close* if for any subset $S \subseteq U$, it holds that $\Pr[y \leftarrow \mathcal{D}_1 : y \in S] \leq \mathrm{e}^{\epsilon} \cdot \Pr[y \leftarrow \mathcal{D}_2 : y \in S] + \delta$.

3 Differentially Private Card-based Protocols

We start with giving the definition of differential privacy of card-based protocols. Our definition is inspired by the framework of [3] defining differential privacy of (non-card-based) distributed protocols. In this paper, we consider an adversary corrupting a set T of at most $n - 1$ parties. We assume that the adversary is *semi-honest*, that is, she tries to learn information from her view during the protocol but does not deviate from the protocol specifications. Let $\mathsf{View}_{\Pi,T}(\boldsymbol{x})$ denote the view of the adversary during the execution of a card-based protocol Π on input $\boldsymbol{x} = (x_1, \dots, x_n)$, which consists of the inputs of the corrupted parties and the information (e.g., the states of cards) that they can learn during the execution of Π.

▶ **Definition 1.** *Let ϵ, δ be non-negative numbers. We say that a card-based protocol Π is (ϵ, δ)-differentially private if for any set T of at most $n - 1$ parties and any pair $(\boldsymbol{x}, \boldsymbol{x}')$ of T-neighboring vectors, $\mathsf{View}_{\Pi,T}(\boldsymbol{x})$ and $\mathsf{View}_{\Pi,T}(\boldsymbol{x}')$ are (ϵ, δ)-DP close.*

We evaluate the utility of a protocol Π with respect to a function $f : D^n \to \mathbb{Z}$ by its mean squared error (MSE) defined as

$$\mathsf{MSE}_f(\Pi) = \max_{\boldsymbol{x}\in D^n} \mathbb{E}\left[|\Pi(\boldsymbol{x}) - f(\boldsymbol{x})|^2\right],$$

where $\Pi(\boldsymbol{x})$ is a random variable corresponding to the output of Π on input $\boldsymbol{x}$.

In this paper, we focus on the setting in which every party has a bit $x_i \in \{0, 1\}$ and they compute the binary sum $f(x_1, \dots, x_n) = x_1 + \cdots + x_n$. Note that the sensitivity of f is $\Delta = 1$.

4 Our Protocol Based on the Hypergeometric Distribution

The hypergeometric distribution is a probability distribution of the number Z of ♡s in k cards chosen uniformly at random from a sequence consisting of $m - \ell$ ♣s and ℓ ♡s. Formally, we define the distribution as follows:

▶ **Definition 2.** *Let k, ℓ, m be positive integers such that $k \leq \ell$ and $k + \ell \leq m$. A random variable Z follows the hypergeometric distribution* $\mathrm{HG}(m, \ell, k)$ *if its probability mass function is given by*

$$\Pr[Z = z] = p_{\mathrm{HG}}(z) = \frac{\binom{\ell}{z}\binom{m-\ell}{k-z}}{\binom{m}{k}},\ z = 0, 1, \ldots, k.$$

First, we show that hypergeometric distributions are able to provide differential privacy.

▶ **Proposition 3.** *Let k, ℓ be positive integers such that $k < \ell$, and α, β be real numbers such that*

$$\alpha \geq \frac{\ell}{\ell - k} \text{ and } \beta > 1.$$

Let $f : D^n \to \mathbb{Z}$ be a function with sensitivity Δ. Let ϵ and δ be real numbers such that

$$\epsilon \geq \Delta \ln(\alpha\beta) \text{ and } \delta \geq \exp\left(-\frac{k}{2}\left(\frac{\beta - 1}{\beta + 1} - \frac{2\Delta}{k}\right)^2\right). \tag{1}$$

Define a randomized algorithm $\mathcal{M}$ as

$$\mathcal{M}(\boldsymbol{x}) = f(\boldsymbol{x}) + z,\ z \leftarrow \mathrm{HG}(2\ell, \ell, k).$$

Then, for any pair $(\boldsymbol{x}, \boldsymbol{x}')$ of neighboring vectors, $\mathcal{M}(\boldsymbol{x})$ and $\mathcal{M}(\boldsymbol{x}')$ are (ϵ, δ)-DP close.

Proof. It is sufficient to show that

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

for any subset $S \subseteq \mathbb{Z}$. Let $y = f(\boldsymbol{x})$, $y' = f(\boldsymbol{x}')$. We assume that $y \leq y'$. The case of $y \geq y'$ can be dealt with similarly. Since $z \leftarrow \mathrm{HG}(2\ell, \ell, k)$ takes values between 0 and k, we may assume that $S \subseteq \{s \in \mathbb{Z} : y \leq s \leq y' + k\}$. Letting $z_0 = k/(\beta + 1)$, we decompose S into three subsets: $S_1 = \{s \in S : s \leq y' + z_0\}$, $S_2 = \{s \in S : y' + z_0 < s \leq y + k\}$, and $S_3 = \{s \in S : y + k < s \leq y' + k\}$. We will show that

$$\Pr[\mathcal{M}(\boldsymbol{x}) \in S_1] \leq \delta \text{ and } \Pr[\mathcal{M}(\boldsymbol{x}) = s] \leq \mathrm{e}^{\epsilon} \cdot \Pr[\mathcal{M}(\boldsymbol{x}') = s]\ (\forall s \in S_2).$$

If this is shown, since $\Pr[\mathcal{M}(\boldsymbol{x}) \in S_3] = \Pr[z \leftarrow \mathrm{HG}(2\ell, \ell, k) : z > k] = 0$, we obtain that

$$\begin{aligned}
\Pr[\mathcal{M}(\boldsymbol{x}) \in S] &\leq \Pr[\mathcal{M}(\boldsymbol{x}) \in S_1] + \sum_{s \in S_2} \Pr[\mathcal{M}(\boldsymbol{x}) = s] \\
&\leq \delta + \sum_{s \in S_2} \mathrm{e}^{\epsilon} \cdot \Pr[\mathcal{M}(\boldsymbol{x}') = s] \\
&\leq \mathrm{e}^{\epsilon} \cdot \Pr[\mathcal{M}(\boldsymbol{x}') \in S] + \delta.
\end{aligned}$$

First, since $y' \leq y + \Delta$ and the mean of $\mathrm{HG}(2\ell, \ell, k)$ is $k\ell/(2\ell) = k/2$, the Chernoff inequality [5] implies that

$$\Pr[\mathcal{M}(\boldsymbol{x}) \in S_1] \leq \sum_{0 \leq z \leq \Delta + z_0} p_{\mathrm{HG}}(z) \leq \exp(-2t^2 k),$$

where

$$t = \frac{1}{2} - \frac{\Delta + z_0}{k} = \frac{1}{2}\left(\frac{\beta - 1}{\beta + 1} - \frac{2\Delta}{k}\right)$$

We thus obtain that

$$\Pr[\mathcal{M}(\boldsymbol{x}) \in S_1] \le \exp\left(-\frac{k}{2}\left(\frac{\beta-1}{\beta+1} - \frac{2\Delta}{k}\right)^2\right) \le \delta.$$

Next, let $s \in S_2$ and set $z = s - y, z' = s - y'$. We then obtain that $\max\{z - \Delta, z_0\} \le z' \le z \le k$, and

$$\begin{aligned}
\frac{\Pr[\mathcal{M}(\boldsymbol{x}) = s]}{\Pr[\mathcal{M}(\boldsymbol{x}') = s]} &= \frac{p_{\mathrm{HG}}(z)}{p_{\mathrm{HG}}(z')} \\
&= \frac{\binom{\ell}{z}\binom{\ell}{k-z}}{\binom{\ell}{z'}\binom{\ell}{k-z'}} \\
&= \prod_{z'<i\le z} \frac{k+1-i}{i} \prod_{\ell-k+z'<i\le \ell-k+z} \frac{2\ell-k+1-i}{i} \\
&\le \left(\frac{k+1}{z'+1} - 1\right)^{z-z'} \left(\frac{2\ell-k+1}{\ell-k+z'+1} - 1\right)^{z-z'} \\
&\le \left(\frac{k}{z_0} - 1\right)^{z-z'} \alpha^{z-z'} \quad (\because z' \ge z_0) \\
&= (\alpha\beta)^{z-z'}.
\end{aligned}$$

Here, we use the fact that

$$\ell \ge \frac{\alpha}{\alpha-1}k \ge \frac{\alpha}{\alpha-1}(k-1) - \frac{\alpha+1}{\alpha-1}z_0$$

and hence $(\ell - z_0)/(\ell - k + z_0 + 1) \le \alpha$. Since $\alpha\beta > 1$, we obtain that

$$\frac{\Pr[\mathcal{M}(\boldsymbol{x}) = s]}{\Pr[\mathcal{M}(\boldsymbol{x}') = s]} \le (\alpha\beta)^{\Delta} \le \mathrm{e}^{\epsilon}$$ ◀

We show a protocol $\Pi^{\mathrm{HG}}_{k,\ell}$ based on the hypergeometric distribution in Figure 1. The following theorem shows the differential privacy, MSE and complexities of $\Pi^{\mathrm{HG}}_{k,\ell}$.

▶ **Theorem 4.** *Let ϵ, δ be positive real numbers such that $\delta < 1/\sqrt{\mathrm{e}}$. Let k, ℓ be integers such that*

$$k \ge 4\left(\frac{\mathrm{e}^{\epsilon}+1+\epsilon}{\mathrm{e}^{\epsilon}-1-\epsilon}\right)^2 \ln\frac{1}{\delta} + \frac{2(\mathrm{e}^{\epsilon}+1+\epsilon)}{\mathrm{e}^{\epsilon}-1-\epsilon} \text{ and } \ell \ge \left(1+\frac{1}{\epsilon}\right)k. \tag{2}$$

Then, the protocol $\Pi^{\mathrm{HG}}_{k,\ell}$ satisfies (ϵ, δ)-differential privacy. The MSE of $\Pi^{\mathrm{HG}}_{k,\ell}$ with respect to $f : \{0,1\}^n \ni (x_i)_{i\in[n]} \mapsto \sum_{i\in[n]} x_i \in \mathbb{Z}$ is

$$\mathsf{MSE}_f(\Pi^{\mathrm{HG}}_{k,\ell}) = \frac{k(2\ell-k)}{4(2\ell-1)}. \tag{3}$$

The complexities of $\Pi^{\mathrm{HG}}_{k,\ell}$ are

$$\#\mathsf{Card}(\Pi^{\mathrm{HG}}_{k,\ell}) = n + 2\ell = n + O\left(\frac{\mathrm{e}^{2\epsilon}}{\epsilon(\mathrm{e}^{\epsilon}-1-\epsilon)^2}\ln\frac{1}{\delta}\right) \text{ and } \#\mathsf{Shuffle}(\Pi^{\mathrm{HG}}_{k,\ell}) = 2.$$

Setup: Arrange a sequence of 2ℓ face-down cards consisting of ℓ $\boxed{\heartsuit}$s and ℓ $\boxed{\clubsuit}$s:

$$\underbrace{\boxed{\clubsuit}\boxed{\clubsuit}\cdots\boxed{\clubsuit}}_{\ell\text{ cards}}\;\underbrace{\boxed{\heartsuit}\boxed{\heartsuit}\cdots\boxed{\heartsuit}}_{\ell\text{ cards}} \rightarrow \underbrace{\boxed{?}\boxed{?}\cdots\boxed{?}}_{2\ell\text{ cards}}.$$

Apply a complete shuffle to the sequence:

$$\left[\,\boxed{?}\boxed{?}\cdots\boxed{?}\,\right] \rightarrow \boxed{?}\boxed{?}\cdots\boxed{?}.$$

We set them as supplementary cards.

Input: Following the encoding rule $\clubsuit = 0$ and $\heartsuit = 1$, the i-th party submits a face-down card corresponding to x_i. Then we have the following sequence of cards:

$$\underset{x_1}{\boxed{?}}\underset{x_2}{\boxed{?}}\cdots\underset{x_n}{\boxed{?}}.$$

We set them as main cards.

Insertion: Append any k out of the 2ℓ supplementary cards (e.g., the leftmost k cards) to the main cards:

$$\underbrace{\boxed{?}\boxed{?}\cdots\boxed{?}}_{\text{main cards}}\underbrace{\boxed{?}\boxed{?}\cdots\boxed{?}}_{k\text{ cards}}.$$

Shuffle: Apply a complete shuffle to the sequence of the $n+k$ cards:

$$\left[\,\boxed{?}\boxed{?}\cdots\boxed{?}\,\right] \rightarrow \boxed{?}\boxed{?}\cdots\boxed{?}.$$

Output: Open all the $n+k$ cards:

$$\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\cdots\boxed{?} \rightarrow \boxed{\heartsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}\cdots\boxed{\heartsuit}.$$

Output $y - k/2$, where y is the number of $\boxed{\heartsuit}$ in the opened cards.

Figure 1 A protocol $\Pi^{\mathrm{HG}}_{k,\ell}$.

Proof. First, we show the differential privacy of the protocol $\Pi^{\mathrm{HG}}_{k,\ell}$. Let T be a set of corrupted parties such that $|T| \leq n-1$, and let $\boldsymbol{x} = (x_i)_{i\in[n]}, \boldsymbol{x}' = (x'_i)_{i\in[n]} \in \{0,1\}^n$ be T-neighboring inputs. Define Y (resp. Y') be random variables corresponding to the number y computed during the execution of $\Pi^{\mathrm{HG}}_{k,\ell}$ on input $\boldsymbol{x}$ (resp. $\boldsymbol{x}'$). Note that $x_i = x'_i$ for all $i \in T$ and the cards opened during the protocol are a uniformly random permutation of y $\heartsuit$s and $n+k-y$ $\clubsuit$s. Thus, the distributions of $\mathsf{View}_{\Pi^{\mathrm{HG}}_{k,\ell},T}(\boldsymbol{x})$ and $\mathsf{View}_{\Pi^{\mathrm{HG}}_{k,\ell},T}(\boldsymbol{x}')$ can be simulated from Y and Y', respectively. From the post-processing property of differential privacy, it is sufficient to show that Y and Y' are (ϵ,δ)-DP close.

If parties' inputs are $\boldsymbol{x}$, then the number of $\heartsuit$s included in the main cards is $f(\boldsymbol{x})$ just after all parties submit their cards. Furthermore, the number of $\heartsuit$s included in k cards drawn from supplementary cards follows the distribution $\mathrm{HG}(2\ell,\ell,k)$. Thus, the number Y of $\heartsuit$s included in the main cards at the end of the protocol follows the same distribution as $\mathcal{M}(\boldsymbol{x}) = f(\boldsymbol{x}) + z,\ z \leftarrow \mathrm{HG}(2\ell,\ell,k)$. Similarly, Y' follows the same distribution as $\mathcal{M}(\boldsymbol{x}')$.

Define α and β as $\alpha = \ell/(\ell-k)$ and $\beta = \mathrm{e}^\epsilon/(1+\epsilon)$, respectively. Since $\epsilon > 0$, we have that $\beta > 1$. Furthermore, since $\ell \geq (1+\epsilon^{-1})k$, it holds that $\alpha \leq 1+\epsilon$. We then obtain that

$$\alpha\beta \leq \mathrm{e}^\epsilon. \tag{4}$$

We will show that

$$\frac{k}{2}\left(\frac{\mathrm{e}^\epsilon - 1 - \epsilon}{\mathrm{e}^\epsilon + 1 + \epsilon} - \frac{2}{k}\right)^2 \geq \ln \delta^{-1}. \tag{5}$$

If this is shown, the condition (4) and the assumption that the sensitivity of f is $\Delta = 1$ imply the condition (1), and hence it follows from Proposition 3 that Y and Y' are (ϵ, δ)-DP close. Let

$$a = \frac{\mathrm{e}^\epsilon - 1 - \epsilon}{\mathrm{e}^\epsilon + 1 + \epsilon},\quad b = \sqrt{2\ln\delta^{-1}},\quad \text{and } t = \sqrt{k}.$$

Then, the condition (5) is equivalent to $(at - 2/t)^2 \geq b^2$, i.e., $at^2 - bt - 2 \geq 0$. Furthermore, it is equivalent to

$$k = t^2 \geq \frac{b^2}{2a^2}\left(1 + \sqrt{1 + \frac{8a}{b^2}}\right) + \frac{2}{a}.$$

Since $\delta \leq 1/\sqrt{\mathrm{e}}$, we have that $a \leq 1$ and $b \geq 1$. Thus, it holds that

$$\frac{b^2}{2a^2}\left(1 + \sqrt{1 + \frac{8a}{b^2}}\right) + \frac{2}{a} \leq \frac{2b^2}{a^2} + \frac{2}{a} = 4\left(\frac{\mathrm{e}^\epsilon + 1 + \epsilon}{\mathrm{e}^\epsilon - 1 - \epsilon}\right)^2 \ln\frac{1}{\delta} + \frac{2(\mathrm{e}^\epsilon + 1 + \epsilon)}{\mathrm{e}^\epsilon - 1 - \epsilon}.$$

The condition (5) then follows from the condition (2).

Finally, we analyze the utility of $\Pi^{\mathrm{HG}}_{k,\ell}$. If parties' inputs are $\boldsymbol{x}$, the output of the protocol is given as $y - k/2 = f(\boldsymbol{x}) + z - k/2$, $z \leftarrow \mathrm{HG}(2\ell, \ell, k)$. Since the mean of the hypergeometric distribution $\mathrm{HG}(2\ell, \ell, k)$ is $k/2$, $\mathsf{MSE}_f(\Pi^{\mathrm{HG}}_{k,\ell})$ is equal to the variance of $\mathrm{HG}(2\ell, \ell, k)$. We therefore obtain (3). ◀

5 Our Protocols Based on Randomized Response

Randomized Response [10, 24] guarantees differential privacy by having parties flip their input bits with a certain probability p. Specifically, for a privacy parameter $\epsilon > 0$, let p be such that

$$\frac{1}{\mathrm{e}^\epsilon + 1} \leq p < \frac{1}{2}. \tag{6}$$

We define an algorithm $\mathcal{R}_p$ as follows: On input $x \in \{0,1\}$, $\mathcal{R}_p$ chooses $r \in \{0,1\}$ according to the Bernoulli distribution with parameter p, i.e.,

$$\Pr[r = 1] = p \text{ and } \Pr[r = 0] = 1 - p,$$

and then outputs $y = x \oplus r$. The condition (6) implies that $\Pr[\mathcal{R}_p(x) = b] \leq \mathrm{e}^\epsilon \cdot \Pr[\mathcal{R}_p(1-x) = b]$ for any $x, b \in \{0,1\}$. Hence $\mathcal{R}_p(0)$ and $\mathcal{R}_p(1)$ are $(\epsilon, 0)$-DP close.

Setup: Arrange n sequences of ℓ face-down cards each consisting of k $\boxed{\heartsuit}$s and $\ell-k$ $\boxed{\clubsuit}$s:

$$\underbrace{\boxed{\heartsuit}\boxed{\heartsuit}\cdots\boxed{\heartsuit}}_{k\text{ cards}}\;\underbrace{\boxed{\clubsuit}\boxed{\clubsuit}\cdots\boxed{\clubsuit}}_{\ell-k\text{ cards}}\;\to\;\underbrace{\boxed{?}\boxed{?}\cdots\boxed{?}}_{\ell\text{ cards}}.$$

Apply a complete shuffle to each of the n sequences:

$$\left[\,\boxed{?}\boxed{?}\cdots\boxed{?}\,\right]\;\to\;\boxed{?}\boxed{?}\cdots\boxed{?}.$$

We set the whole sequence as supplementary cards, and call the i-th sub-sequence as the i-th sequence of the supplementary cards.

Input: The i-th party performs a private reveal for any card (e.g., the leftmost one) in the i-th sequence of the supplementary cards. Let $r_i \in \{\clubsuit, \heartsuit\}$ be the opened symbol. Following the encoding rule $\clubsuit = 0$ and $\heartsuit = 1$, the i-th party submits a face-down card corresponding to $x_i \oplus r_i$. Then we have the following sequence of cards:

$$\underset{x_1\oplus r_1}{\boxed{?}}\;\underset{x_2\oplus r_2}{\boxed{?}}\;\cdots\;\underset{x_n\oplus r_n}{\boxed{?}}\;.$$

We set them as main cards.

Output: Open all the n cards:

$$\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\cdots\boxed{?}\;\to\;\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\heartsuit}\cdots\boxed{\clubsuit}.$$

Output $z = \frac{y-nk/\ell}{1-2k/\ell}$, where y is the number of $\boxed{\heartsuit}$ in the opened cards.

Figure 2 A protocol $\Pi^{\mathrm{RR}}_{k,\ell}$.

5.1 A Direct Implementation

Our first realization, denoted by $\Pi^{\mathrm{RR}}_{k,\ell}$, is a direct implementation of the above procedure: We prepare n sequences each consisting of randomly permuted k $\heartsuit$s and $\ell-k$ $\clubsuit$s such that $p \approx k/\ell$ and let the i-th party privately open a card in the i-th sequence and flip his input if and only if he draws $\heartsuit$. The formal description of $\Pi^{\mathrm{RR}}_{k,\ell}$ is given in Figure 2.

The following theorem shows the differential privacy, MSE and complexities of $\Pi^{\mathrm{RR}}_{k,\ell}$.

▶ **Theorem 5.** *Let ϵ be a positive real number. Let k, ℓ be integers such that*

$$\ell \geq \frac{3(\mathrm{e}^{\epsilon}+1)}{\mathrm{e}^{\epsilon}-1} \text{ and } \frac{1}{\mathrm{e}^{\epsilon}+1} \leq p := \frac{k}{\ell} \leq \frac{\mathrm{e}^{\epsilon}+2}{3(\mathrm{e}^{\epsilon}+1)}. \tag{7}$$

Then, the protocol $\Pi^{\mathrm{RR}}_{k,\ell}$ satisfies $(\epsilon, 0)$-differential privacy. The MSE of $\Pi^{\mathrm{RR}}_{k,\ell}$ with respect to $f : \{0,1\}^n \ni (x_i)_{i\in[n]} \mapsto \sum_{i\in[n]} x_i \in \mathbb{Z}$ satisfies

$$\mathsf{MSE}_f(\Pi^{\mathrm{RR}'}_{k,\ell}) = \frac{np(1-p)}{(1-2p)^2} \leq \frac{n(\mathrm{e}^{\epsilon}+2)(2\mathrm{e}^{\epsilon}+1)}{(\mathrm{e}^{\epsilon}-1)^2}.$$

The complexities of $\Pi^{\mathrm{RR}}_{k,\ell}$ are

$$\#\mathsf{Card}(\Pi^{\mathrm{RR}}_{k,\ell}) = n(\ell+1) = O\left(\frac{n\mathrm{e}^{\epsilon}}{\mathrm{e}^{\epsilon}-1}\right) \text{ and } \#\mathsf{Shuffle}(\Pi^{\mathrm{RR}}_{k,\ell}) = n.$$

Proof. To begin with, it holds that

$$\frac{\mathrm{e}^{\epsilon}+2}{3(\mathrm{e}^{\epsilon}+1)}-\frac{1}{\mathrm{e}^{\epsilon}+1}=\frac{\mathrm{e}^{\epsilon}-1}{3(\mathrm{e}^{\epsilon}+1)}\geq\frac{1}{\ell}.$$

Hence, there indeed exists integers k, ℓ satisfying the condition (7).

To see the differential privacy of the protocol $\Pi^{\mathrm{RR}}_{k,\ell}$, let $Y_i(\boldsymbol{x})$ denote a random variable corresponding to the state of the card that the i-th party submits to main cards when parties' inputs are $\boldsymbol{x}$. Let T be a set of corrupted parties such that $|T| \leq n-1$, and let $\boldsymbol{x} = (x_i)_{i\in[n]}, \boldsymbol{x}' = (x'_i)_{i\in[n]} \in \{0,1\}^n$ be T-neighboring inputs. For any $i \in [n]$, the distribution of the state r_i of the card that the i-th party draws from supplementary cards is given as $\Pr[r_i = \clubsuit] = 1-p$ and $\Pr[r_i = \heartsuit] = p$. Furthermore, since the 1-to-n-th sub-sequences are prepared independently, $r_1, \ldots, r_n$ are independent. Thus, if we encode $\heartsuit = 1, \clubsuit = 0$, then $Y_i(\boldsymbol{x}) = \mathcal{R}_p(x_i)$. Similarly, we have that $Y_i(\boldsymbol{x}') = \mathcal{R}_p(x'_i)$. We also have that $p < 1/2$ since

$$\frac{1}{2}-\frac{\mathrm{e}^{\epsilon}+2}{3(\mathrm{e}^{\epsilon}+1)}=\frac{\mathrm{e}^{\epsilon}-1}{6(\mathrm{e}^{\epsilon}+1)}>0.$$

The condition (6) is then satisfied and the differential privacy of the algorithm $\mathcal{R}_p$ implies that $(Y_i(\boldsymbol{x}))_{i\notin T}$ and $(Y_i(\boldsymbol{x}'))_{i\notin T}$ are $(\epsilon, 0)$-DP close. The adversarys view during the execution of the protocol on input $\boldsymbol{x}$ (resp. $\boldsymbol{x}'$) can be simulated from $((x_i)_{i\in T}, (Y_i(\boldsymbol{x}))_{i\notin T})$ (resp. $((x_i)_{i\in T}, (Y_i(\boldsymbol{x}))_{i\notin T})$). Since $x_i = x'_i$ ($\forall i \in T$), the post-processing property implies that $\Pi^{\mathrm{RR}}_{k,\ell}$ is $(\epsilon, 0)$-differentially private.

To analyze the utility of $\Pi^{\mathrm{RR}}_{k,\ell}$, let $\boldsymbol{x} \in \{0,1\}^n$. For ease of notations, we write $Y_i = Y_i(\boldsymbol{x}), s = \sum_{i\in[n]} x_i$. Note that $Y_i = 1$ if and only if the i-th party submits $\heartsuit$ to main cards, and $Y_i = 0$ if and only if he submits $\clubsuit$. Furthermore, $\sum_{i\in[n]} Y_i$ is equal to the total number y of $\heartsuit$s included in main cards.

Since $x^2 = x$ if $x \in \{0,1\}$, the expectation and variance of Y_i are given by

$$\begin{aligned}&\mathbb{E}[Y_i] = 1\cdot\Pr[Y_i = 1] = (1-2p)x_i + p \text{ and}\\ &\mathrm{Var}[Y_i] = \mathbb{E}\left[Y_i^2\right] - (\mathbb{E}[Y_i])^2 = p(1-p) + (1-2p)^2 x_i - (1-2p)^2 x_i^2 = p(1-p).\end{aligned}$$

Since $\mathbb{E}\left[(\sum_{i\in[n]} Y_i - np)/(1-2p)\right] = s$, the expectation of an output z of $\Pi^{\mathrm{RR}}_{k,\ell}$ is $s = f(\boldsymbol{x})$. Hence, $\mathrm{MSE}_{\Pi^{\mathrm{RR}'}_{k,\ell}}(f)$ is given by the variance of z. Since the 1-to-n-th sub-sequences of supplementary cards are prepared independently, $Y_1, \ldots, Y_n$ are independent and

$$\mathrm{Var}[z] = \frac{1}{(1-2p)^2}\mathrm{Var}\left[\sum_{i\in[n]} Y_i\right] = \frac{1}{(1-2p)^2}\sum_{i\in[n]}\mathrm{Var}\left[Y_i\right] = \frac{np(1-p)}{(1-2p)^2}.$$

On the other hand, $g(t) := t(1-t)/(1-2t)^2$ is monotonically increasing with respect to t. We therefore conclude that $\mathrm{Var}[z] \leq n(\mathrm{e}^{\epsilon}+2)(2\mathrm{e}^{\epsilon}+1)/(\mathrm{e}^{\epsilon}-1)^2$. ◀

5.2 Reducing the Number of Shuffles

A possible drawback of our first realization is that the number of shuffles grows linearly in the number of parties. In this section, we propose an alternative implementation denoted by $\Pi^{\mathrm{RR}'}_{k,\ell}$: We prepare *one* supplementary sequence consisting of randomly permuted k $\heartsuit$s and $\ell - k$ $\clubsuit$s such that $p \approx k/\ell$ and let the i-th party privately open the i-th card in the sequence and flip his input if and only if he draws $\heartsuit$. The formal description of $\Pi^{\mathrm{RR}'}_{k,\ell}$ is given in Figure 3.

The following theorem shows the differential privacy, MSE and complexities of $\Pi^{\mathrm{RR}'}_{k,\ell}$.

Setup: Arrange a sequence of ℓ face-down cards consisting of k $\boxed{\heartsuit}$s and $\ell - k$ $\boxed{\clubsuit}$s:

$$\underbrace{\boxed{\heartsuit}\boxed{\heartsuit}\cdots\boxed{\heartsuit}}_{k\text{ cards}}\ \underbrace{\boxed{\clubsuit}\boxed{\clubsuit}\cdots\boxed{\clubsuit}}_{\ell-k\text{ cards}} \to \underbrace{\boxed{?}\boxed{?}\cdots\boxed{?}}_{\ell\text{ cards}}.$$

Apply a complete shuffle to the sequence:

$$\left[\boxed{?}\boxed{?}\cdots\boxed{?}\right] \to \boxed{?}\boxed{?}\cdots\boxed{?}.$$

We set them as supplementary cards.

Input: The i-th party performs a private reveal for the i-th card in the supplementary cards. Let $r_i \in \{\clubsuit, \heartsuit\}$ be the opened symbol. Following the encoding rule $\clubsuit = 0$ and $\heartsuit = 1$, the i-th party submits a face-down card corresponding to $x_i \oplus r_i$. Then we have the following sequence of cards:

$$\underset{x_1\oplus r_1}{\boxed{?}}\ \underset{x_2\oplus r_2}{\boxed{?}}\ \cdots\ \underset{x_n\oplus r_n}{\boxed{?}}\ .$$

We set them as main cards.

Output: Open all the n cards:

$$\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\cdots\boxed{?} \to \boxed{\heartsuit}\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\heartsuit}\cdots\boxed{\clubsuit}.$$

Output $z = \frac{y - nk/\ell}{1-2k/\ell}$, where y is the number of $\boxed{\heartsuit}$ in the opened cards.

Figure 3 A protocol $\Pi^{\mathrm{RR}'}_{k,\ell}$.

▶ **Theorem 6.** *Let ϵ be a positive real number and assume that $n \geq 2$. Let k, ℓ be integers such that*

$$\alpha := \frac{n}{\ell} \leq \frac{\mathrm{e}^\epsilon - 1}{5\mathrm{e}^\epsilon} \text{ and } \frac{1+\alpha\mathrm{e}^\epsilon}{\mathrm{e}^\epsilon+1} \leq p := \frac{k}{\ell} \leq \frac{1+2\alpha\mathrm{e}^\epsilon}{\mathrm{e}^\epsilon+1}. \tag{8}$$

Then, the protocol $\Pi^{\mathrm{RR}'}_{k,\ell}$ satisfies $(\epsilon, 0)$-differential privacy. The MSE of $\Pi^{\mathrm{RR}'}_{k,\ell}$ with respect to $f : \{0,1\}^n \ni (x_i)_{i\in[n]} \mapsto \sum_{i\in[n]} x_i \in \mathbb{Z}$ satisfies

$$\mathsf{MSE}_f(\Pi^{\mathrm{RR}}_{k,\ell}) \leq \frac{25n(1+2\alpha\mathrm{e}^\epsilon)(1-2\alpha)(1+4\alpha)\mathrm{e}^\epsilon}{(\mathrm{e}^\epsilon - 1)^2}.$$

The complexities of $\Pi^{\mathrm{RR}'}_{k,\ell}$ are

$$\#\mathsf{Card}(\Pi^{\mathrm{RR}'}_{k,\ell}) = n + \ell = O\left(\frac{n\mathrm{e}^\epsilon}{\mathrm{e}^\epsilon - 1}\right) \text{ and } \#\mathsf{Shuffle}(\Pi^{\mathrm{RR}'}_{k,\ell}) = 1.$$

Proof. To begin with, it holds that

$$\frac{1+2\alpha\mathrm{e}^\epsilon}{\mathrm{e}^\epsilon+1} - \frac{1+\alpha\mathrm{e}^\epsilon}{\mathrm{e}^\epsilon+1} \geq \frac{1}{\ell}. \tag{9}$$

Indeed, since $\alpha = n/\ell$, the inequality (9) is equivalent to $n \geq (\mathrm{e}^\epsilon + 1)/\mathrm{e}^\epsilon$. Since $n \geq 2$ and $2\mathrm{e}^\epsilon > \mathrm{e}^\epsilon + 1$, (9) actually holds. Thus, there exists an integer k satisfying the condition (8).

To see the differential privacy of the protocol $\Pi^{\mathrm{RR}'}_{k,\ell}$, let T be a set of corrupted parties such that $|T| \leq n - 1$, and let $\boldsymbol{x} = (x_i)_{i\in[n]}, \boldsymbol{x}' = (x'_i)_{i\in[n]} \in \{0,1\}^n$ be T-neighboring inputs.

Let $H = [n] \setminus T$. Then $x_i \neq x'_i$ for some $i \in H$. We assume that $x_i = 0, x'_i = 1$. The case of $x_i = 1, x'_i = 0$ can be dealt with similarly. Let $H_i = H \setminus \{i\}$. For $j \in [n]$, define R_j as a random variable corresponding to the state $r_j \in \{♣, ♡\}$ of the card that the j-th party draws from supplementary cards. For $j \in [n]$, define $Y_j(\boldsymbol{x})$ as a random variable corresponding to the state of the card that the j-th party submits when parties' inputs are $\boldsymbol{x}$. Similarly, we define $Y_j(\boldsymbol{x}')$ as a corresponding random variable when parties' inputs are $\boldsymbol{x}'$. For a subset $S \subseteq [n]$, we denote $R_S = (R_j)_{j\in S}, Y_S(\boldsymbol{x}) = (Y_j(\boldsymbol{x}))_{j\in S}, Y_S(\boldsymbol{x}') = (Y_j(\boldsymbol{x}'))_{j\in S}$.

Then, the joint view of corrupted parties in T is given as $\mathsf{View}_{\Pi^{\mathrm{RR}}_{k,\ell},T}(\boldsymbol{x}) = (Y_H(\boldsymbol{x}), R_T)$ and $\mathsf{View}_{\Pi^{\mathrm{RR}}_{k,\ell},T}(\boldsymbol{x}') = (Y_H(\boldsymbol{x}'), R_T)$. For any outcome (y_H, r_T) of $(Y_H(\boldsymbol{x}), R_T)$, it holds that

$$\begin{aligned}
&\Pr\Big[\mathsf{View}_{\Pi^{\mathrm{RR}}_{k,\ell},T}(\boldsymbol{x}) = (y_H, r_T)\Big] \\
&= \Pr[R_T = r_T]\Pr[Y_H(\boldsymbol{x}) = y_H \mid R_T = r_T] \\
&= \Pr[R_T = r_T] \cdot \sum_{r_{H_i}} \Pr[R_{H_i} = r_{H_i}]\Pr[Y_H(\boldsymbol{x}) = y_H \mid R_T = r_T, R_{H_i} = r_{H_i}],
\end{aligned}$$

where r_{H_i} ranges over the set of all outcomes of R_{H_i}. Since $Y_{H_i}(\boldsymbol{x})$ is uniquely determined by R_{H_i}, we have that

$$\begin{aligned}
&\Pr\Big[\mathsf{View}_{\Pi^{\mathrm{RR}}_{k,\ell},T}(\boldsymbol{x}) = (y_H, r_T)\Big] \\
&= \Pr[R_T = r_T] \cdot \sum_{r_{H_i}} \Pr[R_{H_i} = r_{H_i}]\Pr\big[Y_i(\boldsymbol{x}) = y_i \bigm| R_{[n]\setminus\{i\}} = r_{[n]\setminus\{i\}}\big].
\end{aligned}$$

Let

$$\begin{aligned}
&P(y_i) = \Pr\big[Y_i(\boldsymbol{x}) = y_i \bigm| R_{[n]\setminus\{i\}} = r_{[n]\setminus\{i\}}\big] \text{ and} \\
&P'(y_i) = \Pr\big[Y_i(\boldsymbol{x}') = y_i \bigm| R_{[n]\setminus\{i\}} = r_{[n]\setminus\{i\}}\big].
\end{aligned}$$

Suppose that $r_{[n]\setminus\{i\}}$ is composed of $n-1-j$ ♣s and j ♡s. Since we assume that $x_i = 0 = ♣$ and $x'_i = 1 = ♡$, $P(♣)$ and $P'(♡)$ are equal to the probability of the event that the i-th party draws $r_i = ♣$ from supplementary cards, and hence we obtain that

$$P(♣) = P'(♡) = \frac{\ell - n + 1 - k + j}{\ell - n + 1}.$$

In addition, $P(♡)$ and $P'(♣)$ are equal to the probability of the event that the i-th party draws $r_i = ♡$ from supplementary cards, and hence we have that

$$P(♡) = P'(♣) = \frac{k - j}{\ell - n + 1}.$$

Therefore, it holds that

$$\frac{P(y_i)}{P'(y_i)} \leq \max\left\{\frac{k-j}{\ell - n + 1 - k + j}, \frac{\ell - n + 1 - k + j}{k - j}\right\}.$$

Since $0 \leq j \leq n-1$, we obtain that

$$\begin{aligned}
\frac{P(y_i)}{P'(y_i)} &\leq \max\left\{\frac{k}{\ell - n + 1 - k}, \frac{\ell - k}{k - n + 1}\right\} \\
&= \max\left\{\frac{p}{(1-p) - (n-1)/\ell}, \frac{1-p}{p - (n-1)/\ell}\right\} \\
&\leq \max\left\{\frac{p}{(1-p) - \alpha}, \frac{1-p}{p - \alpha}\right\}.
\end{aligned}$$

On the other hand, we have that

$$\max\left\{\frac{p}{(1-p)-\alpha}, \frac{1-p}{p-\alpha}\right\} \leq \mathrm{e}^{\epsilon} \iff \frac{\alpha\mathrm{e}^{\epsilon}+1}{\mathrm{e}^{\epsilon}+1} \leq p \leq \frac{(1-\alpha)\mathrm{e}^{\epsilon}}{\mathrm{e}^{\epsilon}+1}$$

and

$$\frac{(1-\alpha)\mathrm{e}^{\epsilon}}{\mathrm{e}^{\epsilon}+1} \geq \frac{1+2\alpha\mathrm{e}^{\epsilon}}{\mathrm{e}^{\epsilon}+1} \iff \frac{n}{\ell} \leq \frac{\mathrm{e}^{\epsilon}-1}{3\mathrm{e}^{\epsilon}}.$$

Thus, it follows from the condition (8) that $P(y_i) \leq \mathrm{e}^{\epsilon} \cdot P'(y_i)$. We therefore conclude that

$$\begin{aligned}
&\Pr\left[\mathsf{View}_{\Pi^{\mathrm{RR}}_{k,\ell},T}(\boldsymbol{x}) = (y_H, r_T)\right] \\
&= \Pr[R_T = r_T]\sum_{r_{H_i}} \Pr[R_{H_i} = r_{H_i}]P(y_i) \\
&\leq \Pr[R_T = r_T]\sum_{r_{H_i}} \Pr[R_{H_i} = r_{H_i}]\mathrm{e}^{\epsilon}P'(y_i) \\
&= \mathrm{e}^{\epsilon}\Pr[R_T = r_T] \cdot \sum_{r_{H_i}} \Pr[R_{H_i} = r_{H_i}]\Pr\left[Y_i(\boldsymbol{x}') = y_i \;\middle|\; R_{[n]\setminus\{i\}} = r_{[n]\setminus\{i\}}\right] \\
&= \mathrm{e}^{\epsilon}\Pr\left[\mathsf{View}_{\Pi^{\mathrm{RR}}_{k,\ell},T}(\boldsymbol{x}') = (y_H, r_T)\right].
\end{aligned}$$

To analyze the utility of the protocol, let $\boldsymbol{x} \in \{0,1\}^n$. For ease of notations, we write $Y_i = Y_i(\boldsymbol{x}), s = \sum_{i\in[n]} x_i$. Note that $Y_i = 1$ if and only if the i-th party submits ♡ to main cards, and $Y_i = 0$ if and only if the i-th party submits ♣. Furthermore, $\sum_{i\in[n]} Y_i$ is equal to the total number y of ♡s included in main cards.

The expectations of Y_i and Y_i^2 are

$$\mathbb{E}[Y_i] = 1 \cdot \Pr[Y_i = 1] = (1-2p)x_i + p \text{ and } \mathbb{E}\left[Y_i^2\right] = 1^2 \cdot \Pr[Y_i = 1] = (1-2p)x_i + p.$$

In particular, the expectation of an output z of $\Pi^{\mathrm{RR}'}_{k,\ell}$ is $s = f(\boldsymbol{x})$ and hence $\mathrm{MSE}_{\Pi^{\mathrm{RR}}_{k,\ell}}(f)$ is equal to the variance of z.

Since $x^2 = x$ if $x \in \{0,1\}$, the variance of Y_i is

$$\mathrm{Var}[Y_i] = \mathbb{E}\left[Y_i^2\right] - (\mathbb{E}[Y_i])^2 = p(1-p) + (1-2p)^2x_i - (1-2p)^2x_i^2 = p(1-p).$$

For any $i \neq j$, if $x_i = x_j = 0$, then

$$\Pr[Y_i = 1, Y_j = 1] = \frac{\binom{\ell-2}{k-2}}{\binom{\ell}{k}} = \frac{k(k-1)}{\ell(\ell-1)} =: a_1.$$

If $x_i = 1, x_j = 0$ or $x_i = 0, x_j = 1$, then

$$\Pr[Y_i = 1, Y_j = 1] = \frac{\binom{\ell-2}{k-1}}{\binom{\ell}{k}} = \frac{k(\ell-k)}{\ell(\ell-1)} =: a_2.$$

If $x_i = x_j = 1$, then

$$\Pr[Y_i = 1, Y_j = 1] = \frac{\binom{\ell-2}{k}}{\binom{\ell}{k}} = \frac{(\ell-k)(\ell-k-1)}{\ell(\ell-1)} =: a_3.$$

We have that

$$\mathbb{E}[Y_iY_j] = \Pr[Y_i = 1, Y_j = 1] = (1-x_i)(1-x_j)a_1 + ((1-x_i)x_j + x_i(1-x_j))a_2 + x_ix_ja_3.$$

Thus, the covariance of Y_i and Y_j is

$$\begin{aligned}\mathrm{Cov}[Y_i, Y_j] &= \mathbb{E}[Y_iY_j] - \mathbb{E}[Y_i]\,\mathbb{E}[Y_j]\\ &= (a_1 - p^2) + (-a_1 + a_2 - (1-2p)p)(x_i + x_j) + (a_1 - 2a_2 + a_3 - (1-2p)^2)x_ix_j\\ &= -\frac{k(\ell-k)}{\ell^2(\ell-1)} + \frac{2k(\ell-k)}{\ell^2(\ell-1)}(x_i+x_j) - \frac{4k(\ell-k)}{\ell^2(\ell-1)}x_ix_j\end{aligned}$$

We thus obtain that

$$\begin{aligned}\mathrm{Var}\left[\sum_{i\in[n]} Y_i\right] &= \sum_{i\in[n]} \mathrm{Var}[Y_i] + \sum_{i\neq j} \mathrm{Cov}[Y_i, Y_j]\\ &= p(1-p)n - \frac{k(\ell-k)}{\ell^2(\ell-1)}n(n-1) + \frac{4k(\ell-k)}{\ell^2(\ell-1)}(n-1)s - \frac{4k(\ell-k)}{\ell^2(\ell-1)}\sum_{i\neq j} x_ix_j\\ &\leq p(1-p)n + \frac{4p(1-p)}{\ell}n^2\\ &\leq \frac{(1+2\alpha \mathrm{e}^\epsilon)(\mathrm{e}^\epsilon - 2\alpha\mathrm{e}^\epsilon)}{(\mathrm{e}^\epsilon+1)^2}(1+4\alpha)n.\end{aligned}$$

On the other hand, the condition (8) implies that

$$1 - 2p \geq 1 - \frac{2(1+2\alpha\mathrm{e}^\epsilon)}{\mathrm{e}^\epsilon + 1} = \frac{\mathrm{e}^\epsilon - 1 - 4\alpha\mathrm{e}^\epsilon}{\mathrm{e}^\epsilon+1} \geq \frac{\mathrm{e}^\epsilon - 1}{5(\mathrm{e}^\epsilon+1)} > 0.$$

Thus, the variance of z is upper bounded by

$$\mathrm{Var}\left[\frac{\sum_{i\in[n]} Y_i - np}{1-2p}\right] = \frac{1}{(1-2p)^2}\mathrm{Var}\left[\sum_{i\in[n]} Y_i\right] \leq \frac{25n(1+2\alpha\mathrm{e}^\epsilon)(1-2\alpha)(1+4\alpha)\mathrm{e}^\epsilon}{(\mathrm{e}^\epsilon-1)^2}. ◀$$

6 Performance Evaluation

We evaluate our proposed protocols based on the following performance metrics:

- Number of cards: The total number of main cards and supplementary cards.
- Error: The mean squared error with respect to the binary sum $f(x_1,\ldots,x_n) = \sum_{i\in[n]} x_i$.
- Number of shuffles: The total number of shuffles in the protocol, including the preparation of supplementary cards.

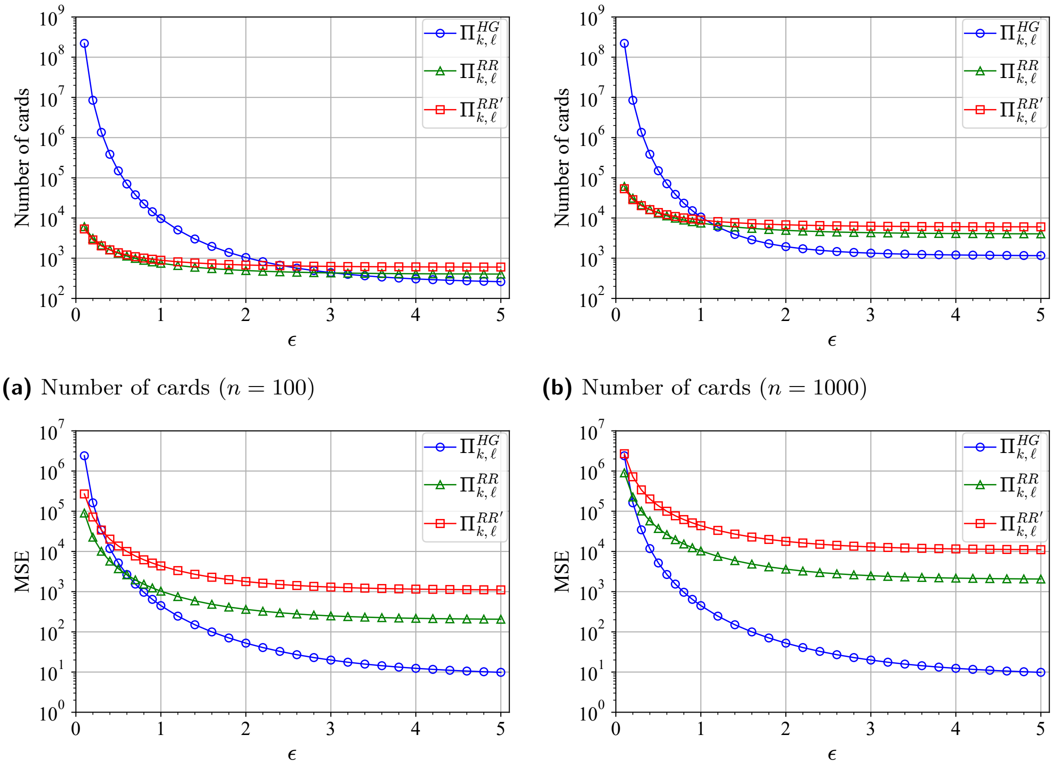
Table 1 in Section 1.1 shows the performance of our protocols in the asymptotic setting where $\epsilon \to 0$. Here, we use the approximation $\mathrm{e}^\epsilon \approx 1 + \epsilon + \epsilon^2/2$. We set k and ℓ to the minimum integers that satisfy the conditions in Theorems 4, 5, and 6.

Figure 4 shows the performance of our protocols for concrete values of n, ϵ, and δ. We set $n \in \{100, 1000\}$, $\epsilon \in \{0.1, 0.2, \ldots, 1, 1.2, \ldots, 5.0\}$, and $\delta = 10^{-6}$, and plot the number of cards and the MSE.

Below, we highlight the advantage of each of the protocols $\Pi^{\mathrm{HG}}_{k,\ell}$, $\Pi^{\mathrm{RR}}_{k,\ell}$, and $\Pi^{\mathrm{RR}'}_{k,\ell}$.

$\Pi^{\mathrm{HG}}_{k,\ell}$: The error and the number of shuffles do not depend on n. The additive overhead in the number of cards, i.e., $O(\epsilon^{-5}\ln\delta^{-1})$, is independent of n. In contrast, $\Pi^{\mathrm{RR}}_{k,\ell}$ and $\Pi^{\mathrm{RR}'}_{k,\ell}$ suffer from a larger number of cards and a larger error when n becomes larger, as shown in Figure 4.

$\Pi^{\mathrm{RR}}_{k,\ell}$: When ϵ is close to 0, $\Pi^{\mathrm{RR}}_{k,\ell}$ achieves a smaller number of cards and a smaller error than $\Pi^{\mathrm{HG}}_{k,\ell}$ and $\Pi^{\mathrm{RR}'}_{k,\ell}$, as shown in Figure 4. In addition, $\Pi^{\mathrm{RR}}_{k,\ell}$ achieves pure differential privacy (i.e., $\delta = 0$).



(a) Number of cards ($n = 100$)

(b) Number of cards ($n = 1000$)

(c) MSE ($n = 100$)

(d) MSE ($n = 1000$)

Figure 4 The number of cards and MSE of our protocols.

$\Pi^{\mathrm{RR}'}_{k,\ell}$: The number of shuffles is only one. $\Pi^{\mathrm{RR}'}_{k,\ell}$ achieves asymptotically the same upper bound on the number of cards and the error as $\Pi^{\mathrm{RR}}_{k,\ell}$. It also achieves pure differential privacy (i.e., $\delta = 0$).

7 Removing Private Operations

In the protocols $\Pi^{\mathrm{RR}}_{k,\ell}$ and $\Pi^{\mathrm{RR}'}_{k,\ell}$, parties need to perform private reveals and privately decide which cards to submit based on the results. While such private operations could be easily realized in practice, it has also been considered as theoretically important to construct protocols without any private operations in the literature (i.e., those following the traditional model of card-based protocols [16]). In this section, we show a variant $\tilde{\Pi}^{\mathrm{RR}}_{k,\ell}$ (resp. $\tilde{\Pi}^{\mathrm{RR}'}_{k,\ell}$) of $\Pi^{\mathrm{RR}}_{k,\ell}$ (resp. $\Pi^{\mathrm{RR}'}_{k,\ell}$) where parties do not perform private reveals at the cost of doubling the number of cards and requiring n more shuffles. Note that the protocol $\Pi^{\mathrm{HG}}_{k,\ell}$ can be described following the model of [16] as it assumes no private operation.

Our solution is to emulate private XOR operations done by each party with an existing XOR protocol without private reveals [17]. To this end, we first modify a way of encoding bits: We encode 0 into a pair of cards $\boxed{\clubsuit}\boxed{\heartsuit}$ and 1 into $\boxed{\heartsuit}\boxed{\clubsuit}$, instead of encoding 0 into $\boxed{\clubsuit}$ and 1 into $\boxed{\heartsuit}$. To preserve the structure of encoding, we consider a pair of cards encoding a bit as a minimum unit. In particular, we replace every complete shuffle with a pile-scramble shuffle. That is, whenever we shuffle m cards in $\Pi^{\mathrm{RR}}_{k,\ell}$ and $\Pi^{\mathrm{RR}'}_{k,\ell}$, we shuffle m pairs of cards in such a way that the pairs are uniformly permuted but the order of cards in each pair is preserved.

Next, the XOR protocol in [17] allows parties to perform the following conversion of cards:

$$\underbrace{\boxed{?}\boxed{?}}_{a}\underbrace{\boxed{?}\boxed{?}}_{b} \rightarrow \underbrace{\boxed{?}\boxed{?}}_{a \oplus b}.$$

In the above protocol, parties do not perform any private operation, and the trace of states of cards is independent of inputs a, b or an output $a \oplus b$.[2] We modify $\Pi^{\mathrm{RR}}_{k,\ell}$ and $\Pi^{\mathrm{RR}'}_{k,\ell}$ as follows: Whenever a party randomizes his input bit, he first submits a pair of face-down cards encoding x_i, picks a pair of face-down cards encoding a random bit r_i, and then computes their XOR with the protocol in [17]. Since the XOR protocol in [17] requires no additional card and only one pile-scramble shuffle, the cost for executing n instances of the XOR protocol is n pile-scramble shuffles.

Finally, observe that the final states of odd-numbered cards in main cards in $\tilde{\Pi}^{\mathrm{RR}}_{k,\ell}$ and $\tilde{\Pi}^{\mathrm{RR}'}_{k,\ell}$ is equal to the final states of main cards in the original protocols $\Pi^{\mathrm{RR}}_{k,\ell}$ and $\Pi^{\mathrm{RR}'}_{k,\ell}$, respectively. We thus calculate the number y of ♡s in the odd-numbered cards and output $z = \frac{y - nk/\ell}{1-2k/\ell}$.

The security of the XOR protocol ensures that the trace of states visible to parties is simulated from that of $\Pi^{\mathrm{RR}}_{k,\ell}$ or $\Pi^{\mathrm{RR}'}_{k,\ell}$. Hence, the resultant protocols $\tilde{\Pi}^{\mathrm{RR}}_{k,\ell}$ and $\tilde{\Pi}^{\mathrm{RR}'}_{k,\ell}$ achieve the same level of differential privacy and MSE as the original protocols. On the other hand, due to the structure of encoding and the additional shuffles to execute the XOR protocol, the complexities of $\tilde{\Pi}^{\mathrm{RR}}_{k,\ell}$ and $\tilde{\Pi}^{\mathrm{RR}'}_{k,\ell}$ are given as follows:

$$\#\mathsf{Card}(\tilde{\Pi}^{\mathrm{RR}}_{k,\ell}) = 2n(\ell+1) = O\left(\frac{n\mathrm{e}^{\epsilon}}{\mathrm{e}^{\epsilon}-1}\right),\ \#\mathsf{Shuffle}(\tilde{\Pi}^{\mathrm{RR}}_{k,\ell}) = 2n,$$
$$\#\mathsf{Card}(\tilde{\Pi}^{\mathrm{RR}'}_{k,\ell}) = 2(n+\ell) = O\left(\frac{n\mathrm{e}^{\epsilon}}{\mathrm{e}^{\epsilon}-1}\right),\ \text{and } \#\mathsf{Shuffle}(\tilde{\Pi}^{\mathrm{RR}'}_{k,\ell}) = n+1.$$

References

1 Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. cpSGD: Communication-efficient and differentially-private distributed SGD. In *Advances in Neural Information Processing Systems*, pages 7564–7575, 2018.

2 Miguel E. Andrés, Nicolás E. Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 20th ACM Conference on Computer and Communications Security (CCS'13)*, pages 901–914, 2013.

3 Amos Beimel, Kobbi Nissim, and Eran Omri. Distributed private data analysis: Simultaneously solving how and what. In *Advances in Cryptology – CRYPTO 2008*, pages 451–468, 2008.

4 Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *Advances in Cryptology – EUROCRYPT 2019*, pages 375–403, 2019.

5 Václav Chvátal. The tail of the hypergeometric distribution. *Discrete Mathematics*, 25(3):285–287, 1979.

6 Claude Crépeau and Joe Kilian. Discreet solitary games. In *Advances in Cryptology – CRYPTO' 93*, pages 319–330, 1994.

[2] We show a self-contained exposition of the XOR protocol in [17] in Appendix A.

7 Bert Den Boer. More efficient match-making and satisfiability the five card trick. In *Advances in Cryptology – EUROCRYPT' 89*, pages 208–217, 1990.

8 Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology – EUROCRYPT 2006*, pages 486–503, 2006.

9 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, pages 265–284, 2006.

10 Cynthia Dwork and Aaron Roth. *The Algorithmic Foundations of Differential Privacy*. Now Publishers, 2014.

11 Ronen Gradwohl, Moni Naor, Benny Pinkas, and Guy N. Rothblum. Cryptographic and physical zero-knowledge proof systems for solutions of Sudoku puzzles. *Theory of Computing Systems*, 44(2):245–268, 2009.

12 Rie Ishikawa, Eikoh Chida, and Takaaki Mizuki. Efficient card-based protocols for generating a hidden random permutation without fixed points. In *Unconventional Computation and Natural Computation*, pages 215–226, 2015.

13 Yoshifumi Manabe and Hibiki Ono. Secure card-based cryptographic protocols using private operations against malicious players. In *Innovative Security Solutions for Information Technology and Communications*, pages 55–70, 2021.

14 Ilya Mironov. On significance of the least significant bits for differential privacy. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, pages 650–661, 2012.

15 Daiki Miyahara, Itaru Ueda, Yu-ichi Hayashi, Takaaki Mizuki, and Hideaki Sone. Analyzing execution time of card-based protocols. In *Unconventional Computation and Natural Computation*, pages 145–158, 2018.

16 Takaaki Mizuki and Hiroki Shizuya. A formalization of card-based cryptographic protocols via abstract machine. *International Journal of Information Security*, 13(1):15–23, 2014.

17 Takaaki Mizuki and Hideaki Sone. Six-card secure AND and four-card secure XOR. In *Frontiers in Algorithmics*, volume 5598, pages 358–369, 2009.

18 Valtteri Niemi and Ari Renvall. Solitaire zero-knowledge. *Fundamenta Informaticae*, 38(1,2):181–188, 1999.

19 Takuya Nishida, Yu-ichi Hayashi, Takaaki Mizuki, and Hideaki Sone. Card-based protocols for any Boolean function. In *Theory and Applications of Models of Computation*, pages 110–121, 2015.

20 Hibiki Ono and Yoshifumi Manabe. Efficient card-based cryptographic protocols for the Millionaires' problem using private input operations. In *Asia Joint Conference on Information Security (AsiaJCIS)*, pages 23–28, 2018.

21 Sofya Raskhodnikova and Adam Smith. *Differentially Private Analysis of Graphs*, pages 543–547. Springer, 2016.

22 Kazumasa Shinagawa and Koji Nuida. A single shuffle is enough for secure card-based computation of any Boolean circuit. *Discrete Applied Mathematics*, 289:248–261, 2021.

23 Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. Locally differentially private protocols for frequency estimation. In *Proceedings of the 26th USENIX Security Symposium (USENIX'17)*, pages 729–745, 2017.

24 Stanley L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

25 Yohei Watanabe, Yoshihisa Kuroki, Shinnosuke Suzuki, Yuta Koga, Mitsugu Iwamoto, and Kazuo Ohta. Card-based majority voting protocols with three inputs using three cards. In *2018 International Symposium on Information Theory and Its Applications (ISITA)*, pages 218–222, 2018.

26 Tianqing Zhu, Gang Li, Wanlei Zhou, and Philip S. Yu. *Differential Privacy and Applications*. Springer, 2017.

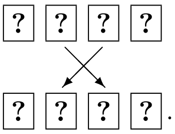
A The XOR Protocol in [17]

Mizuki and Sone [17] proposed the following four-card XOR protocol, which takes commitments to a, b with the two-card encoding $\boxed{\clubsuit}\boxed{\heartsuit} = 0$, $\boxed{\heartsuit}\boxed{\clubsuit} = 1$ and outputs a commitment to $a \oplus b$ without additional cards:

1. Arrange the input commitments as follows:

$$\underbrace{\boxed{?}\boxed{?}}_{a}\underbrace{\boxed{?}\boxed{?}}_{b}.$$

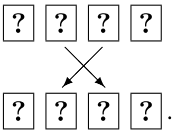
2. Rearrange the order of the sequence as follows:



3. Apply a pile-scramble shuffle with two piles (also known as a *random bisection cut*):

$$\left[\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\,\middle|\,\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\right] \rightarrow \left\{\begin{array}{l} \overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\;\overset{3}{\boxed{?}}\overset{4}{\boxed{?}} \\ \overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\;\overset{1}{\boxed{?}}\overset{2}{\boxed{?}} \end{array}\right. .$$

4. Rearrange the order of the sequence as follows:



5. Reveal the leftmost two cards and determine the output commitment as follows:

$$\boxed{\clubsuit}\boxed{\heartsuit}\underbrace{\boxed{?}\boxed{?}}_{a\oplus b} \text{ or } \boxed{\heartsuit}\boxed{\clubsuit}\underbrace{\boxed{?}\boxed{?}}_{\overline{a\oplus b}}.$$