A Blockchain-Based Approach for Saving and Tracking Differential-Privacy Cost

Yang Zhao, *Graduate Student Member, IEEE*, Jun Zhao, *Member, IEEE*, Jiawen Kang, Zehang Zhang, Dusit Niyato, *Fellow, IEEE*, Shuyu Shi, *Member, IEEE*, and Kwok-Yan Lam, *Senior Member, IEEE*

***Abstract*—An increasing amount of users' sensitive information is now being collected for analytics purposes. Differential privacy has been widely studied in the literature to protect the privacy of users' information. The privacy parameter bounds the information about the data set leaked by the noisy output. Oftentimes, a data set needs to be used for answering multiple queries, so the level of privacy protection may degrade as more queries are answered. Thus, it is crucial to keep track of privacy budget spending, which should not exceed the given limit of privacy budget. Moreover, if a query has been answered before and is asked again on the same data set, we may reuse the previous noisy response for the current query to save the privacy cost. In view of the above, we design an algorithm to reuse previous noisy responses if the same query is asked repeatedly. In particular, considering that different requests of the same query may have different privacy requirements, our algorithm can set the optimal reuse fraction of the old noisy response and add new noise to minimize the accumulated privacy cost. Furthermore, we design and implement a blockchain-based system for tracking and saving differential-privacy cost. As a result, the owner of the data set will have full knowledge about how the data set has been used and be confident that no new privacy cost will be incurred for answering queries once the specified privacy budget is exhausted.**

***Index Terms*—Blockchain, data analytics, differential privacy (DP), Gaussian mechanism.**

Manuscript received July 19, 2020; revised October 21, 2020 and December 22, 2020; accepted January 9, 2021. Date of publication February 9, 2021; date of current version May 21, 2021. The work of Yang Zhao and Jun Zhao was supported in part by the Nanyang Technological University (NTU) Startup Grant; in part by the Alibaba–NTU Singapore Joint Research Institute (JRI); in part by the Singapore Ministry of Education Academic Research Fund Tier 1 RG128/18, Tier 1 RG115/19, Tier 1 RT07/19, Tier 1 RT01/19, and Tier 2 MOE2019-T2-1-176; in part by NTU–WASP Joint Project; in part by the Singapore National Research Foundation (NRF) under its Strategic Capability Research Centres Funding Initiative: Strategic Centre for Research in Privacy-Preserving Technologies and Systems (SCRIPTS); in part by the Energy Research Institute @NTU (ERIAN); in part by the Singapore NRF National Satellite of Excellence, Design Science and Technology for Secure Critical Infrastructure NSoE DeST-SCI2019-0012; in part by AI Singapore (AISG) 100 Experiments (100E) Programme; and in part by the NTU Project for Large Vertical Take-Off and Landing (VTOL) Research Platform. The work of Dusit Niyato was supported in part by the National Research Foundation (NRF), Singapore, through Singapore Energy Market Authority (EMA), Energy Resilience under Grant NRF2017EWT-EP003-041, Singapore NRF under Grant NRF2015-NRF-ISF001-2277, Singapore NRF National Satellite of Excellence, Design Science and Technology for Secure Critical Infrastructure under Grant NSoE DeST-SCI2019-0007; in part by the A*STAR-NTU-SUTD Joint Research Grant on Artificial Intelligence for the Future of Manufacturing under Grant RGANS1906, Wallenberg AI, Autonomous Systems and Software Program and Nanyang Technological University (WASP/NTU) under Grant M4082187 (4080); in part by the Singapore Ministry of Education (MOE) Tier 1 (RG16/20); and in part by the Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba–NTU Singapore Joint Research Institute (JRI). The work of Kwok-Yan Lam was supported by the National Research Foundation, Singapore, under its Strategic Capability Research Centres Funding Initiative. *(Corresponding author: Jun Zhao.)*

Yang Zhao, Jun Zhao, Jiawen Kang, Zehang Zhang, Dusit Niyato, and Kwok-Yan Lam are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: s180049@e.ntu.edu.sg; junzhao@ntu.edu.sg; kavinkang@ntu.edu.sg; prestonzzh@163.com; dniyato@ntu.edu.sg; kwokyan.lam@ntu.edu.sg).

Shuyu Shi is with the Department of Computer Science and Engineering, Nanjing University, Nanjing 210008, China (e-mail: ssy@nju.edu.cn).

Digital Object Identifier 10.1109/JIOT.2021.3058209

I. Introduction

MASSIVE volumes of users' sensitive information are being collected for data analytics and machine learning such as large-scale Internet of Things (IoT) data. Some IoT data contain users' confidential information, for example, energy consumption or location data. They may expose a family's habits [1]–[5]. To protect personal privacy, many countries have strict policies about how technology companies collect and process users' data. However, the companies need to analyze users' data for service quality improvement. To preserve privacy while revealing useful information about data sets, differential privacy (DP) has been proposed [6]–[8]. Intuitively, by incorporating some noise, the output of an algorithm under DP will not change significantly due to the presence or absence of one user's information in the data set. Due to its introduction [6], [7], DP has attracted much interest from both academia [9]–[13] and industry [14]–[16]. For example, Apple has incorporated DP into its mobile operating system iOS [14]; Google has implemented a DP tool called RAPPOR in the Chrome browser to collect information [15].

Roughly speaking, a randomized mechanism achieving (ϵ, δ)-DP [6] means that except with a (typically small) probability δ, altering a record in a database cannot change the probability that an output is seen by more than a multiplicative factor e^{ϵ}. Thus, the information about the data set leaked by the noisy output of an (ϵ, δ)-DP algorithm is bounded by the privacy parameters ϵ and δ. Smaller ϵ and δ mean stronger privacy protection and less information leakage. Note that nonzero information leakage is necessary to achieve nonzero utility. Usually, a data set may be used for answering multiple queries (e.g., for multiple analytics tasks), thus accumulating the information leakage and degrading the privacy protection level, which can be intuitively understood as the increase of privacy spending. Therefore, it is necessary to record the privacy cost to prevent it from exceeding the privacy budget. The privacy budget is used to quantify the privacy risk when a differential private scheme is applied to real-world applications. Besides, we reduce privacy cost by reusing old noisy response to answer the current query if the query was answered before.

Traditionally, the privacy cost incurred by answering queries on a data set is claimed by the data set holder. Users whose information is in the data set are not clear about the usage. It is possible that privacy consumption has exceeded the privacy budget. To solve this problem, the emerging blockchain technology provides a new solution to manage the privacy cost. Blockchain is a chain of blocks storing cryptographic and tamper-resistant transaction records without using a centralized server [17], [18]. With blockchain recording how the data set is used for answering queries, users have full knowledge of how their information is analyzed. Users can easily access the blockchain to check the consumption of the privacy budget. The data set holder has the motivation to adopt our blockchain-based approach to provide the following accountability guarantee to users whose information is in the data set: if the data set holder uses the data set more than the set of queries recorded by the blockchain, measures can be taken to catch the data set holder with cheating because transactions written into the blockchain are tamper-resistant. Yang *et al.* [19] proposed to leverage blockchain to track DP budget, but they do not propose a mechanism to reuse noise. In contrast, we design a DP mechanism to effectively reuse previous queries' results to reuse noise and reduce privacy cost. In Section VI-A, we present more detailed comparisons.

In view of the above, we propose a blockchain-based Algorithm 1 and implement it to track and manage differential-privacy cost, which uses blockchain to make the privacy spending transparent to the data owner. Consequently, the data owner can track how data set used by checking blockchain transactions' information, including each query's type, the noisy response used to answer each query, the associated noise level added to the true query result, and the remaining privacy budget. In addition to providing transparency of privacy management, another advantage of our blockchain-based system is as follows. Once the specified privacy budget is exhausted, a smart contract implemented on the blockchain ensures that no new privacy cost will be incurred, and this can be verified. Furthermore, since the blockchain stores the noisy response used to answer each query, we also design an algorithm to minimize the accumulated privacy cost by reusing previous noisy response if the same query is asked again. Our algorithm (via a rigorous proof) is able to set the optimal reuse fraction of the old noisy response and add new noise (if necessary) considering different requests of the same query may be sent with different privacy requirements. In our blockchain-based system, reusing noisy responses not only saves privacy cost, but also reduces communication overhead when the noisy response is generated without contacting the server hosting the data set.

Contributions: The major contributions of this article are summarized as follows.

1) A novel privacy-preserving algorithm with a rigorous mathematical proof is designed to minimize accumulated privacy cost under a limited privacy budget by reusing previous noisy responses if the same query is received. Thus, a data set can be used to answer more queries while preventing the privacy leakage, which is essential for the data sets with frequent queries, e.g., medical record data sets.
2) Our designed approach reduces the number of times to request the server significantly by taking advantage of recorded noisy results.
3) We implement the proposed system and algorithm according to a detailed sequence diagram, and conduct experiments by using a real-world data set. Numerical results demonstrate that our proposed system and algorithm are effective in saving the privacy cost while keeping accuracy.

TABLE I
SUMMARY OF NOTATIONS

(ϵ, δ)	privacy parameters
$\mathbb{P}[\cdot]$	probability
$\mathbb{F}[\cdot]$	probability density function
D	dataset
D'	neighbouring dataset of D
Δ_Q	ℓ_2-sensitivity of query Q
σ	standard deviation of the Gaussian noise
$\widetilde{Q}_m(D)$	noisy query response for query Q_m on dataset D
$Y_1, Y_2, \ldots, Y_m$	randomized mechanisms
r_{optimal}	the optimal fraction
$L_Y(D, D'; y)$	privacy loss
$\mathcal{N}(0, A)$	a Gaussian random variable with zero mean and variance A
V	variance

Organization: The remainder of the article is organized as follows. Section II introduces preliminaries about DP and blockchains. Section III presents system design including our proposed noise reuse algorithm. Section IV describes challenges in implementing our system. In Section V, we discuss experimental results to validate the effectiveness of our system. Section VI surveys related work. Section VII discusses assumptions and limitations of our proposed scheme. Besides, we identify some future directions. Section VIII concludes this article.

Notation: Throughout this article, $\mathbb{P}[\cdot]$ denotes the probability, and $\mathbb{F}[\cdot]$ stands for the probability density function. The notation $\mathcal{N}(0, A)$ denotes a Gaussian random variable with zero mean and variance A, and means a fresh Gaussian noise when it is used to generate a noisy query response. Notations used in the rest of the paper are summarized in Table I.

II. Preliminaries

We organize this section on preliminaries as follows. In Section II-A, we introduce the formal definition of DP. In Section II-B, we explain the concepts of blockchain, Ethereum, and smart contract.

A. Differential Privacy

DP intuitively means that the adversary cannot determine with high confidence whether the randomized output comes

Algorithm 1 Our Proposed Algorithm to Answer the *m*th Query and Adjust Remaining Privacy Cost

```
Input: $D$: dataset; $Q_m$: the $m$-th query; $(\epsilon_m, \delta_m)$: requested privacy parameters for query $Q_m$; $(\sqrt{\epsilon\_squared\_remaining\_budget}, \delta_{\text{budget}})$: remaining privacy budget (at the beginning, it is $(\sqrt{\epsilon\_squared\_budget}, \delta_{\text{budget}})$ for $\epsilon\_squared\_budget = {\epsilon_{\text{budget}}}^2$); $\Delta_{Q_m}$: $\ell_2$ sensitivity of query $Q_m$;
Output: $\widetilde{Q}_m(D)$: noisy query response for query $Q_m$ on dataset $D$ under $(\epsilon_m, \delta_m)$-DP;
$\sigma_m \leftarrow \text{Gaussian}(\Delta_{Q_m}, \epsilon_m, \delta_m)$; //Comment: From Lemma 1, it holds that $\text{Gaussian}(\Delta_{Q_m}, \epsilon_m, \delta_m) := \sqrt{2\ln\frac{1.25}{\delta_m}} \times \frac{\Delta_{Q_m}}{\epsilon_m}$.
if the query $Q_m$ is seen for the first time then
    Client computes $\epsilon\_squared\_cost$ such that $\text{Gaussian}(\Delta_{Q_m}, \sqrt{\epsilon\_squared\_cost}, \delta_{\text{budget}}) = \sigma_m$;
    //Comment: This means $\sqrt{2\ln\frac{1.25}{\delta_{budget}}} \times \frac{\Delta_{Q_m}}{\sqrt{\epsilon\_squared\_cost}} = \sigma_m$, where $\sigma_m$ as $\text{Gaussian}(\Delta_{Q_m}, \epsilon_m, \delta_m)$ is $\sqrt{2\ln\frac{1.25}{\delta_m}} \times \frac{\Delta_{Q_m}}{\epsilon_m}$.
    Client computes $\epsilon\_squared\_remaining\_budget \leftarrow \epsilon\_squared\_remaining\_budget - \epsilon\_squared\_cost$;
    if $\epsilon\_squared\_remaining\_budget \geq 0$ then
        return $\widetilde{Q}_m(D) \leftarrow Q_m(D) + \mathcal{N}(0,1) \times \sigma_m$; //Comment: We refer to this Case 1) in Section III-D. If $Q_m$ is multidimensional, independent Gaussian noise will be added to each dimension.
        Blockchain records $\langle Q_m$'s query type, $\epsilon_m, \delta_m, \sigma_m, \widetilde{Q}_m(D)\rangle$; //Comment: This information will be kept together with a cryptographic hash of the dataset D, which Blockchain stores so it knows which records are for the same dataset D.
    else
        return an error of insufficient privacy budget;
    end if
else
    Suppose $Q_m$ is a type $t$-query. Blockchain compares $\sigma_m$ with values in $\mathbf{\Sigma}_t := \{\sigma_j : \sigma_j$ has been recorded in Blockchain and $Q_j$ is a type $t$-query$\}$ (i.e., $\mathbf{\Sigma}_t$ consists of the corresponding noise amounts for previous instances of type $t$-query), resulting in the following subcases.
    if there exists $\sigma_j \in \mathbf{\Sigma}_t$ such that $\sigma_m = \sigma_j$ then
        Blockchain returns $\widetilde{Q}_m(D) \leftarrow \widetilde{Q}_j(D)$; //Comment: We refer to this Case 2A) in Section III-D.
    else if $\sigma_m < \min(\mathbf{\Sigma}_t)$ then
        //Comment: The case of partially reusing an old noise:
        Client computes $\epsilon\_squared\_cost$ such that $[\text{Gaussian}(\Delta_{Q_m}, \sqrt{\epsilon\_squared\_cost}, \delta_{\text{budget}})]^{-2} = {\sigma_m}^{-2} - [\min(\mathbf{\Sigma}_t)]^{-2}$;
        Client computes $\epsilon\_squared\_remaining\_budget \leftarrow \epsilon\_squared\_remaining\_budget - \epsilon\_squared\_cost$;
        if $\epsilon\_squared\_remaining\_budget \geq 0$ then
            Blockchain computes $\text{NoiseReuseRatio} \leftarrow \frac{{\sigma_m}^2}{[\min(\mathbf{\Sigma}_t)]^2}$ and $\text{AdditionalNoise} \leftarrow \mathcal{N}(0,1) \times \sqrt{{\sigma_m}^2 - \frac{{\sigma_m}^4}{[\min(\mathbf{\Sigma}_t)]^2}}$
            Blockchain contacts Server to compute
            $\widetilde{Q}_m(D) \leftarrow Q_m(D) + \text{NoiseReuseRatio} \times [\widetilde{Q}_{t,\min}(D) - Q_m(D)] + \text{AdditionalNoise}$, where $\widetilde{Q}_{t,\min}(D)$ denotes the noisy response (kept in Blockchain) corresponding to $\min(\mathbf{\Sigma}_t)$; //Comment: We refer to this Case 2B) in Section III-D.

            Blockchain records $\langle Q_m$'s query type, $\epsilon_m, \delta_m, \sigma_m, \widetilde{Q}_m(D)\rangle$;
        else
            return an error of insufficient privacy budget;
        end if
    else
        //Comment: The case of fully reusing an old noise:
        With $\sigma_\ell$ denoting the maximal possible value in $\mathbf{\Sigma}_t$ that is also smaller than $\sigma_m$, Blockchain reuses $\widetilde{Q}_\ell(D)$, which denotes the noisy response (kept in Blockchain) corresponding to $\sigma_\ell$;
        Blockchain computes $\widetilde{Q}_m(D) \leftarrow \widetilde{Q}_\ell(D) + \mathcal{N}(0,1) \times \sqrt{{\sigma_m}^2 - {\sigma_\ell}^2}$; //Comment: We refer to this Case 2C) in Section III-D.
        Blockchain records $\langle Q_m$'s query type, $\epsilon_m, \delta_m, \sigma_m, \widetilde{Q}_m(D)\rangle$;
    end if
end if
```

from a data set D or its neighboring data set D' which differs from D by one record. The formal definition of (ϵ, δ)-DP is given in Definition 1, and the notion of neighboring data sets is discussed in Remark 2.

Definition 1 ((ϵ, δ)-DP [8]): A randomized mechanism Y, which generates a randomized output given a data set as the input, achieves (ϵ, δ)*-DP* if

$$\begin{aligned}&\mathbb{P}[Y(D) \in \mathcal{Y}] \leq e^{\epsilon}\mathbb{P}\big[Y(D') \in \mathcal{Y}\big] + \delta\\&\text{for } D \text{ and } D' \text{ iterating through}\\&\text{all pairs of neighboring data sets, and}\\&\text{for } \mathcal{Y} \text{ iterating through all subsets of the output range}\end{aligned} \tag{1}$$

where $\mathbb{P}[\cdot]$ denotes the probability, and the probability space is over the coin flips of the randomized mechanism Y.

Remark 1: The notion of (ϵ, δ)-DP under $\delta = 0$ becomes ϵ-*DP*. ϵ-DP and (ϵ, δ)-DP are also referred to as *pure* and *approximate* DP, respectively, in many studies [9]–[11].

Remark 2 (Notion of Neighboring Data Sets): Two data sets D and D' are called neighboring if they differ only in one tuple. There are still variants about this. In the first case, the sizes of D and D' differ by one so that D' is obtained by adding one record to D or deleting one record from D. In the second case, D and D' have the same size (say n), and have different records at only one of the n positions. Finally, the notion of neighboring data sets can also be defined to include both the cases above. Our results in this article apply to all of the above cases.

Among various mechanisms to achieve DP, the *Gaussian mechanism* for real-valued queries proposed in [6] has received much attention. The improved result given by [8] is Lemma 1.

Lemma 1 (Theorem A.1 by Dwork and Roth [8]): To answer a query Q with ℓ_2-sensitivity Δ_Q, adding a zero-mean Gaussian noise with standard deviation $\sqrt{2\ln(1.25/\delta)} \times (\Delta_Q/\epsilon)$ (denoted by Gaussian$(\Delta_Q, \epsilon, \delta)$ hereafter in this article) to each dimension of the true query result achieves (ϵ, δ)-DP. The above ℓ_2-sensitivity Δ_Q of a query Q is defined as the maximal ℓ_2 distance between the true query results for any two neighboring data sets D and D' that differ in one record; i.e., $\Delta_Q = \max_{\text{neighboring } D, D'} \|Q(D) - Q(D')\|_2$.

More discussions on the ℓ_2-sensitivity of a query are given in Section III-H. Section VII-A discusses the setting of privacy parameters ϵ and δ.

B. Blockchain, Ethereum, and Smart Contracts

Blockchain: The blockchain technology is popularly used in systems requiring high security and transparency, such as Bitcoin and Ethereum [20]. The blockchain can be effectively used to solve the double-spending problem in Bitcoin transaction by using a peer-to-peer network. The solution is to hash transaction information in a chain of hash-based proof of work (PoW, used by Bitcoin) which is the consensus mechanism algorithm used to confirm transactions and produce new blocks to the chain. Once the record is formed, it cannot be changed except redoing PoW.

Besides, the blockchain is constantly growing with appending "completed" blocks. Blocks consisting of the most recent transactions are added to the chain in chronological order [21]. Each blockchain node can have a copy of the blockchain. The blockchain allows participants to track their transactions without centralized control.

Ethereum: Ethereum is a blockchain platform which allows users to create decentralized end-to-end applications [22]. The miners in Ethereum use PoW consensus algorithm to complete transaction verification and synchronization. Besides, Ethereum can run smart contracts elaborated below.

Smart Contract: The smart contract was first proposed by Nick Szabo as a computerized transaction protocol that can execute terms of a contract automatically [23]. It intends to make a contract digitally, and allows to maintain credible transactions without a third party. With the development of blockchains, such as Ethereum, smart contracts are stored in the blockchain as scripts. A blockchain with a Turing-complete programming language allows everyone to customize smart contract scripts for transactions [24]. Smart contracts are triggered when transactions are created or generated on the blockchain to finished specific tasks or services.

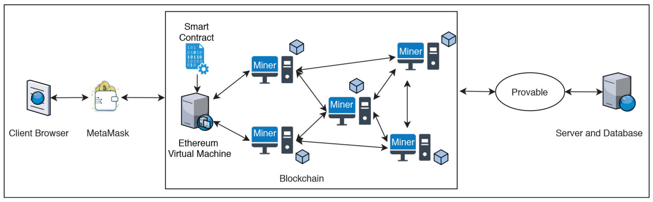


Fig. 1. Proposed blockchain-based system architecture for differential-privacy cost management.

III. System Description

Our blockchain-based system provides differentially private responses to queries while minimizing the privacy cost via noise reuse. We design a Web application to implement our Algorithm 1, which generates noisy responses to queries with the minimal privacy cost by setting the optimal reuse fraction of the old noisy response and adding new noise (if necessary). For clarity, we defer Algorithm 1 and its discussion to Section III. The design of the system is illustrated in Fig. 1 and we discuss the details in the following. In Section V, we will discuss the implementation and experiments of our blockchain-based system, and present more figures about the implementation. In particular, Fig. 3 there shows the screenshot of our blockchain-based privacy management system [25], while Fig. 4 presents outputs while using the system.

A. System Architecture

Our system includes the client, the blockchain, the server, and smart contract followed by more details as below.

Client: The primary function of the client is to transfer users' queries to the blockchain smart contract. The client computes the required parameter standard deviation for the server to generate the Gaussian noise using the privacy parameters ϵ and δ and forwards the query to the blockchain. Also, the client can display the query result to the analyst after getting the noisy response to the query.

Blockchain Smart Contract: The blockchain serves as a middleware between the client and the server. It decides which query should be submitted to the server. The blockchain records the remaining privacy budget, query type, the noisy response to answer the query, the privacy parameters, and the amount of corresponding noise. If the remaining privacy budget is enough, the smart contract will execute the query match function with the recorded history. Otherwise, the smart contract will reject this query. If the current query does not match with any query in the history, the smart contract will call the server to calculate the result. If the query has been received before, the blockchain smart contract will not call the server if

TABLE II
EXAMPLE TO EXPLAIN ALGORITHM 1

Q_m's query type	Q_1 =type-1	Q_2 =type-2	Q_3 =type-3	Q_4 =type-1	Q_5 =type-2	Q_6 =type-1	Q_7 =type-3	Q_8 =type-2	Q_9 =type-2	Q_{10} =type-1	Q_{11} =type-2	Q_{12} =type-1	Q_{13} =type-3
σ_m computed by Line 1 of Alg. 1	$\sigma_1 = 1$	$\sigma_2 = 3$	$\sigma_3 = 2$	$\sigma_4 = 2.5$	$\sigma_5 = 2$	$\sigma_6 = 0.5$	$\sigma_7 = 2$	$\sigma_8 = 2.5$	$\sigma_9 = 1.5$	$\sigma_{10} = 0.25$	$\sigma_{11} = 1$	$\sigma_{12} = 0.75$	$\sigma_{13} = 1.5$
Case involved in Alg. 1	1): $\widetilde{Q}_1 \leftarrow Q_1 + \mathcal{N}(0,1) \times \sigma_1$ with accessing D	1): $\widetilde{Q}_2 \leftarrow Q_2 + \mathcal{N}(0,1) \times \sigma_2$ with accessing D	1): $\widetilde{Q}_3 \leftarrow Q_3 + \mathcal{N}(0,1) \times \sigma_3$ with accessing D	2C): $\widetilde{Q}_4$ reuses $\widetilde{Q}_1$ without accessing D	2B): $\widetilde{Q}_5$ reuses $\widetilde{Q}_2$ with accessing D	2B): $\widetilde{Q}_6$ reuses $\widetilde{Q}_1$ with accessing D	2A): $\widetilde{Q}_7$ reuses $\widetilde{Q}_3$ without accessing D	2C): $\widetilde{Q}_8$ reuses $\widetilde{Q}_5$ without accessing D	2B): $\widetilde{Q}_9$ reuses $\widetilde{Q}_5$ with accessing D	2B): $\widetilde{Q}_{10}$ reuses $\widetilde{Q}_6$ with accessing D	2B): $\widetilde{Q}_{11}$ reuses $\widetilde{Q}_9$ with accessing D	2C): $\widetilde{Q}_{12}$ reuses $\widetilde{Q}_6$ without accessing D	2B): $\widetilde{Q}_{13}$ reuses $\widetilde{Q}_7$ with accessing D

the noisy response can be completely generated by old noisy answers and will call the server if access to the data set is still needed to generate the noisy response.

Server: The data provider hosts the server. The server provides APIs to answer analysts' queries. When the API is called, the server will query the data set to calculate the respective answer. After the true value $Q(D)$ is calculated, the server will add noise to perturb the answer. Then, the server returns the noisy answer to the blockchain.

The remainder of the article, we use `Blockchain`, `Client`, and `Server` to denote the blockchain, client, and server, respectively.

B. System Functionality

Match Query With Query History and Generate Noisy Response: `Blockchain` compares the current query type with saved query types to retrieve previous query results. If it is the first time for `Blockchain` to see the query, `Blockchain` will forward the query to the server, and `Server` will return the perturbed result which satisfies DP to `Blockchain`. If the current query type matches previous answers' query type, `Blockchain` will compare the computed amount of noise with all previously saved amounts of noise under the same query type. Based on the comparison result, `Blockchain` will completely reuse old responses or call `Server`.

Manage Privacy Budget: `Blockchain` updates the privacy budget as queries are answered and the Blockchain ensures no new privacy cost will be incurred for answering queries once the specified privacy budget is exhausted.

C. Adversary Model

The adversary model for our system is similar to [19]. Assume that there are two kinds of adversaries.

First, adversaries can obtain perturbed query results. They may try to infer users' real information using perturbed queries' results.

Second, adversaries attempt to modify the privacy budget. For example, they would like to decrease the used privacy budget so that users may exceed the privacy budget. As a result, privacy will leak. However, in our case, the privacy budget is recorded on the blockchain. The adversaries cannot tamper it once the privacy budget is stored in the blockchain.

D. Our Algorithm 1 Based on Reusing Noise

We present our solution for reusing noise in Algorithm 1 in Section III-E. We consider real-valued queries so that the Gaussian mechanism can be used. Extensions to nonreal-valued queries can be regarded as the future work, where we can apply the exponential mechanism of [12].

To clarify notation use, we note that Q_i means the ith query (ordered chronologically) and is answered by a randomized algorithm $\widetilde{Q}_i$. A type t-query means that the query's type is t. Queries asked at different time can have the same query type. This is the reason that we reuse noise in Algorithm 1.

Suppose a data set D has been used to answer $m-1$ queries $Q_1, Q_2, \ldots, Q_{m-1}$, where the ith query Q_i for $i = 1, 2, \ldots, m-1$ is answered under (ϵ_i, δ_i)-DP (by reusing noise, or generating fresh noise, or combining both). For $i = 1, 2, \ldots, m$, we define $\sigma_i := \text{Gaussian}(\Delta_{Q_i}, \epsilon_i, \delta_i)$, where Δ_{Q_i} denotes the ℓ_2-sensitivity of Q_i, where we defer the discussion of Δ_{Q_i} to Section III-H. As presented in Algorithm 1, we have several cases discussed below. For better understanding of these cases, we later discuss an example given in Table II in Section II.

Case 1): If Q_m is seen for the first time, we obtain the noisy response $\widetilde{Q}_m(D)$ by adding a zero-mean Gaussian noise with standard deviation $\text{Gaussian}(\Delta_{Q_m}, \epsilon_m, \delta_m)$ independently to each dimension of the true result $Q_m(D)$ (if the privacy budget allows), as given by Line 7 of Algorithm 1, where $\text{Gaussian}(\Delta_{Q_m}, \epsilon_m, \delta_m) := \sqrt{2\ln(1.25/\delta_m)} \times (\Delta_{Q_m}/\epsilon_m)$ from Lemma 1.

Case 2): If Q_m has been received before, suppose Q_m is a type t-query, and among the previous $m-1$ queries $Q_1, Q_2, \ldots, Q_{m-1}$, let $\mathbf{\Sigma}_t$ consist of the corresponding noise amounts for previous instances of type t-query; i.e., $\mathbf{\Sigma}_t := \{\sigma_j{:}\sigma_j$ has been recorded in `Blockchain` and Q_j is a type t-query$\}$. `Blockchain` compares σ_m and the values in $\mathbf{\Sigma}_t$, resulting in the following subcases.

Case 2A): If there exists $\sigma_j \in \mathbf{\Sigma}_t$ such that $\sigma_m = \sigma_j$, then $\widetilde{Q}_m(D)$ is set as $\widetilde{Q}_j(D)$.

Case 2B): This case considers that σ_m is less than $\min(\mathbf{\Sigma}_t)$ which denotes the minimum in $\mathbf{\Sigma}_t$. Let $\widetilde{Q}_{t,\min}(D)$ denote the noisy response (kept in `Blockchain`) corresponding to $\min(\mathbf{\Sigma}_t)$; specifically, if $\min(\mathbf{\Sigma}_t) = \sigma_j$ for some j, then $\widetilde{Q}_{t,\min}(D) = \widetilde{Q}_j(D)$. Under $\sigma_m < \min(\mathbf{\Sigma}_t)$, to minimize the privacy cost, we reuse $[{\sigma_m}^2/([\min(\mathbf{\Sigma}_t)]^2)]$ fraction of noise in $\widetilde{Q}_{t,\min}(D)$ to generate $\widetilde{Q}_m(D)$ (if the privacy budget allows). This will be obtained by Theorem 1's result 2) to be presented in Section III-E. Specifically, under $\min(\boldsymbol{\Sigma}_t) > \sigma_m$, as given by line 22 of Algorithm 1, $\widetilde{Q}_m(D)$ is set by $\widetilde{Q}_m(D) \leftarrow Q_m(D) + [\sigma_m{}^2/([\min(\boldsymbol{\Sigma}_t)]^2)] \times [\widetilde{Q}_{t,\min}(D) - Q_m(D)] + \mathcal{N}(0,1) \times \sqrt{\sigma_m{}^2 - [\sigma_m{}^4/([\min(\boldsymbol{\Sigma}_t)]^2)]}$. Note that if Q_m is multidimensional, independent Gaussian noise will be added to each dimension according to the above formula. This also applies to other places of this article.

Case 2C): This case considers that σ_m is greater than $\min(\boldsymbol{\Sigma}_t)$ and σ_m is different from all values in $\boldsymbol{\Sigma}_t$. Let σ_ℓ be the maximal possible value in $\boldsymbol{\Sigma}_t$ that is also smaller than σ_m; i.e., $\sigma_\ell = \max\{\sigma_j : \sigma_j \in \boldsymbol{\Sigma}_t \text{ and } \sigma_j < \sigma_m\}$. Then $\widetilde{Q}_m(D)$ is set as $\widetilde{Q}_\ell(D) + \mathcal{N}(0,1) \times \sqrt{\sigma_m{}^2 - \sigma_\ell{}^2}$. This will become clear by Theorem 1's result 2) to be presented in Section III-E.

An Example to Explain Algorithm 1: Table II provides an example for better understanding of Algorithm 1. We consider three types of queries. In particular, Q_1, Q_4, Q_6, Q_{10}, and Q_{12} are type 1-queries; Q_2, Q_5, Q_8, Q_9, and Q_{11} are type 2-queries, and Q_3, Q_7,and Q_{13} are type 3-queries.

E. Explaining the Noise Reuse Rules of Algorithm 1

Our noise-reuse rules of Algorithm 1 are designed to minimize the accumulated privacy cost. To explain this, inspired by [13], we define the privacy loss to quantify privacy cost. We analyze the privacy loss to characterize how privacy degrades in a fine-grained manner, instead of using the composition theorem by Kairouz *et al.* [26]. Although [26] gives the state-of-the-art results for the composition of differentially private algorithms, the results do not assume the underlying mechanisms to achieve DP. In our analysis, by analyzing the privacy loss of Gaussian mechanisms specifically, we can obtain smaller privacy cost.

For a randomized algorithm Y, neighboring data sets D and D', and output y, the privacy loss $L_Y(D, D'; y)$ represents the multiplicative difference between the probabilities that the same output y is observed when the randomized algorithm Y is applied to D and D'. Specifically, we define

$$L_Y(D, D'; y) := \ln \frac{\mathbb{F}\big[Y(D) = y\big]}{\mathbb{F}\big[Y(D') = y\big]}, \tag{2}$$

where $\mathbb{F}[\cdot]$ denotes the probability density function.

For simplicity, we use the probability density function $\mathbb{F}[\cdot]$ in (2) above by assuming that the randomized algorithm Y has the continuous output. If Y has the discrete output, we replace $\mathbb{F}[\cdot]$ by probability mass function $\mathbb{P}[\cdot]$.

When y follows the probability distribution of random variable $Y(D)$, $L_Y(D, D'; y)$ follows the probability distribution of random variable $L_Y(D, D'; Y(D))$, which we write as $L_Y(D, D')$ for simplicity.

We denote the composition of some randomized mechanisms $Y_1, Y_2, \ldots, Y_m$ for a positive integer m by $Y_1 \| Y_2 \| \ldots \| Y_m$. For the composition, the privacy loss with respect to neighboring data sets D and D' when the outputs of randomized mechanisms $Y_1, Y_2, \ldots, Y_m$ are $y_1, y_2, \ldots, y_m$ is defined by

$$L_{Y_1\|Y_2\|\ldots\|Y_m}\big(D, D'; y_1, y_2, \ldots, y_m\big) := \ln \frac{\mathbb{F}\big[\cap_{i=1}^m \big[Y_i(D) = y_i\big]\big]}{\mathbb{F}\big[\cap_{i=1}^m \big[Y_i(D') = y_i\big]\big]}.$$

When y_i follows the probability distribution of random variable $Y_i(D)$ for each $i \in \{1, 2, \ldots, m\}$, clearly $L_{Y_1\|Y_2\|\ldots\|Y_m}(D, D'; y_1, y_2, \ldots, y_m)$ follows the probability distribution of random variable $L_{Y_1\|Y_2\|\ldots\|Y_m}(D, D'; Y_1(D), Y_2(D), \ldots, Y_m(D))$, which we write as $L_{Y_1\|Y_2\|\ldots\|Y_m}(D, D')$ for simplicity.

With the privacy loss defined above, we now analyze how to reuse noise when a series of queries are answered under DP. To this end, we present Theorem 1, which presents the optimal ratio of reusing noise to minimize privacy cost.

Theorem 1 (Optimal Ratio of Reusing Noise to Minimize Privacy Cost): Suppose that before answering query Q_m and after answering $Q_1, Q_2, \ldots, Q_{m-1}$, the privacy loss $L_{\widetilde{Q}_1\|\widetilde{Q}_2\|\ldots\|\widetilde{Q}_{m-1}}(D, D')$ is given by $\mathcal{N}([A(D, D')/2], A(D, D'))$ for some $A(D, D')$. For the mth query Q_m, suppose that Q_m is the same as Q_j for some $j \in \{1, 2, \ldots, m-1\}$ and we reuse r fraction of noise in $\widetilde{Q}_j(D)$ to generate $\widetilde{Q}_m(D)$ for $0 \le r \le 1$ satisfying $\sigma_m{}^2 - r^2\sigma_j{}^2 > 0$, where r is a constant to be decided. If $\widetilde{Q}_j(D) - Q_j(D)$ follows a Gaussian probability distribution with mean 0 and standard deviation σ_j, we generate the noisy response $\widetilde{Q}_m(D)$ to answer query Q_m as follows:

$$\widetilde{Q}_m(D) \leftarrow Q_m(D) + r\big[\widetilde{Q}_j(D) - Q_j(D)\big] + \mathcal{N}\Big(0, \sigma_m{}^2 - r^2\sigma_j{}^2\Big) \tag{3}$$

so that $\widetilde{Q}_m(D) - Q_m(D)$ follows a Gaussian probability distribution with mean 0 and standard deviation σ_m.

Note that Δ_{Q_m} and Δ_{Q_j} are the same since Q_m and Q_j are the same. Then, we have the following results.

1) After answering the m queries $Q_1, Q_2, \ldots, Q_m$, the privacy loss $L_{\widetilde{Q}_1\|\widetilde{Q}_2\|\ldots\|\widetilde{Q}_m}(D, D')$ will be $\mathcal{N}([(B_r(D, D'))/2], B_r(D, D'))$ for $B_r(D, D') := A(D, D') + [([\|Q_m(D) - Q_m(D')\|_2]^2(1-r)^2)/(\sigma_m{}^2 - r^2\sigma_j{}^2)]$.
2) We clearly require $r \ge 0$ and $\sigma_m{}^2 - r^2\sigma_j{}^2 \ge 0$ in (3) above [note that $\mathcal{N}(0,0) \equiv 0$]. To minimize the total privacy cost [which is equivalent to minimize $B_r(D, D')$ above], the optimal r is given by

$$r_{\text{optimal}} = \begin{cases} 1, & \text{if } \sigma_m \ge \sigma_j, \\ \left(\frac{\sigma_m}{\sigma_j}\right)^2, & \text{if } \sigma_m < \sigma_j, \end{cases} \tag{4}$$

so that substituting (4) into the expression of $B_r(D, D')$ gives

$$B_{r_{\text{optimal}}}(D, D') = \begin{cases} A(D, D'), & \text{if } \sigma_m \ge \sigma_j \\ A(D, D') + \big[\big\|Q_m(D) - Q_m(D')\big\|_2\big]^2\Big(\frac{1}{\sigma_m{}^2} - \frac{1}{\sigma_j{}^2}\Big) & \\ & \text{if } \sigma_m < \sigma_j. \end{cases} \tag{5}$$

Note that if $\sigma_m = \sigma_j$ for some $j \in \{1, 2, \ldots, m-1\}$, we have $r_{\text{optimal}} = 1$ and just set $\widetilde{Q}_m(D)$ as $\widetilde{Q}_j(D)$.

Proof: The proof is in Appendix A. ■

Equation (4) of Theorem 1 clearly indicates the noise use ratio $[{\sigma_m}^2/([\min(\boldsymbol{\Sigma}_t)]^2)]$ of Case 2B) in Algorithm 1 (see Line 22 of Algorithm 1), and the noise use ratio 1 of Case 2A) and C) in Algorithm 1 (see Lines 15 and 30 of Algorithm 1).

By considering $r = 0$ in Result 1) of Theorem 1, we obtain Corollary 1, which presents the classical result on the privacy loss of a single run of the Gaussian mechanism.

Corollary 1: By considering $m = 1$ in Result 1) of Theorem 1, we have that for a randomized algorithm $\widetilde{Q}$ which adds Gaussian noise amount σ to a query Q, the privacy loss with respect to neighboring data sets D and D' is given by $\mathcal{N}([A(D, D')/2], A(D, D'))$ for $A(D, D') := [([\|Q(D) - Q(D')\|_2]^2)/\sigma^2]$.

Corollary 1 has been shown in many prior studies [8]–[10] on the Gaussian mechanism for DP.

By considering $r = 0$ in Result 1) of Theorem 1, we obtain Corollary 2, which presents the privacy loss of the naive algorithm where the noisy response to each query is generated independently using fresh noise.

Corollary 2 (Privacy Loss of the Naive Algorithm Where Each Query Is Answered Independently): Suppose a data set has been used to answer n queries $Q_1, Q_2, \ldots, Q_n$ under DP. Specifically, for $i = 1, 2, \ldots, n$, to answer the ith query Q_i under (ϵ_i, δ_i)-DP, a noisy response $\widetilde{Q}_i$ is generated by adding independent Gaussian noise $\sigma_i := \text{Gaussian}(\Delta_{Q_i}, \epsilon_i, \delta_i)$ to the true query result Q_i, where Δ_{Q_i} is the ℓ_2-sensitivity of Q_i. Then, after answering n queries $Q_1, Q_2, \ldots, Q_n$ independently as above, the privacy loss with respect to neighboring data sets D and D' is given by $\mathcal{N}([F(D, D')/2], F(D, D'))$ for $F(D, D') := \sum_{i=1}^{n}([\|Q_i(D) - Q_i(D')\|_2]^2/{\sigma_i}^2)$.

F. Explaining Privacy Cost Update in Algorithm 1

Among the above cases, Case 2A) and C) do not incur additional privacy cost since they just use previous noisy results and generate fresh Gaussian noise, without access to the data set D. In contrast, Cases 1) and 2B) incur additional privacy cost since they need to access the data set D to compute the true query result $Q_m(D)$. Hence, in Algorithm 1, the privacy cost is updated in Cases 1) and 2B), but not in Case 2A) and C). In this section, we explain the reason that the privacy cost is updated in Algorithm 1 according to Lines 3 and 5 for Case 1), and Lines 18 and 19 for Case 2B).

When our Algorithm 1 is used, we let the above randomized mechanism Y_i be our noisy response function $\widetilde{Q}_i$. When $\widetilde{Q}_1, \widetilde{Q}_2, \ldots, \widetilde{Q}_{i-1}$ on data set D are instantiated as $y_1, y_2, \ldots, y_{i-1}$, if the generation of $\widetilde{Q}_i$ on data set D uses $\widetilde{Q}_j$ for some $j < i$, then the auxiliary information aux_i in the input to $\widetilde{Q}_i$ contains y_j (aux_1 is $\emptyset$). For the consecutive use of our Algorithm 1, it will become clear that the privacy loss, defined by

$$L_{\widetilde{Q}_1, \widetilde{Q}_2, \ldots, \widetilde{Q}_m}(y_1, y_2, \ldots, y_m)$$
$$:= \ln \max_{\text{neighboring datasets } D, D'} \frac{\mathbb{F}\left[\cap_{i=1}^{m}\left[\widetilde{Q}_i(D) = y_i\right]\right]}{\mathbb{F}\left[\cap_{i=1}^{m}\left[\widetilde{Q}_i(D') = y_i\right]\right]}$$

follows a Gaussian probability distribution with mean $(V/2)$ and variance V for some V, denoted by $\mathcal{N}([V/2], V)$. For such a reason that form of privacy loss, the corresponding differential-privacy level is given by the following lemma.

Lemma 2: If the privacy loss of a randomized mechanism Y with respect to neighboring data sets D and D' is given by $\mathcal{N}([V(D, D')/2], V(D, D'))$ for some $V(D, D')$, then Y achieves (ϵ, δ)-DP for ϵ and δ satisfying $\max_{\text{neighboring datasets } D, D'} V(D, D') = [\text{Gaussian}(1, \epsilon, \delta)]^{-2}$.

Proof: The proof details are in Appendix B. ■

Based on the privacy loss defined above, we have the following theorem which explains the rules to update the privacy cost in our Algorithm 1.

Theorem 2: We consider the consecutive use of Algorithm 1 here. Suppose that after answering $Q_1, Q_2, \ldots, Q_{m-1}$ and before answering query Q_m, the privacy loss with respect to neighboring data sets D and D' is given by $\mathcal{N}([A(D, D')/2], A(D, D'))$ for some $A(D, D')$, and the corresponding privacy level can be given by $(\epsilon_{\text{old}}, \delta_{\text{budget}})$-DP. Then, in Algorithm 1, after answering all m queries $Q_1, Q_2, \ldots, Q_{m-1}, Q_m$, we have as follows.

1) The privacy loss with respect to neighboring data sets D and D':
   ① will still be $\mathcal{N}([A(D, D')/2], A(D, D'))$ in Case 2A) and C);
   ② will be $\mathcal{N}([B(D, D')/2], B(D, D'))$ in Case 1) for $B(D, D') := A(D, D') + ([\|Q_m(D) - Q_m(D')\|_2]^2/{\sigma_m}^2)$;
   ③ will be $\mathcal{N}([C(D, D')/2], C(D, D'))$ in Case 2B) for $C(D, D') := A(D, D') + [\|Q_m(D) - Q_m(D')\|_2]^2 \times [(1/{\sigma_m}^2) - (1/[\min(\boldsymbol{\Sigma}_t)]^2)]$.
2) The corresponding privacy level can be given by $(\epsilon_{\text{new}}, \delta_{\text{budget}})$-DP with the following ϵ_{new}:
   ④ $\epsilon_{\text{new}} = \epsilon_{\text{old}}$ in Case 2A) and C);
   ⑤ ${\epsilon_{\text{new}}}^2 = {\epsilon_{\text{old}}}^2 + \epsilon\_\text{squared\_cost}$ in Case 1) for $\epsilon\_\text{squared\_cost}$ satisfying $\text{Gaussian}(\Delta_{Q_m}{:}\sqrt{\epsilon\_\text{squared\_cost}}, \delta_{\text{budget}}) = \sigma_m$,
   ⑥ ${\epsilon_{\text{new}}}^2 = {\epsilon_{\text{old}}}^2 + \epsilon\_\text{squared\_cost}$ in Case 2B) for $\epsilon\_\text{squared\_cost}$ satisfying $[\text{Gaussian}(\Delta_{Q_m}, \sqrt{\epsilon\_\text{squared\_cost}}, \delta_{\text{budget}})]^{-2} = {\sigma_m}^{-2} - [\min(\boldsymbol{\Sigma}_t)]^{-2}$.

Theorem 2 explains the rules to update the privacy cost in Algorithm 1. Specifically, Result ⑤ gives Lines 3 and 5 for Case 1), and Result ⑥ gives Lines 18 and 19 for Case 2B).

Proof: The proof is in Appendix C. ■

G. Analyzing the Total Privacy Cost

Based on Theorem 2, we now analyze the total privacy cost when our system calls Algorithm 1 consecutively.

At the beginning when no query has been answered, we have $V = 0$ (note that $\mathcal{N}(0, 0) \equiv 0$). Then, by induction via Corollary 1 and Theorem 2, for the consecutive use of Algorithm 1, the privacy loss is always in the form of $\mathcal{N}([V/2], V)$ for some V. In our Algorithm 1, the privacy loss changes only when the query being answered belongs to Cases 1) and 2B). More formally, we have the following theorem.

Theorem 3: Among queries $Q_1, Q_2, \ldots, Q_n$, let N_1, N_{2A}, N_{2B}, and N_{2C} be the set of $i \in \{1, 2, \ldots, n\}$ such that Q_i is in Cases 1), 2A)–C), respectively. For queries in Case 2B), let T_{2B} be the set of query types. In Case 2B), for query type $t \in T_{2B}$, suppose the number of type-t queries be m_t, and let these type-t queries be $Q_{j_{t,1}}, Q_{j_{t,2}}, \ldots, Q_{j_{t,m_t}}$ for indices $j_{t,1}, j_{t,2}, \ldots, j_{t,m_t}$ (ordered chronologically) all belonging to N_{2B}. From Case 2B) of Algorithm 1, we have $\sigma_{j_{t,1}} > \sigma_{j_{t,2}} > \ldots > \sigma_{j_{t,m_t}}$, and for $k \in \{2, 3, \ldots, m_t\}$, $\widetilde{Q}_{j_{t,k}}$ is answered by reusing $({\sigma_{j_{t,k}}}^2/{\sigma_{j_{t,k-1}}}^2)$ fraction of old noise in $\widetilde{Q}_{j_{t,k-1}}$; more specifically, $\widetilde{Q}_{j_{t,k}} = Q_{j_{t,k}} + ({\sigma_{j_{t,k}}}^2/{\sigma_{j_{t,k-1}}}^2)[\widetilde{Q}_{j_{t,k-1}} - Q_{j_{t,k-1}}] + \mathcal{N}(0, {\sigma_{j_{t,k}}}^2 - [{\sigma_{j_{t,k}}}^4/{\sigma_{j_{t,k-1}}}^2])$ from Line 22 of Algorithm 1 in Section III-E for Case 2B). We also consider that $\widetilde{Q}_{j_{t,1}}$ is answered by reusing $({\sigma_{j_{t,1}}}^2/{\sigma_{j_{t,0}}}^2)$ fraction of old noise in $\widetilde{Q}_{j_{t,0}}$. Let the ℓ_2-sensitivity of a type-t query be $\Delta(\text{type-}t)$.

In the example provided in Table II, we have $N_1 = \{1, 2, 3\}$, $N_{2A} = \{7\}$, $N_{2B} = \{5, 6, 9, 10, 11, 13\}$, and $N_{2C} = \{8, 12\}$. $T_{2B} = \{\text{type-1}, \text{type-2}, \text{type-3}\}$. In Case 2B), the number of type-1 queries is $m_1 = 2$, and these type-1 queries are Q_6 and Q_{10} so $j_{1,1} = 6$ and $j_{1,2} = 10$ (also $j_{1,0} = 1$ since $\widetilde{Q}_6$ reuses $\widetilde{Q}_1$); the number of type-2 queries is $m_2 = 3$, and these type-2 queries are Q_5, Q_9, and Q_{11} so $j_{2,1} = 5$ and $j_{2,2} = 9$, $j_{2,3} = 11$ (also $j_{2,0} = 2$ since $\widetilde{Q}_5$ reuses $\widetilde{Q}_2$); the number of type-3 queries is $m_3 = 1$, and this type-3 query is Q_{13} so $j_{3,1} = 13$ (also $j_{3,0} = 3$ since $\widetilde{Q}_{13}$ reuses $\widetilde{Q}_3$).

Then, after Algorithm 1 is used to answer all n queries with query Q_i being answered under (ϵ_i, δ_i)-DP, we have

1) The total privacy loss with respect to neighboring data sets D and D' is given by $\mathcal{N}([G(D, D')/2], G(D, D'))$, where

$$G(D, D') := \sum_{i \in N_1} \frac{[\|Q_i(D) - Q_i(D')\|_2]^2}{{\sigma_i}^2} + \sum_{t \in T_{2B}} \left\{ \frac{\left[\left\|Q_{j_{t,m_t}}(D) - Q_{j_{t,m_t}}(D')\right\|_2\right]^2}{{\sigma_{j_{t,m_t}}}^2} - \frac{\left[\left\|Q_{j_{t,0}}(D) - Q_{j_{t,0}}(D')\right\|_2\right]^2}{{\sigma_{j_{t,0}}}^2} \right\} \tag{6}$$

and the first summation is the contribution from queries in Case 1), and the second summation is the contribution from queries in Case 2B). When D and D' iterate the space of neighboring data sets, the maximum of $\|Q_i(D) - Q_i(D')\|$ is Q_i's ℓ_2-sensitivity Δ_{Q_i}, and the maximum of both $\|Q_{j_{t,m_t}}(D) - Q_{j_{t,m_t}}(D')\|_2$ and $\|Q_{j_{t,0}}(D) - Q_{j_{t,0}}(D')\|_2$ are $\Delta(\text{type-}t)$ since $Q_{j_{t,m_t}}$ and $Q_{j_{t,0}}$ are both type-t queries, we obtain

$$\max_{\text{neighboring datasets } D,D'} G\big(D, D'\big) = \sum_{i \in N_1} \frac{{\Delta_{Q_i}}^2}{{\sigma_i}^2} + \sum_{t \in T_{2B}} \left[\frac{\left[\Delta(\text{type-}t)\right]^2}{{\sigma_{j_{t,m_t}}}^2} - \frac{\left[\Delta(\text{type-}t)\right]^2}{{\sigma_{j_{t,0}}}^2} \right]. \tag{7}$$

In the example provided in Table II in Section II, $\max_{\text{neighboring datasets } D,D'} G(D, D')$ is given by

$$\begin{aligned} &\frac{{\Delta_{Q_1}}^2}{{\sigma_1}^2} + \frac{{\Delta_{Q_2}}^2}{{\sigma_2}^2} + \frac{{\Delta_{Q_3}}^2}{{\sigma_3}^2} + \left[\frac{\left[\Delta(\text{type-1})\right]^2}{{\sigma_{10}}^2} - \frac{\left[\Delta(\text{type-1})\right]^2}{{\sigma_1}^2} \right] \\ &\quad + \left[\frac{\left[\Delta(\text{type-2})\right]^2}{{\sigma_{11}}^2} - \frac{\left[\Delta(\text{type-2})\right]^2}{{\sigma_2}^2} \right] \\ &\quad + \left[\frac{\left[\Delta(\text{type-3})\right]^2}{{\sigma_{13}}^2} - \frac{\left[\Delta(\text{type-3})\right]^2}{{\sigma_3}^2} \right] \\ &= \frac{\left[\Delta(\text{type-1})\right]^2}{{\sigma_{10}}^2} + \frac{\left[\Delta(\text{type-2})\right]^2}{{\sigma_{11}}^2} + \frac{\left[\Delta(\text{type-3})\right]^2}{{\sigma_{13}}^2}. \end{aligned}$$

2) From Lemma 2, the total privacy cost of our Algorithm 1 can be given by $(\epsilon_{\text{ours}}, \delta_{\text{budget}})$-DP for ϵ_{ours} satisfying

$$\left[\text{Gaussian}\big(1, \epsilon_{\text{ours}}, \delta_{\text{budget}}\big)\right]^{-2} = \max_{\text{neighboring datasets } D,D'} G(D, D') \tag{8}$$

or (ϵ, δ)-DP for any ϵ and δ satisfying $[\,\text{Gaussian}(1, \epsilon, \delta)]^{-2} = \max_{\text{neighboring datasets } D,D'} G(D, D')$.

Proof: The proof is in Appendix D. ■

Remark 3: Theorem 3 can be used to understand that our Algorithm 1 incurs less privacy cost than that of the naive algorithm where n queries are answered independently. As given in Corollary 2, the privacy loss with respect to neighboring data sets D and D' is given by $\mathcal{N}([F(D, D')/2], F(D, D'))$ for $F(D, D') := \sum_{i=1}^{n}([\|Q_i(D) - Q_i(D')\|_2]^2/{\sigma_i}^2)$. Clearly, $F(D, D') \geq G(D, D')$ for $G(D, D')$ given by (6) above. From Lemma 2, the privacy cost of the naive algorithm can be given by $(\epsilon_{\text{naive}}, \delta_{\text{budget}})$-DP for ϵ_{naive} satisfying $[\,\text{Gaussian}(1, \epsilon_{\text{naive}}, \delta_{\text{budget}})]^{-2} = \max_{\text{neighboring datasets } D,D'} F(D, D')$, which with (8) in Theorem 3 and the expression of $\text{Gaussian}(\cdot, \cdot, \cdot)$ in Lemma 1 implies

$$\frac{\epsilon_{\text{ours}}}{\epsilon_{\text{naive}}} = \sqrt{\frac{\max_{\text{neighboring datasets } D,D'} G(D, D')}{\max_{\text{neighboring datasets } D,D'} F(D, D')}} \leq 1,$$

where the equal sign is taken only when all n queries are different so no noise reuse is incurred in our Algorithm 1.

H. Computing the ℓ_2-Sensitivity of Query

The ℓ_2-sensitivity of a query Q is defined as the maximal ℓ_2 distance between the (true) query results for any neighboring data sets D and D' that differ in one record: $\Delta_Q = \max_{\text{neighboring datasets } D,D'} \|Q(D) - Q(D')\|_2$. For one-dimensional real-valued query Q, Δ_Q is simply the maximal absolute difference between $Q(D)$ and $Q(D')$ for any neighboring data sets D and D'. In Section V for performance evaluation, we define neighboring data sets by considering modifying an entry. Then, if the data set has n users' information, and the domain of each user's income is within the interval [min_income, max_income], then Δ_Q for query Q being the average income of all users is $[(\text{max\_income} - \text{min\_income})/n]$ since this is the maximal variation in the output when a user's record changes. Similarly, Δ_Q for query Q being the percentage of female users is $(1/n)$.

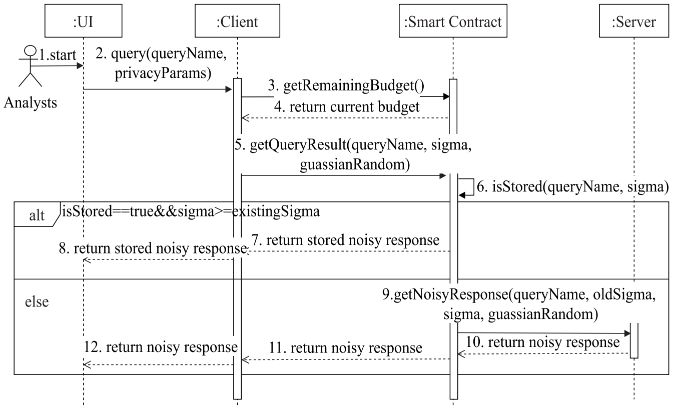
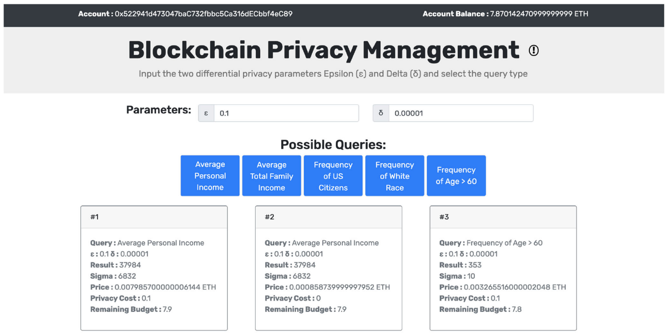

:UI
:Client
:Smart Contract
:Server
1.start
Analysts
2. query(queryName, privacyParams)
3. getRemainingBudget()
4. return current budget
5. getQueryResult(queryName, sigma, guassianRandom)
6. isStored(queryName, sigma)
alt
isStored==true&&sigma>=existingSigma
7. return stored noisy response
8. return stored noisy response
else
9.getNoisyResponse(queryName, oldSigma, sigma, guassianRandom)
10. return noisy response
11. return noisy response
12. return noisy response

Fig. 2. Proposed blockchain-based system working flow for differential-privacy cost management.



Account : 0x522941d473047baC732fbbc5Ca316dECbbf4eC89 **Account Balance :** 7.870142470999999999 ETH

Blockchain Privacy Management

Input the two differential privacy parameters Epsilon (ε) and Delta (δ) and select the query type

Parameters: ε 0.1 δ 0.00001

Possible Queries:

#1

Query : Average Personal Income
ε : 0.1 **δ :** 0.00001
Result : 37984
Sigma : 6832
Price : 0.007985700000006144 ETH
Privacy Cost : 0.1
Remaining Budget : 7.9

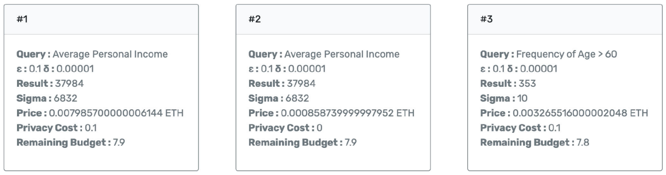
#2

Query : Average Personal Income
ε : 0.1 **δ :** 0.00001
Result : 37984
Sigma : 6832
Price : 0.000858739999997952 ETH
Privacy Cost : 0
Remaining Budget : 7.9

#3

Query : Frequency of Age > 60
ε : 0.1 **δ :** 0.00001
Result : 353
Sigma : 10
Price : 0.003265516000002048 ETH
Privacy Cost : 0.1
Remaining Budget : 7.8

Fig. 3. Screenshot of blockchain-based privacy management system demo.



#1

Query : Average Personal Income
ε : 0.1 **δ :** 0.00001
Result : 37984
Sigma : 6832
Price : 0.007985700000006144 ETH
Privacy Cost : 0.1
Remaining Budget : 7.9

#2

Query : Average Personal Income
ε : 0.1 **δ :** 0.00001
Result : 37984
Sigma : 6832
Price : 0.000858739999997952 ETH
Privacy Cost : 0
Remaining Budget : 7.9

#3

Query : Frequency of Age > 60
ε : 0.1 **δ :** 0.00001
Result : 353
Sigma : 10
Price : 0.003265516000002048 ETH
Privacy Cost : 0.1
Remaining Budget : 7.8

Fig. 4. Displaying of outputs with ϵ privacy cost.

IV. Implementation Challenges of Our Blockchain-Based System

We now discuss challenges and countermeasures during the design and implementation of our blockchain-based system.

Smart Contract Fetches External Data: Ethereum blockchain applications, such as Bitcoin scripts and smart contracts are unable to access and fetch directly the external data they need. However, in our application, `Blockchain` needs to fetch data from `Server` then returns them to `Client`. This requires smart contract to send the HTTP POST request. Hence, we use the Provable, a service integrated with a number of blockchain protocols and can be accessed by nonblockchain applications as well. It guarantees that data fetched from the original data-source is genuine and untampered.

By using the Provable, smart contracts can directly access data from Web sites or APIs. In our case, `Blockchain` can send HTTP requests to `Server` with parameters, and then process and store data after `Server` responds successfully.

Mathematical Operations With Solidity: `Blockchain` is written using solidity language which is designed to target Ethereum virtual machine. However, current solidity language does not have inherent functions for complex mathematical operations, such as taking the square root or logarithm. We write a function to implement the square root operation. To avoid using Lemma 1 to compute logarithm in `Blockchain`, we generate Gaussian noise in `Client`, and pass the value to `Blockchain` as one of the parameters in function QueryMatch. Besides, current solidity version cannot operate float or double type data. To keep the precision, we scale up the noise amount during calculation, and then scale down the value before returning the noisy data to analysts.

V. Implementation and Experiments

In this section, we perform experiments to validate that the proposed system and algorithm are effective in saving privacy cost according to the system flow shown in Fig. 2. More specifically, a user sends a query through the UI, and then `Client` receives the query and forwards it to `Blockchain` smart contract. After the smart contract checks with stored data, it will decide whether to return the noisy response to `Client` directly or forward the request to `Server`. If `Server` receives the request, it will generate and return a noisy response to the smart contract.

A. Experiment Setup

We prototype a Web application based on the system description in Section III. We use the Javascript language to write `Client`, whereas the Solidity language is for `Blockchain` smart contract. Besides, Web3 is used as the Javascript API to exchange information between `Client` and `Blockchain` smart contract, and then Node.js and express Web framework are leveraged to set up `Server`. In addition, MongoDB is used as the database to host the real-world data set. Our designed smart contracts are deployed on the Ropsten [27] testnet with the MetaMask extension of the Chrome browser. The Ropsten testnet is a testing blockchain environment maintained by Ethereum, and it implements the same PoW protocol as the main Ethereum network. Fig. 3 shows the screenshot of our blockchain-based privacy management system. Fig. 4 presents outputs while sending queries using the system.

We evaluate the performance of the proposed DP mechanism based on a real-world data set containing American community survey samples extracted from the *Integrated Public Use Microdata Series* at https://www.ipums.org. There are 5000 records in the data set. Each record includes the following numerical attributes: "total personal income," "total family income," "age," and categorical attributes: "race," "citizenship status." We set the privacy budget as $\epsilon_{\text{budget}} = 8$ and $\delta_{\text{budget}} = 10^{-4}$, which are commonly used to protect the privacy of a data set [28], [29]. We consider five types of queries: 1) "average personal income;" 2) "average total family income;" 3) "frequency of U.S. citizens;" 4) "frequency of white race;" and 5) "frequency of age more than 60." For the privacy parameter of each query Q_i, we sample ϵ_i uniformly from $[0.1, 1.1]$ and sample δ_i uniformly from $[10^{-5}, 10^{-4}]$. The sensitivities of these queries are 202, 404, 0.0002, 0.0002, and 0.0002, respectively. We compute the sensitivity of a query based on Section III-H. For the query "average total personal income," since the user's total personal income ranges from -5000 to 700000 in the data set mentioned above, we assume the domain of total personal income is in the range of $[-10000, 1000000]$ for all possible data sets. The sensitivity is $(1000000 - (-10000))/5000 = 202$ and the mechanism protects the privacy of all data within $[-10000, 1000000]$. Thus, it can protect the privacy of the data set in our experiment. Suppose the received query is "average total family income." In that case, we assume the maximal variation is $[-20000, 2000000]$ for all possible data sets because the total family income's range is $[-5000, 1379500]$ in the data set we use. The sensitivity is $(2000000 - (-20000))/5000 = 404$. Hence, our generated noise with the sensitivity of 404 can protect the privacy of all data within $[-20000, 2000000]$. Therefore, it can protect the privacy of the data set we use as well. The sensitivity for queries "frequency of U.S. citizens," "frequency of white race," and "frequency of age more than 60" is $1/5000 = 0.0002$.

B. Experimental Results

The benchmark of our experiment is a naive scheme which does not contain Algorithm 1 in the smart contract. That is, every query will be forwarded by the smart contract to `Server` to get the noisy response. Hence, no DP cost can be reused in the naive scheme.

First, we use an experiment to validate that our proposed Algorithm 1 is effective in saving privacy cost. Thus, we design a performance comparison experiment by tracking privacy cost using our Algorithm 1 and the naive scheme, respectively. Specifically, we deploy two smart contracts implementing our Algorithm 1 and the naive scheme, respectively, on the Ropsten testnet. Then, we send 150 requests randomly selected in five query types from `Client` of the Web application, and record the privacy cost of each query. As shown in Fig. 5, compared with the naive scheme, the proposed algorithm saves significant privacy cost. When the number of the queries is 150, the differential-privacy cost of Algorithm 1 is about 52% less than that of the naive algorithm. We also observe that the privacy cost in the proposed scheme increases slowly when the number of queries increases, even trending to converge to a specific value. The reason is that, in Algorithm 1, for each query type, we can always partially or fully reuse previous noisy answers when the query type is asked for a second time or more. Therefore, in our scheme, many queries are answered without incurring additional privacy cost if noisy responses fully reuse previous noisy answers.

Second, to prove that the proposed Algorithm 1 retains the accuracy of the data set, we design another experiment to compare the sum of relative errors. We use the same smart contracts as those in the last experiment. We accumulate relative errors incurred in each query. Fig. 6 shows that the sum of relative errors of Algorithm 1 is comparable with that of the naive scheme. Since relative errors are similar between two schemes, our results demonstrate that the proposed Algorithm 1 keeps the accuracy.

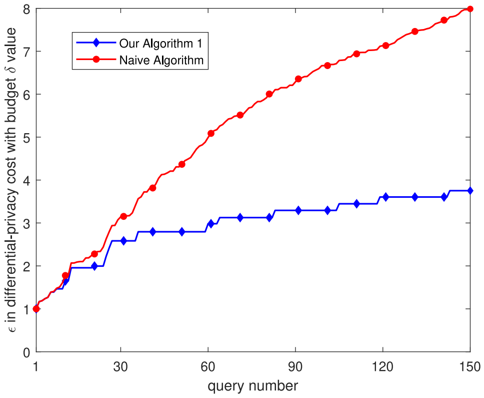


Fig. 5. Performance comparison of the sum of privacy cost.

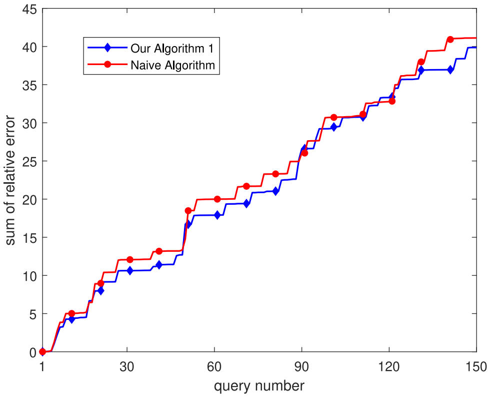


Fig. 6. Performance comparison of the sum of relative error.

As a summary, Figs. 5 and 6 together demonstrate that our Algorithm 1 can save privacy cost significantly without sacrificing the accuracy of the data set.

Third, we evaluate the latency of our system. The latency of our system is affected by the blockchain, MetaMask, network condition, and the server. To shorten the speed of network transmission, we setup a local testnet at http://localhost:3000/ using Ganache-cli client with blockTime set as 15 s [30]. Fig. 7 shows that the latency increases as the number of queries increases. The capacity of Ethereum's throughput is 20∼60 transmission per second (TPS) [31]–[36]. When the number of queries reaches 60, the latency increases significantly. In addition to Ethereum's throughput, both MetaMask and the capacity of the deployed device affect the latency. We test the case when the query results have been saved into the system. Thus, smart contract does not need to send requests to the server. We can obtain query results by using previous query results. The worst case is that smart contract has to send request every query which takes longer time to obtain the result because of the third party service Provable.

Forth, we evaluate the relationship between the query utility and the privacy budget. As defined in [37], the privacy utility of a mechanism satisfies (α, β)-useful if $|\widetilde{Q}_m(D) - Q_m(D)| \leq \alpha$ with probability at least $1 - \beta$. Thus, a small α means that

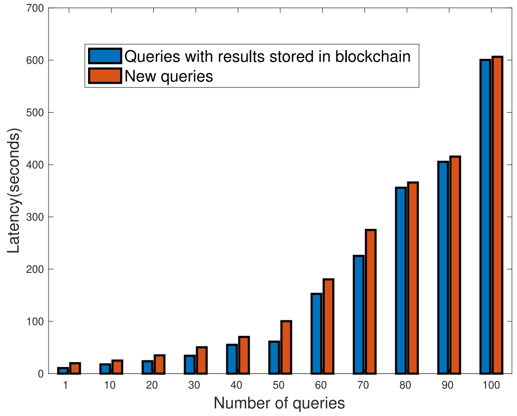


Fig. 7. Latency versus the number of queries.

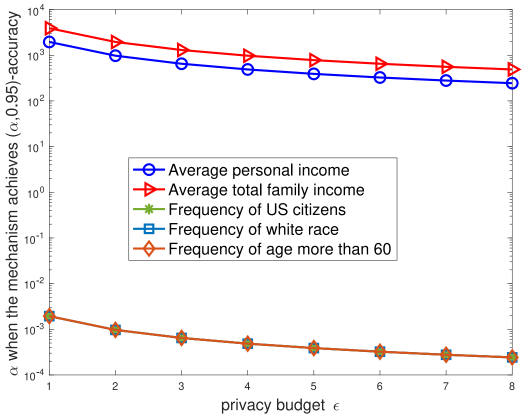


Fig. 8. Utility versus the privacy budget.

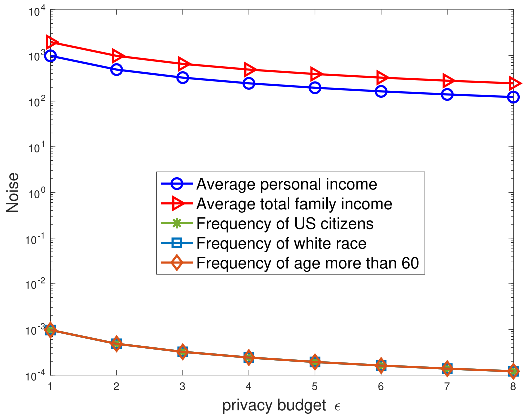


Fig. 9. Noise versus the privacy budget.

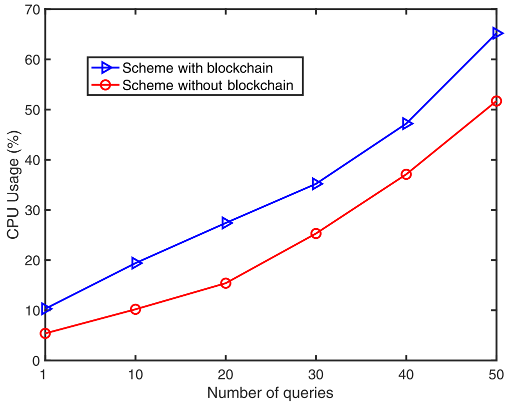


Fig. 10. Compare CPU usages of our scheme with or without the smart contract.

the difference between the perturbed result and the actual result is small, which also reflects that the mechanism has a high utility. The noise added to a query can be calculated as $\sigma = \text{Gaussian}(\Delta_Q, \epsilon, \delta)$, where $\text{Gaussian}(\Delta_Q, \epsilon, \delta) = \sqrt{2\ln(1.25/\delta)} \times (\Delta_Q/\epsilon)$. We set $\delta = 10^{-5}$ and $\epsilon \in [1, 8]$. Appendix E proves that when we set $\beta = 0.05$, $\alpha = 2\sigma$. Figs. 8 and 9 illustrate how the utility and noise change as the privacy budget ϵ increases. Fig. 8 shows the value of α decreases when the privacy budget ϵ increases, meaning that the utility increases. In addition, the amount of noise added reflects the query utility as well. When less noise is added to the query response, the more utility the response gains. Fig. 9 shows that how the noise changes with the privacy budget. As the privacy budget increases, noise decreases, which means that the query utility increases. The amount of noise depends on the privacy budget and the sensitivity value. Queries, such as "frequency of U.S. citizens," "frequency of white race," and "frequency of age more than 60" have the same sensitivity value 0.0002, so the noise added to their responses is the same when the privacy budgets they use are equal.

Fifth, we evaluate the efficiency of computing resources used by the DP scheme using or without the blockchain. The scheme without the blockchain's involvement is when data analysts call the server directly to obtain noisy answers. We simulate this case using "Apache JMeter" [38] API testing software to send queries with DP parameters to the server directly. We consider the CPU usage as the metric for evaluating the efficiency of computing resources [39], [40].

Experimental Settings: We use a MacBook Pro with 2.3-GHz Quad-Core Intel Core i5, and 8-GB 2133-MHz LPDDR3 to run applications. In the experiment, we deploy a locally private blockchain testnet using "Go-Ethereum" platform with three nodes mining. We then hold a separate server locally for testing CPU usage for processing API requests sent from "Apache JMeter." We randomly send 50 queries for testing. At the same time, we use the "Activity Monitor" software [41] to track and obtain their CPU usage [42].

Experimental Results: Fig. 10 compares CPU usages spent by schemes with and without the blockchain. We can observe that the computation efficiency (i.e., CPU usage) in our blockchain-based scheme is higher than that using the server to handle directly. However, it is still acceptable. Node.js and Express.js Web application frameworks, which we use to build the Web server application, are CPU-intensive. One approach to save computing resources when using blockchain is to stop mining when there is not any incoming task. Only start mining when it is necessary. Therefore, our proposed scheme is efficient and practical with acceptable computation cost.

VI. Related Work

In this section, we first compare our paper and a closely related study [19], and then discuss other related work.

A. Comparison With Yang et al. [19]

Yang *et al.* [19] utilized blockchain and DP technologies to achieve the security and privacy protection during data sharing. Compared with [19], we summarize the differences between our work and [19] as follows.

1) Although [19, Algorithm 1] claims to satisfy ϵ-DP, it does not since the noisy output's domain (i.e., the set of all possible values) depends on the input. The explanation is as follows. In [19], for two neighboring data sets D and D', there exists a subset $\mathcal{Y}$ of outputs such that $\mathbb{P}\big[\widetilde{Q}(D) \in \mathcal{Y}\big] > 0$ but $\mathbb{P}\big[\widetilde{Q}(D') \in \mathcal{Y}\big] = 0$. This means $[(\mathbb{P}\big[\widetilde{Q}(D) \in \mathcal{Y}\big])/(\mathbb{P}\big[\widetilde{Q}(D') \in \mathcal{Y}\big])] = \infty > e^{\epsilon}$, which violates ϵ-differential privacy for any $\epsilon < \infty$.
2) Yang *et al.* [19] did not discuss how to choose the small additional privacy parameter in its Algorithm 1.
3) In [19], when a query is asked for the first time, the Laplace mechanism of [7] for ϵ-differential privacy is used to add Laplace noise to the true query result. Afterward, Yang *et al.* [19] added new Laplacian noise on previous noisy output, which makes the new noisy response no longer follow Laplace distribution since the sum of independent Laplace random variables does not follow a Laplace distribution. Hence, the analysis in [19] is not effective.

We consider (ϵ, δ)-DP by using the Gaussian noise. The advantage of Gaussian noise over Laplace noise lies in the easier privacy analysis for the composition of different privacy-preserving algorithms, since the sum of independent Gaussian random variables still follows the Gaussian distribution, while the sum of independent Laplace random variables does not obey a Laplace distribution.

B. Other Related Work

DP, a strong mathematical model to guarantee the database's privacy, has attracted much attention in recent years. Blockchain is a fast-growing technology to provide security and privacy in a decentralized manner [18], [43]–[47]. Feng *et al.* [48] summarized prior studies about privacy protection in the blockchain system, including methodology for identity and transaction privacy preservation. In the following, we will introduce more recent studies utilizing blockchain or privacy techniques to provide privacy or security protection in identity, data, and transactions.

Leveraging Blockchains for Identity Privacy/Security Protection: A few studies have focused on leveraging the blockchain to guarantee privacy/security in access control management or identity protection. For example, Zyskind *et al.* [49] and Xia *et al.* [50] both used blockchain in access control management. Zyskind *et al.* [49] created a decentralized personal data management system to address users' concerns about privacy when using third-party mobile platforms. Xia *et al.* [50] proposed a permissioned blockchain-based data sharing framework to allow only verified users to access the cloud data. Lu *et al.* [51] developed a private and anonymous decentralized crowdsourcing system ZebraLancer, which overcomes data leakage and identity breach in traditional decentralized crowdsourcing. The above studies focus on identity privacy because the Blockchain is anonymous, whereas they do not consider the privacy protection for the database.

Leveraging Blockchains for Data Privacy/Security Protection: In addition to the identity privacy preservation, Hu *et al.* [52] replaced the central server with a smart contract and construct a decentralized privacy-preserving search scheme for computing encrypted data while ensuring the privacy of data to prevent from misbehavings of a malicious centralized server. Luongo and Pon [53] used secure multiparty computation to design a privacy primitive named Keep which allows contracts to manage and use private data without exposing the data to the public blockchain for protecting smart contracts on public blockchains. Alternatively, we use DP standard to guarantee privacy. Moreover, blockchains are popular to be used for security protection of data sharing in IoT scenarios [44], [45], [50].

Leveraging Blockchains for Transaction Privacy/Security Protection: Moreover, some previous studies use blockchain to guarantee security and privacy in transactions. For example, Henry *et al.* [17] proposed that the blockchain should use mechanisms that piggyback on the overlay network, which is ready for announcing transactions to de-link users' network-level information instead of using an external service such as Tor to protect users' privacy. Gervais [31] proposed a quantitative framework to analyze the security of proof-of-work in blockchains, where the framework's inputs included security, consensus, and network parameters. Herrera-Joancomartí and Pérez-Solà [54] focused on privacy in bitcoin transactions. Sani *et al.* [55] proposed a new blockchain Xyreum with high-performance and scalability to secure transactions in the Industrial IoT.

Leveraging Differential Privacy for Protecting Privacy of IoT Data: IoT devices collect users' usage status periodically, which may contain sensitive information, such as the energy consumption or location information. To avoid the privacy leakage, some studies use DP mechanisms to protect the privacy of data [1]–[5]. For example, Tudor *et al.* [1] proposed a streaming-based framework, Bes, to disclose IoT data. Hassan *et al.* [2] treated each IoT node as a node of blockchain to guarantee the IoT nodes' security, and they leverage DP mechanisms to protect the privacy of data of each node. To prevent adversaries from intercepting the Internet traffic from/to the smart home gateway or profile residents' behaviors through digital traces, Liu *et al.* [4] leveraged DP to develop a framework to prevent from attacks. However, none of them discusses how to reuse the DP budget.

Differentially Private Algorithms: Some DP algorithms are proposed to provide privacy protection. Xiao *et al.* [56] proposed an algorithm to correlate the Lapalce noise added to different queries to improve the overall accuracy. Given a series of counting queries, the mechanism proposed by Li and Miklau [57] selected a subset of queries to answer privately and used their noisy answers to derive answers for the remaining queries. For a set of nonoverlapping counting queries, Kellaris and Papadopoulos [58] preprocessed the counts by elaborate grouping and smoothing them via averaging to reduce the sensitivity and thus the amount of injected noise. Given a workload of queries, Yaroslavtsev *et al.* [59] introduced a solution to balance accuracy and efficiency by answering some queries more accurately than others.

VII. Discussion and Future Work

In this section, we will discuss the meaning of DP parameters, privacy of the smart contract, and queries.

A. Differential Privacy Parameters

The value of DP parameter ϵ represents the level of the protection provided by DP mechanisms. McSherry and Mahajan [60] and Aaby *et al.* [61] quantified the strength of DP as follows. When $\epsilon = 0$, the DP mechanism provides perfect privacy. Then, when $\epsilon \leq 0.1$, the protection is considered as strong, while $\epsilon \geq 10$, the privacy protection is weak. The privacy parameter δ represents the small probability that a record gets altered in the database, so it should be very small. We sample δ uniformly from $[10^{-5}, 10^{-4}]$ for each query.

B. Privacy for Smart Contract

A smart contract is publicly available when it is deployed to the public blockchain. In our experiment, attackers can obtain the algorithm implemented in the smart contract. However, they still cannot obtain the accurate responses from noisy results even if they obtain the algorithm. There are some approaches that can be used to protect the privacy of the smart contract as follows:

First, Kosba *et al.* [62] proposed Hawk, a framework to build the privacy-preserving smart contracts. Hawk enables that programmers write private smart contracts without considering cryptography. The framework will generate a cryptographic protocol during compiling automatically.

Second, we can partially address this problem by deploying Ethereum to a private blockchain. We may combine the private blockchain and proof-of-authority [63] consensus mechanism. When Ethereum is deployed to the private blockchain, the private blockchain can set access control. Thus, the attackers need to break access control before accessing the smart contract. Therefore, when using a private blockchain, we consider access control to protect the smart contracts' privacy.

As it is complex to protect the smart contract's privacy, we would like to consider the smart contract's privacy as our future work.

C. Privacy for Queries

DP mechanisms consider that data analysts are untrusted, and the curator who holds the database is trusted. The trusted curator stores the database and responds to statistical queries made by an untrusted analyst so that DP will not protect the privacy of data analysts' queries. Moreover, DP supports statistical queries that may not include much sensitive information. When we use the smart contract, some data analysts may worry about the privacy of their queries. There are two ways to protect the privacy of queries.

First, we may use the private blockchain and conduct experiments using the private blockchain and proof-of-authority consensus mechanism. Ethereum also supports deploying the smart contract to the private blockchain. In this case, the smart contract can be considered trusted, so that the sensitive information of queries will not leak.

Second, we may combine other cryptographic techniques with DP. For example, Agarwal *et al.* [64] designed the encrypted databases (EDBs) that support differentially private statistical queries. In their paper, both the curator and the analyst are untrusted. The curator can outsource the database to an untrusted server securely by leveraging EDBs to encrypt operations. Since the privacy protection for queries is complicated and may involve more privacy and security techniques, we would like to consider the privacy of queries as our future work.

Third, query privacy can be avoided by fixing the types of queries. Since the privacy budget is quite limited, it is impossible to let data analysts ask too many questions. Thus, one solution is to control types of queries. The system builder may take some time to select commonly used questions by data analysts, and then they set a dropdown list for data analysts to select questions. In this case, our system will not leak the queries' privacy because queries are standard.

D. Difference Between Our Proposed Scheme and Normal DP Schemes

We predefine queries for efficiently calculating the sensitivity values and saving users' time. The sensitivity values for different queries should be predefined even if we do not use blockchain. When a query comes in many times with different DP parameters, our scheme will play an essential role in saving the DP budget. For example, many companies are trying to send the same query to a data set because of the similar data analysis tasks. In this case, some of the privacy budgets can be saved. Since the privacy budget is a scarce resource regarding to the data set, it is necessary to use our scheme.

However, when a query is seen for the first time, our scheme can only treat it as the new query. Since the DP budget is quite limited, and sensitivities have to be calculated beforehand, a data set will not support too many different statistical queries. If our system is implemented in the real world, similar to Fig. 3, a list of queries supported maybe provided to control the variations of queries instead of letting data analysts type in different queries freely.

VIII. Conclusion

In this article, we use a blockchain-based approach for tracking and saving differential-privacy cost. In our design, we propose an algorithm that reuses noise fully or partially for different instances of the same query type to minimize the accumulated privacy cost. The efficiency of the algorithm is proved via a rigorous mathematical proof. Moreover, we design a blockchain-based system for conducting real-world experiments to confirm the effectiveness of the proposed approach.

Appendix A
Proof of Theorem 1

1) As noted in the statement of Theorem 1, we suppose that before answering query Q_m and after answering

$Q_1, Q_2, \ldots, Q_{m-1}$, the privacy loss $L_{\widetilde{Q}_1\|\widetilde{Q}_2\|\cdots\|\widetilde{Q}_{m-1}}(D, D')$ is given by $\mathcal{N}([A(D, D')/2], A(D, D'))$ for some $A(D, D')$. Later, we will show the existence of such $A(D, D')$. Then, when y_i follows the probability distribution of random variable $\widetilde{Q}_i(D)$ for each $i \in \{1, 2, \ldots, m-1\}$, we have the following for the privacy loss $L_{\widetilde{Q}_1\|\widetilde{Q}_2\|\cdots\|\widetilde{Q}_{m-1}}(D, D'; y_1, y_2, \ldots, y_{m-1})$:

$$
\begin{aligned}
&L_{\widetilde{Q}_1\|\widetilde{Q}_2\|\cdots\|\widetilde{Q}_{m-1}}\big(D, D'; y_1, y_2, \ldots, y_{m-1}\big)\\
&\quad := \ln \frac{\mathbb{F}\Big[\cap_{i=1}^{m-1}\big[\widetilde{Q}_i(D) = y_i\big]\Big]}{\mathbb{F}\Big[\cap_{i=1}^{m-1}\big[\widetilde{Q}_i(D') = y_i\big]\Big]} \sim \mathcal{N}\left(\frac{A\big(D, D'\big)}{2}, A\big(D, D'\big)\right),
\end{aligned}
\tag{9}
$$

where we use "$\sim$" to mean "obeying the distribution."

Now, we need to analyze the privacy loss after answering the m queries $Q_1, Q_2, \ldots, Q_m$. We look at the privacy loss $L_{\widetilde{Q}_1\|\widetilde{Q}_2\|\cdots\|\widetilde{Q}_m}(D, D'; y_1, y_2, \ldots, y_m)$ defined as follows:

$$
\begin{aligned}
&L_{\widetilde{Q}_1\|\widetilde{Q}_2\|\cdots\|\widetilde{Q}_m}\big(D, D'; y_1, y_2, \ldots, y_m\big)\\
&\quad := \ln \frac{\mathbb{F}\big[\cap_{i=1}^{m}\big[\widetilde{Q}_i(D) = y_i\big]\big]}{\mathbb{F}\big[\cap_{i=1}^{m}\big[\widetilde{Q}_i(D') = y_i\big]\big]}.
\end{aligned}
\tag{10}
$$

Hence, we use (9) to analyze (10). From (3), since $\widetilde{Q}_m(D)$ is generated by reusing $\widetilde{Q}_j(D)$ and generating additional noise (if necessary), where j is an integer in $\{1, 2, \ldots, m-1\}$ as noted in the statement of Theorem 1, we have

$$
\begin{aligned}
&L_{\widetilde{Q}_1\|\widetilde{Q}_2\|\cdots\|\widetilde{Q}_m}\big(D, D'; y_1, y_2, \ldots, y_m\big)\\
&= \ln \frac{\mathbb{F}\Big[\cap_{i=1}^{m-1}\big[\widetilde{Q}_i(D) = y_i\big]\Big]\mathbb{F}\big[\widetilde{Q}_m(D) = y_m \mid \widetilde{Q}_j(D) = y_j\big]}{\mathbb{F}\Big[\cap_{i=1}^{m-1}\big[\widetilde{Q}_i(D') = y_i\big]\Big]\mathbb{F}\big[\widetilde{Q}_m(D') = y_m \mid \widetilde{Q}_j(D') = y_j\big]}\\
&= \ln \frac{\mathbb{F}\Big[\cap_{i=1}^{m-1}\big[\widetilde{Q}_i(D) = y_i\big]\Big]}{\mathbb{F}\Big[\cap_{i=1}^{m-1}\big[\widetilde{Q}_i(D') = y_i\big]\Big]}\\
&\quad + \ln \frac{\mathbb{F}\big[\widetilde{Q}_m(D) = y_m \mid \widetilde{Q}_j(D) = y_j\big]}{\mathbb{F}\big[\widetilde{Q}_m(D') = y_m \mid \widetilde{Q}_j(D') = y_j\big]}.
\end{aligned}
\tag{11}
$$

We now discuss the first term $\ln[(\mathbb{F}[\cap_{i=1}^{m-1}[\widetilde{Q}_i(D) = y_i]])/(\mathbb{F}[\cap_{i=1}^{m-1}[\widetilde{Q}_i(D') = y_i]])]$ and the second term $\ln[(\mathbb{F}[\widetilde{Q}_m(D) = y_m \mid \widetilde{Q}_j(D) = y_j])/(\mathbb{F}[\widetilde{Q}_m(D') = y_m \mid \widetilde{Q}_j(D') = y_j])]$ in the last row of (11). To begin with, from (9), the first term in the last row of (11) follows the Gaussian distribution $\mathcal{N}([A(D, D')/2], A(D, D'))$. Next, we analyze the second term in the last row of (11).

When $\widetilde{Q}_j(D)$ and $\widetilde{Q}_m(D)$ take y_j and y_m, respectively, $\widetilde{Q}_j(D) - Q_j(D)$ and $\widetilde{Q}_m(D) - Q_m(D) - r[\widetilde{Q}_j(D) - Q_j(D)]$ take the following defined g_j and g_m, respectively:

$$g_j := y_j - Q_j(D), \tag{12}$$

$$g_m := y_m - Q_m(D) - r\big[y_j - Q_j(D)\big]. \tag{13}$$

For D' being a neighboring dataset of D, we further define

$$h_j := Q_j(D) - Q_j\big(D'\big), \tag{14}$$

$$h_m := Q_m(D) - Q_m\big(D'\big), \tag{15}$$

so that

$$g_j + h_j = y_j - Q_j\big(D'\big), \tag{16}$$

$$g_m + h_m - rh_j = y_m - Q_m\big(D'\big) - r\big[y_j - Q_j\big(D'\big)\big]. \tag{17}$$

Note that h_j and h_m are the same since Q_j and Q_m are the same. From the above analysis, we obtain

$$
\begin{aligned}
&\mathbb{F}\big[\widetilde{Q}_m(D) = y_m \mid \widetilde{Q}_j(D) = y_j\big]\\
&\quad = \mathbb{F}\left[\begin{array}{l}\widetilde{Q}_m(D) - Q_m(D) - r\big[\widetilde{Q}_j(D) - Q_j(D)\big] = g_m\\ \mid \widetilde{Q}_j(D) = y_j\end{array}\right]\\
&\quad \overset{\text{(b)}}{=} \frac{1}{\sqrt{2\pi\left(\sigma_m{}^2 - r^2\sigma_j{}^2\right)}} e^{-\frac{g_m{}^2}{2\left(\sigma_m{}^2 - r^2\sigma_j{}^2\right)}},
\end{aligned}
\tag{18}
$$

where step (b) follows since where $\widetilde{Q}_j(D) - Q_j(D)$ is a zero-mean Gaussian random variable with variance σ_j^2 and $\widetilde{Q}_m(D) - Q_m(D) - r[\widetilde{Q}_j(D) - Q_j(D)]$ is a zero-mean Gaussian random variable with variance $\sigma_m{}^2 - r^2\sigma_j{}^2$.

Similarly, for data set D', we have

$$
\begin{aligned}
&\mathbb{F}\big[\widetilde{Q}_m\big(D'\big) = y_m \mid \widetilde{Q}_j\big(D'\big) = y_j\big]\\
&\quad = \mathbb{F}\left[\begin{array}{l}\widetilde{Q}_m\big(D'\big) - Q_m\big(D'\big) - r\big[\widetilde{Q}_j\big(D'\big) - Q_j\big(D'\big)\big]\\ = g_m + h_m - rh_j\\ \mid \widetilde{Q}_j\big(D'\big) = y_j\end{array}\right]\\
&\quad \overset{\text{(b)}}{=} \frac{1}{\sqrt{2\pi\left(\sigma_m{}^2 - r^2\sigma_j{}^2\right)}} e^{-\frac{\left(g_m + h_m - rh_j\right)^2}{2\left(\sigma_m{}^2 - r^2\sigma_j{}^2\right)}},
\end{aligned}
\tag{19}
$$

where step (b) follows since where $\widetilde{Q}_j(D') - Q_j(D')$ is a Gaussian random variable with variance σ_j^2 and $\widetilde{Q}_m(D') - Q_m(D') - r[\widetilde{Q}_j(D') - Q_j(D')$ is a zero-mean Gaussian random variable with variance $\sigma_m{}^2 - r^2\sigma_j{}^2$.

Then

$$
\begin{aligned}
&\ln \frac{\mathbb{F}\big[\widetilde{Q}_m(D) = y_m \mid \widetilde{Q}_j(D) = y_j\big]}{\mathbb{F}\big[\widetilde{Q}_m(D') = y_m \mid \widetilde{Q}_j(D') = y_j\big]}\\
&= \ln \frac{\frac{1}{\sqrt{2\pi\left(\sigma_m{}^2 - r^2\sigma_j{}^2\right)}} e^{-\frac{g_m{}^2}{2\left(\sigma_m{}^2 - r^2\sigma_j{}^2\right)}}}{\frac{1}{\sqrt{2\pi\left(\sigma_m{}^2 - r^2\sigma_j{}^2\right)}} e^{-\frac{\left(g_m + h_m - rh_j\right)^2}{2\left(\sigma_m{}^2 - r^2\sigma_j{}^2\right)}}}\\
&= \frac{\big(g_m + h_m - rh_j\big)^2 - g_m{}^2}{2\big(\sigma_m{}^2 - r^2\sigma_j{}^2\big)}\\
&= \frac{g_m\big(h_m - rh_j\big)}{\sigma_m{}^2 - r^2\sigma_j{}^2} + \frac{\big(h_m - rh_j\big)^2}{2\big(\sigma_m{}^2 - r^2\sigma_j{}^2\big)}.
\end{aligned}
\tag{20}
$$

The above (20) presents the second term in the last row of (11). At first glance, it may seem that the first term $\ln[(\mathbb{F}[\cap_{i=1}^{m-1}[\widetilde{Q}_i(D) = y_i]])/(\mathbb{F}[\cap_{i=1}^{m-1}[\widetilde{Q}_i(D') = y_i]])]$ and the second term $\ln[(\mathbb{F}[\widetilde{Q}_m(D) = y_m \mid \widetilde{Q}_j(D) = y_j])/(\mathbb{F}[\widetilde{Q}_m(D') = y_m \mid \widetilde{Q}_j(D') = y_j])]$ in the last row of (11) are dependent since they both involve y_j. However, we have shown from (20) above that the second term in the last row of (11) depends on only the random variable g_m [note that terms in (20) other than g_m are all given], which is the amount of additional Gaussian noise used to generated $\widetilde{Q}_m(D)$ according to (3) and (13); i.e., the second term in the last row of (11) is actually independent of the first term in the last row of (11). From (9), the first term in the last row of (11) follows the Gaussian distribution $\mathcal{N}([A(D, D')/2], A(D, D'))$. Next, we show that (20) presenting the second term in the last row of (11) also follows a Gaussian distribution.

Since g_m follows a zero-mean Gaussian distribution with variance ${\sigma_m}^2 - r^2{\sigma_j}^2$, clearly $([g_m(h_m - rh_j)]/[{\sigma_m}^2 - r^2{\sigma_j}^2])$ follows a zero-mean Gaussian distribution with variance given by:

$$\left[\frac{(h_m - rh_j)}{{\sigma_m}^2 - r^2{\sigma_j}^2}\right]^2 \times \left({\sigma_m}^2 - r^2{\sigma_j}^2\right) = \frac{(h_m - rh_j)^2}{{\sigma_m}^2 - r^2{\sigma_j}^2}. \quad (21)$$

Since Q_m and Q_j are the same, we obtain from (14) and (15) that $h_j = h_m = Q_m(D) - Q_m(D')$, which we use to write (21) as

$$\frac{\left[\left\|Q_m(D) - Q_m(D')\right\|_2\right]^2(1-r)^2}{{\sigma_m}^2 - r^2{\sigma_j}^2}. \quad (22)$$

Summarizing the above, privacy loss is

$$B_r(D, D') := A(D, D') + \frac{\left[\left\|Q_m(D) - Q_m(D')\right\|_2\right]^2(1-r)^2}{{\sigma_m}^2 - r^2{\sigma_j}^2}. \quad (23)$$

As noted in the statement of Theorem 1, we suppose that before answering query Q_m and after answering $Q_1, Q_2, \ldots, Q_{m-1}$, the privacy loss $L_{\widetilde{Q}_1\|\widetilde{Q}_2\|\cdots\|\widetilde{Q}_{m-1}}(D, D')$ is given by $\mathcal{N}([A(D, D')/2], A(D, D'))$ for some $A(D, D')$. With the above result (23), we can actually show that there indeed exists such $A(D, D')$. This follows from mathematical induction. For the base case; i.e., when only one query is answered, the result follows from [65, Lemma 3]. The induction step is given by the above result (23). Hence, we have shown the existence of $A(D, D')$. With this result and (23), we have completed proving Result 1) of Theorem 1.

2) The optimal r is obtained by minimizing $B_r(D, D')$ and hence minimizing $[(1 - r)^2/({\sigma_m}^2 - r^2{\sigma_j}^2)]$. Analyzing the monotonicity of this expression, we derive the optimal r as in (4). The first-order derivative of $B_r(D, D')$ to r is

$$B_r(D, D')' = \frac{-2(r{\sigma_j}^2 - {\sigma_m}^2)(r-1)}{(r^2{\sigma_j}^2 - {\sigma_m}^2)^2}. \quad (24)$$

1) *Case 1:* if $\sigma_m \geq \sigma_j$, $B_r(D, D')' \geq 0$ when $r \in [1, (\sigma_m/\sigma_j)]$, and $B_r(D, D')' < 0$ when $r \in (-\infty, 1) \cup ([\sigma_m/\sigma_j], +\infty)$. Hence, the optimal r to minimize $B_r(D, D')$ is at $r = 1$.
2) *Case 2:* if $\sigma_m < \sigma_j$, $B_r(D, D')' \geq 0$ when $r \in [(\sigma_m/\sigma_j), 1]$, and $B_r(D, D')' < 0$ when $r \in (-\infty, [\sigma_m/\sigma_j]) \cup (1, +\infty)$. Hence, the optimal r to minimize $B_r(D, D')$ is at $r = ([\sigma_m/\sigma_j])^2$.

Thus, we obtain optimal values of r as (4).

Appendix B
Proof of Lemma 2

Consider a query R with ℓ_2-sensitivity being 1. Let $\widetilde{R}$ be the mechanism of adding Gaussian noise amount $\mu := (1/[\sqrt{\max_{\text{neighboring datasets } D,D'} V(D, D')}])$ to R. From Corollary 1, the privacy loss of randomized mechanism $\widetilde{R}$ with respect to neighboring data sets D and D' is given by $\mathcal{N}([U(D, D')/2], U(D, D'))$ for $U(D, D') := ([\|R(D) - R(D')\|_2]^2/\mu^2)$. By considering the ℓ_2-sensitivity of R (i.e., $\|R(D) - R(D')\|_2$) as 1, $\max_{\text{neighboring datasets } D,D'} V(D, D')$ and $\max_{\text{neighboring datasets } D,D'} U(D, D')$ are the same. In addition, from Theorem 5 of [65], letting Y (resp., $\widetilde{R}$) satisfy (ϵ, δ)-differential privacy can be converted to a condition on $\max_{\text{neighboring datasets } D,D'} V(D, D')$ (resp., $\max_{\text{neighboring datasets } D,D'} U(D, D')$). Then, letting Y satisfy (ϵ, δ)-differential privacy is the same as letting $\widetilde{R}$ satisfy (ϵ, δ)-differential privacy. From Lemma 1, $\widetilde{R}$ achieves (ϵ, δ)-differential privacy with $\mu = \text{Gaussian}(1, \epsilon, \delta)$; i.e., if $\max_{\text{neighboring datasets } D,D'} V(D, D') = [\text{Gaussian}(1, \epsilon, \delta)]^{-2}$. Summarizing the above, we complete proving Lemma 2.

Appendix C
Proof of Theorem 2

We use Theorem 1 to show Results ① ② and ③ of Theorem 2. Proof of ①: In Case 2A) and C), Q_m can reuse previous noise. Hence, the privacy loss will still be $\mathcal{N}([A(D, D')/2], A(D, D'))$ according to (5).

Proof of ②: In Case 1), Q_m cannot reuse previous noisy answers, and the new noise follows $\mathcal{N}(0, \sigma_m)$. Thus, $B(D, D') := A(D, D') + ([\|Q_m(D) - Q_m(D')\|_2]^2/{\sigma_m}^2)$.

Proof of ③: In Case 2B), Q_m can reuse previous noisy answers partially, so we can prove it using (5).

Then, Lemma 2 further implies Results ④ ⑤ and ⑥ of Theorem 2.

Proof of ④: Q_m can fully reuse the old noisy result in Case 2A) and C). Thus, the privacy level does not change.

Proof of ⑤: From Lemma 2, we have $\max_{\text{neighboring datasets} D,D'} A(D, D') = [\text{Gaussian}(1, \epsilon_{\text{old}}, \delta_{\text{budget}})]^{-2}$ and $\max_{\text{neighboring datasets } D,D'} \{A(D, D') + [\|Q_m(D) - Q_m(D')\|_2]^2 \times (1/{\sigma_m}^2)\} = [\text{Gaussian}(1, \epsilon_{\text{new}}, \delta_{\text{budget}})]^{-2}$. The above two equations yield $[\text{Gaussian}(1, \epsilon_{\text{new}}, \delta_{\text{budget}})]^{-2} - [\text{Gaussian}(1, \epsilon_{\text{old}}, \delta_{\text{budget}})]^{-2} = \max_{\text{neighboring datasets } D,D'} [\|Q_m(D) - Q_m(D')\|_2]^2 \times (1/{\sigma_m}^2) = {\Delta_{Q_m}}^2 \times (1/{\sigma_m}^2) = {\sigma_m}^2$. Hence, $\text{Gaussian}(\Delta_{Q_m}, \epsilon\_\text{squared\_cost}, \delta_{\text{bugdet}}) = \sigma_m$.

Proof of ⑥: From Lemma 2, we have $\max_{\text{neighboring datasets} D,D'} A(D, D') = [\text{Gaussian}(1, \epsilon_{\text{old}}, \delta_{\text{budget}})]^{-2}$ and

$$\max_{\text{neighboring datasets } D,D'} \left\{A(D, D') + \left[\left\|Q_m(D) - Q_m(D')\right\|_2\right]^2 \times \left[\frac{1}{{\sigma_m}^2} - \frac{1}{[\min(\mathbf{\Sigma}_t)]^2}\right]\right\} = \left[\text{Gaussian}(1, \epsilon_{\text{new}}, \delta_{\text{budget}})\right]^{-2}.$$

The above two equations yield

$$\begin{aligned}&\left[\text{Gaussian}(1, \epsilon_{\text{new}}, \delta_{\text{budget}})\right]^{-2} - \left[\text{Gaussian}(1, \epsilon_{\text{old}}, \delta_{\text{budget}})\right]^{-2}\\ &\quad = \max_{\text{neighboring datasets } D,D'} \left[\left\|Q_m(D) - Q_m(D')\right\|_2\right]^2\\ &\qquad \times \left[\frac{1}{{\sigma_m}^2} - \frac{1}{[\min(\mathbf{\Sigma}_t)]^2}\right]\\ &\quad = {\Delta_{Q_m}}^2 \times \left[\frac{1}{{\sigma_m}^2} - \frac{1}{[\min(\mathbf{\Sigma}_t)]^2}\right].\end{aligned}$$

Then using the expression of $\text{Gaussian}(\Delta_Q, \epsilon, \delta)$ from Lemma 1, we further obtain Result ⑥.

Appendix D
Proof of Theorem 3

First, from Theorem 2, after Algorithm 1 is used to answer all n queries with query Q_i being answered under (ϵ_i, δ_i)-differential privacy, the total privacy loss with respect to neighboring data sets D and D' is given by $\mathcal{N}([G(D, D')/2], G(D, D'))$ for some $G(D, D')$.

Next, we use Theorem 2 to further show that the expression of $G(D, D')$ is given by (6). From Theorem 2, among all queries, only queries belonging to Cases 1) and 2B) contribute to $G(D, D')$. Below we discuss the contributions, respectively.

With N_1 denoting the set of $i \in \{1, 2, \ldots, n\}$ such that Q_i is in Cases 1), we know from Result ② of Theorem 2 that the contributions of queries in Cases 1) to $G(D, D')$ is given by

$$\sum_{i \in N_1} \frac{\left[\left\|Q_i(D) - Q_i(D')\right\|_2\right]^2}{{\sigma_i}^2}. \tag{25}$$

Below we use Result ③ of Theorem 2 to compute the contributions of queries in Case 2B) to $G(D, D')$. For T_{2B} being the set of query types in Case 2B), we discuss each query type $t \in T_{2B}$, respectively.

From Result ③ of Theorem 2, the contribution to $G(D, D')$ by answering $Q_{j_{t,1}}$ under differential privacy is

$$\left[\left\|Q_{j_{t,1}}(D) - Q_{j_{t,1}}(D')\right\|_2\right]^2 \left(\frac{1}{\sigma^2_{j_{t,1}}} - \frac{1}{\sigma^2_{j_{t,0}}}\right).$$

Similarly, the contribution to $G(D, D')$ by answering $Q_{j_{t,2}}$ under differential privacy is

$$\left[\left\|Q_{j_{t,2}}(D) - Q_{j_{t,2}}(D')\right\|_2\right]^2 \left(\frac{1}{\sigma^2_{j_{t,2}}} - \frac{1}{\sigma^2_{j_{t,1}}}\right).$$

Similar analyzes are repeated for additional type-t queries in Case 2B). In particular, for each $s \in \{1, 2, \ldots, m_t\}$, the contribution to $G(D, D')$ by answering $Q_{j_{t,s}}$ under differential privacy is

$$\left[\left\|Q_{j_{t,s}}(D) - Q_{j_{t,s}}(D')\right\|_2\right]^2 \left(\frac{1}{\sigma^2_{j_{t,s}}} - \frac{1}{\sigma^2_{j_{t,s-1}}}\right). \tag{26}$$

Summing all (26) for $s \in \{1, 2, \ldots, m_t\}$, we obtain that for each query type $t \in T_{2B}$, the contributions to $G(D, D')$ by answering $Q_{j_{t,1}}, Q_{j_{t,2}}, \ldots, Q_{j_{t,m_t}}$ under differential privacy is

$$\sum_{s \in \{1,2,\ldots,m_t\}} \left[\left\|Q_{j_{t,s}}(D) - Q_{j_{t,s}}(D')\right\|_2\right]^2 \left(\frac{1}{\sigma^2_{j_{t,s}}} - \frac{1}{\sigma^2_{j_{t,s-1}}}\right). \tag{27}$$

Since $Q_{j_{t,0}}, Q_{j_{t,1}}, \ldots, Q_{j_{t,m_t}}$ for $j_{t,0}, j_{t,1}, \ldots, j_{t,m_t}$ are all type-t queries, $\|Q_{j_{t,s}}(D) - Q_{j_{t,s}}(D')\|_2$ are all the same for $s \in \{1, 2, \ldots, m_t\}$. Hence, we write (27) as

$$\sum_{s \in \{1,2,\ldots,m_t\}} \left\{ \frac{\left[\left\|Q_{j_{t,s}}(D) - Q_{j_{t,s}}(D')\right\|_2\right]^2}{\sigma^2_{j_{t,s}}} - \frac{\left[\left\|Q_{j_{t,s-1}}(D) - Q_{j_{t,s-1}}(D')\right\|_2\right]^2}{\sigma^2_{j_{t,s-1}}} \right\}$$

$$= \frac{\left[\left\|Q_{j_{t,m_t}}(D) - Q_{j_{t,m_t}}(D')\right\|_2\right]^2}{\sigma^2_{j_{t,m_t}}} - \frac{\left[\left\|Q_{j_{t,0}}(D) - Q_{j_{t,0}}(D')\right\|_2\right]^2}{\sigma^2_{j_{t,0}}}. \tag{28}$$

Summing all (28) for $t \in T_{2B}$, the contributions to $G(D, D')$ by answering all queries in Case 2B) is

$$\sum_{t \in T_{2B}} \left\{ \frac{\left[\left\|Q_{j_{t,m_t}}(D) - Q_{j_{t,m_t}}(D')\right\|_2\right]^2}{\sigma^2_{j_{t,m_t}}} - \frac{\left[\left\|Q_{j_{t,0}}(D) - Q_{j_{t,0}}(D')\right\|_2\right]^2}{\sigma^2_{j_{t,0}}} \right\}. \tag{29}$$

Then, $G(D, D')$ as the sum of (25) and (29) is given by (6).

Summarizing the above, we have proved that after Algorithm 1 is used to answer all n queries under differential privacy, the total privacy loss with respect to neighboring data sets D and D' is given by $\mathcal{N}([G(D, D')/2], G(D, D'))$ for $G(D, D')$ in (6). Furthermore, under

$$\max_{\text{neighboring datasets } D, D'} \left\|Q_i(D) - Q_i(D')\right\|_2 = \Delta_{Q_i}$$

and

$$\max_{\text{neighboring datasets } D, D'} \left\|Q_{j_{t,m_t}}(D) - Q_{j_{t,m_t}}(D')\right\|_2 = \max_{\text{neighboring datasets } D, D'} \left\|Q_{j_{t,0}}(D) - Q_{j_{t,0}}(D')\right\|_2 = \Delta(\text{type-}t),$$

we use (6) to have $\max_{\text{neighboring datasets } D, D'} G(D, D')$ given by (7).

Finally, from Lemma 2, the total privacy cost of our Algorithm 1 can be given by $(\epsilon_{\text{ours}}, \delta_{\text{budget}})$-differential privacy for ϵ_{ours} satisfying

$$\left[\text{Gaussian}\left(1, \epsilon_{\text{ours}}, \delta_{\text{budget}}\right)\right]^{-2} = \max_{\text{neighboring datasets } D, D'} G(D, D')$$

or (ϵ, δ)-differential privacy for any ϵ and δ satisfying $[\,\text{Gaussian}(1, \epsilon, \delta)]^{-2} = \max_{\text{neighboring datasets } D, D'} G(D, D')$.

Appendix E
Utility of the Gaussian Mechanism

Proof: The noisy response for 1-D query Q_m is $\widetilde{Q}_m(D) = Q_m(D) + N(0, \sigma^2)$. Letting the probability of $\|\widetilde{Q}_m(D) - Q_m(D)\|_p \le \alpha$ be $1 - \beta$, then we have

$$\begin{aligned} 1 - \beta &= \mathbb{P}\left[\left\|\widetilde{Q}_m(D) - Q(D)\right\|_p \le \alpha\right] \\ &= \mathbb{P}\left[\left|N\left(0, \sigma^2\right)\right| \le \alpha\right] \\ &= \mathbb{P}\left[-\alpha \le N\left(0, \sigma^2\right) \le \alpha\right] \\ &= \mathbb{P}\left[N\left(0, \sigma^2\right) \le \alpha\right] - \mathbb{P}\left[N\left(0, \sigma^2\right) \le -\alpha\right] \\ &= \frac{1}{2}\left[1 + \text{erf}\left(\frac{\alpha}{\sigma\sqrt{2}}\right)\right] - \frac{1}{2}\left[1 + \text{erf}\left(\frac{-\alpha}{\sigma\sqrt{2}}\right)\right] \\ &= \text{erf}\left(\frac{\alpha}{\sigma\sqrt{2}}\right), \end{aligned} \tag{30}$$

where $\text{erf}(\cdot)$ denotes the error function and the last step of (30) uses the fact that $\text{erf}(\cdot)$ is an odd function.

According to the two-sigma rule of Gaussian distribution [66], which can also be obtained from above equation that 95% values lie within two standard deviations of the mean. Thus, if we set $\alpha = 2\sigma$, $\beta \approx 0.05$. ■

Acknowledgment

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

References

[1] V. Tudor, V. Gulisano, M. Almgren, and M. Papatriantafilou, "BES: Differentially private event aggregation for large-scale IoT-based systems," *Future Gener. Comput. Syst.*, vol. 108, pp. 1241–1257, Jul. 2020.

[2] M. U. Hassan, M. H. Rehmani, and J. Chen, "Privacy preservation in blockchain based IoT systems: Integration issues, prospects, challenges, and future research directions," *Future Gener. Comput. Syst.*, vol. 97, pp. 512–529, Aug. 2019.

[3] J. Xiong *et al.*, "Enhancing privacy and availability for data clustering in intelligent electrical service of IoT," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1530–1540, Apr. 2019.

[4] J. Liu, C. Zhang, and Y. Fang, "EPIC: A differential privacy framework to defend smart homes against Internet traffic analysis," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1206–1217, Apr. 2018.

[5] K. Gai, Y. Wu, L. Zhu, Z. Zhang, and M. Qiu, "Differential privacy-based blockchain for industrial Internet-of-Things," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4156–4165, Jun. 2020.

[6] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Proc. Int. Conf. Theory Appl. Cryptogr. Techn. (EUROCRYPT)*, 2006, pp. 486–503.

[7] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory Cryptogr. Conf. (TCC)*, 2006, pp. 265–284.

[8] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.

[9] C. Dwork and G. N. Rothblum, "Concentrated differential privacy," 2016. [Online]. Available: arXiv:1603.01887.

[10] M. Bun and T. Steinke, "Concentrated differential privacy: Simplifications, extensions, and lower bounds," in *Proc. Theory Cryptogr. Conf. (TCC)*, 2016, pp. 635–658.

[11] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth, "Generalization in adaptive data analysis and holdout reuse," in *Proc. Conf. Neural Inf. Process. Syst. (NIPS)*, 2015, pp. 2341–2349.

[12] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Proc. IEEE Symp. Found. Comput. Sci. (FOCS)*, 2007, pp. 94–103.

[13] M. Abadi *et al.*, "Deep learning with differential privacy," in *Proc. ACM Conf. Comput. Commun. Security (CCS)*, 2016, pp. 308–318.

[14] J. Tang, A. Korolova, X. Bai, X. Wang, and X. Wang, "Privacy loss in Apple's implementation of differential privacy on MacOS 10.12," 2017. [Online]. Available: arXiv:1709.02753.

[15] Ú. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: Randomized aggregatable privacy-preserving ordinal response," in *Proc. ACM Conf. Comput. Commun. Security (CCS)*, 2014, pp. 1054–1067.

[16] B. Ding, J. Kulkarni, and S. Yekhanin, "Collecting telemetry data privately," in *Proc. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 3571–3580.

[17] R. Henry, A. Herzberg, and A. Kate, "Blockchain access privacy: Challenges and directions," *IEEE Security Privacy*, vol. 16, no. 4, pp. 38–45, Jul./Aug. 2018.

[18] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10700–10714, Dec. 2019.

[19] M. Yang, A. Margheri, R. Hu, and V. Sassone, "Differentially private data sharing in a cloud federation with blockchain," *IEEE Cloud Comput.*, vol. 5, no. 6, pp. 69–79, Nov./Dec. 2018.

[20] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[21] Investopedia. *Blockchain*. [Online]. Available: https://www.investopedia.com/terms/b/blockchain.asp

[22] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Zug, Switzerland, Ethereum, Yellow Paper, 2014.

[23] N. Szabo. (1994). *Smart Contracts*. [Online]. Available: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html

[24] V. Buterin, "A next-generation smart contract and decentralized application platform," Zug, Switzerland, Ethereum, White Paper, 2014.

[25] L. M. Han, Y. Zhao, and J. Zhao, "Blockchain-based differential privacy cost management system," 2020. [Online]. Available: arXiv:2006.04693.

[26] P. Kairouz, S. Oh, and P. Viswanath, "The composition theorem for differential privacy," *IEEE Trans. Inf. Theory*, vol. 63, no. 6, pp. 4037–4049, Jun. 2017.

[27] *Ropsten's Official Github Page*. Accessed: Jan. 9, 2019. [Online]. Available: https://github.com/ethereum/ropsten

[28] D. Sánchez, J. Domingo-Ferrer, and S. Martínez, "Improving the utility of differential privacy via univariate microaggregation," in *Proc. Int. Conf. Privacy Stat. Databases*, 2014, pp. 130–142.

[29] N. Wang *et al.*, "Collecting and analyzing multidimensional data with local differential privacy," in *Proc. IEEE 35th Int. Conf. Data Eng. (ICDE)*, 2019, pp. 638–649.

[30] *Ganache*. Accessed: 2020. [Online]. Available: https://www.trufflesuite.com/ganache

[31] A. Gervais, G. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS)*, 2016, pp. 3–16.

[32] B. Cao *et al.*, "When Internet of Things meets blockchain: Challenges in distributed consensus," *IEEE Netw.*, vol. 33, no. 6, pp. 133–139, Nov./Dec. 2019.

[33] D. Vujičić, D. Jagodić, and S. Ranđić, "Blockchain technology, bitcoin, and Ethereum: A brief overview," in *Proc. 17th Int. Symp. Infoteh-Jahorina (Infoteh)*, 2018, pp. 1–6.

[34] J. Stark, "Making sense of Ethereum's layer 2 scaling solutions: State channels, plasma, and truebit," 2018. [Online]. Available : https://medium.com/l4-media/making-sense-of-ethereums-layer-2-scaling-solutions-state-channels-plasma-and-truebit-22cb40dcc2f4

[35] H. Tang, Y. Shi, and P. Dong, "Public blockchain evaluation using entropy and TOPSIS," *Expert Syst. Appl.*, vol. 117, pp. 204–210, Mar. 2019.

[36] M. H. Miraz and D. C. Donald, "LApps: Technological, legal and market potentials of blockchain lightning network applications," in *Proc. 3rd Int. Conf. Inf. Syst. Data Min.*, 2019, pp. 185–189.

[37] A. Blum, K. Ligett, and A. Roth, "A learning theory approach to noninteractive database privacy," *J. ACM*, vol. 60, no. 2, pp. 1–25, 2013.

[38] *Apache JMeter*. Accessed: Jan. 20, 2021. [Online]. Available: https://jmeter.apache.org/

[39] P. Zheng, Z. Zheng, X. Luo, X. Chen, and X. Liu, "A detailed and real-time performance monitoring framework for blockchain systems," in *Proc. IEEE/ACM 40th Int. Conf. Softw. Eng. Softw. Eng. Pract. Track (ICSE-SEIP)*, 2018, pp. 134–143.

[40] S. Morishima and H. Matsutani, "Accelerating blockchain search of full nodes using GPUs," in *Proc. 26th Euromicro Int. Conf. Parallel Distrib. Netw. Based Process. (PDP)*, 2018, pp. 244–248.

[41] *Activity Monitor User Guide*. Accessed: Feb. 2, 2021. [Online]. Available: https://support.apple.com/en-sg/guide/activity-monitor/welcome/mac

[42] S. King and S. Nadal, "PPCoin: Peer-to-peer crypto-currency with proof-of-stake," *Self-Published Paper*, vol. 19, p. 1, Aug. 2012.

[43] X. Li, H. Li, H. Yan, Z. Cheng, W. Sun, and H. Zhu, "Mitigating query-flooding parameter duplication attack on regression models with high-dimensional Gaussian mechanism," 2020. [Online]. Available: arXiv:2002.02061.

[44] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled Internet of vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2906–2920, Mar. 2019.

[45] J. Kang *et al.*, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4660–4670, Jun. 2019.

[46] Y. C. Tsai, R. Tso, Z.-Y. Liu, and K. Chen, "An improved non-interactive zero-knowledge range proof for decentralized applications," in *Proc. IEEE Int. Conf. Decentralized Appl. Infrastruct. (DAPPCON)*, 2019, pp. 129–134.

[47] A. Fernández Anta, C. Georgiou, and N. Nicolaou, "Atomic appends: Selling cars and coordinating armies with multiple distributed ledgers," 2019.

[48] Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, "A survey on privacy protection in blockchain system," *J. Netw. Comput. Appl.*, vol. 126, pp. 45–58, Jan. 2019.

[49] G. Zyskind, O. Nathan, and A. S. Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *Proc. IEEE Security Privacy Workshops (SPW)*, 2015, pp. 180–184.

[50] Q. Xia, E. B. Sifah, A. Smahi, S. Amofa, and X. Zhang, "BBDS: Blockchain-based data sharing for electronic medical records in cloud environments," *Information*, vol. 8, no. 2, p. 44, 2017.

[51] Y. Lu, Q. Tang, and G. Wang, "ZebraLancer: Private and anonymous crowdsourcing system atop open Blockchain," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2018, pp. 853–865.

[52] S. Hu, C. Cai, Q. Wang, C. Wang, X. Luo, and K. Ren, "Searching an encrypted cloud meets blockchain: A decentralized, reliable and fair realization," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, 2018, pp. 792–800.

[53] M. Luongo and C. Pon, "The keep network: A privacy layer for public blockchains," Keep Netw., Rep., 2018. [Online]. Available: https://keep.network/whitepaper

[54] J. Herrera-Joancomartí and C. Pérez-Solà, "Privacy in bitcoin transactions: New challenges from blockchain scalability solutions," in *Modeling Decisions for Artificial Intelligence*. Cham, Switzerland: Springer, 2016, pp. 26–44.

[55] A. S. Sani *et al.*, "Xyreum: A high-performance and scalable blockchain for IIoT security and privacy," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2019, pp. 1920–1930.

[56] X. Xiao, G. Bender, M. Hay, and J. Gehrke, "iReduct: Differential privacy with reduced relative errors," in *Proc. ACM Int. Conf. Manag. Data (SIGMOD)*, 2011, pp. 229–240.

[57] C. Li and G. Miklau, "An adaptive mechanism for accurate query answering under differential privacy," *Proc. VLDB Endowment*, vol. 5, no. 6, pp. 514–525, 2012.

[58] G. Kellaris and S. Papadopoulos, "Practical differential privacy via grouping and smoothing," *Proc. VLDB Endowment*, vol. 6, no. 5, pp. 301–312, 2013.

[59] G. Yaroslavtsev, G. Cormode, C. M. Procopiuc, and D. Srivastava, "Accurate and efficient private release of datacubes and contingency tables," in *Proc. Int. Conf. Data Eng. (ICDE)*, 2013, pp. 745–756.

[60] F. McSherry and R. Mahajan, "Differentially-private network trace analysis," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, pp. 123–134, 2010.

[61] P. Aaby, J. M. De Acuna, R. Macfarlane, and W. J. Buchanan, "Privacy parameter variation using RAPPOR on a malware dataset," in *Proc. 17th IEEE Int. Conf. Trust Security Privacy Comput. Commun. 12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, 2018, pp. 938–945.

[62] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *Proc. IEEE Symp. Security Privacy (SP)*, 2016, pp. 839–858.

[63] *Proof of Authority*. Accessed: 2020. [Online]. Available: https://github.com/paritytech/parity/wiki/Proof-of-Authority-Chains

[64] A. Agarwal, M. Herlihy, S. Kamara, and T. Moataz, "Encrypted databases for differential privacy," *Proc. Privacy Enhanc. Technol.*, vol. 2019, no. 3, pp. 170–190, 2019.

[65] B. Balle and Y.-X. Wang, "Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 394–403.

[66] F. Pukelsheim, "The three sigma rule," *Amer. Stat.*, vol. 48, no. 2, pp. 88–91, 1994.

Yang Zhao (Graduate Student Member, IEEE) received the master's degree in electrical engineering from the National University of Singapore, Singapore, in 2015. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Nanyang Technological University, Singapore.

Her research interests include federated learning, blockchain, differential privacy, and 6G.

Jun Zhao (Member, IEEE) received the bachelor's degree from Shanghai Jiao Tong University, Shanghai, China, in 2010, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, (advisors: V. Gligor, O. Yagan; collaborator: A. Perrig), affiliating with CMU's renowned CyLab Security and Privacy Institute in 2015.

Before joining NTU, first as a Postdoctoral Fellow with Xiaokui Xiao and then as a Faculty Member, he was a Postdoctoral Fellow with Arizona State University, Tempe, AZ, USA, as an Arizona Computing Postdoctoral Best Practices Fellow (advisors: J. Zhang, V. Poor). He is currently an Assistant Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include communications, networks, security, and AI.

Jiawen Kang received the M.S. and Ph.D. degrees from Guangdong University of Technology, Guangzhou, China, in 2015 and 2018, respectively.

He is currently a Postdoctoral Fellow with Nanyang Technological University, Singapore. His research interests mainly focus on blockchain, security and privacy protection in wireless communications, and networking.

Zehang Zhang received the M.S. degree from Guangdong University of Technology, Guangzhou, China, in 2019.

He is currently a Research Assistant with Nanyang Technological University, Singapore. His research interests mainly focus on blockchain, Internet data security, and privacy protection.

Dusit Niyato (Fellow, IEEE) received the B.Eng. degree from the King Mongkuts Institute of Technology Ladkrabang, Bangkok, Thailand, in 1999, and the Ph.D. degree in electrical and computer engineering from the University of Manitoba, Winnipeg, MB, Canada, in 2008.

He is currently a Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests are in the area of energy harvesting for wireless communication, Internet of Things, and sensor networks.

Shuyu Shi (Member, IEEE) received the B.E. degree from the University of Science and Technology of China, Hefei, China, in 2011, and the Ph.D. degree from SOKENDAI, Hayama, Japan, in 2011.

She is currently a Research Associate Professor with the Department of Computer Science, Nanjing University, Nanjing, China. She was a Research Fellow with Wireless and Networked Distributed Sensing System Group in Parallel and Distributed Computing Center, School of Computer Science and Engineering, Nanyang Technological University, Singapore, from 2016 to 2018. She was with National Institute of Informatics and Department of Informatics, SOKENDAI. She was also a JSPS Research Fellow from April 2015 to October 2016. Her research interests focus on mobile and ubiquitous computing.

Kwok-Yan Lam (Senior Member, IEEE) received the B.Sc. degree (First Class Hons.) in computer science from the University of London, London, U.K., in 1987, and the Ph.D. degree from the University of Cambridge, Cambridge, U.K., in 1990.

He was a Professor with Tsinghua University, Beijing, China, from 2002 to 2010. He has been a Faculty Member with the National University of Singapore, Singapore, and the University of London since 1990. He is currently a Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He was a Visiting Scientist with the Isaac Newton Institute, Cambridge University, Cambridge, and a Visiting Professor with the European Institute, Washington, DC, USA, for systems security. His research interests include distributed systems, IoT security infrastructure, distributed protocols for blockchain, biometric cryptography, homeland security, and cybersecurity.

Prof. Lam received the Singapore Foundation Award from the Japanese Chamber of Commerce and Industry in recognition of his Research and Development achievement in Information Security, Singapore, in 1998.