Outsourcing High-dimensional Healthcare Data to Cloud with Personalized Privacy Preservation

Wei Wang[a,b,*], Lei Chen[b,c], Qian Zhang[b,c]

[a] *School of Electronic Information and Communications*
Huazhong University of Science and Technology
[b] *Fok Ying Tung Research Institute*
[c] *Department of Computer Science and Engineering*
Hong Kong University of Science and Technology, Hong Kong

Abstract

According to the recent rule released by Health and Human Services (HHS), healthcare data can be outsourced to cloud computing services for medical studies. A major concern about outsourcing healthcare data is its associated privacy issues. However, previous solutions have focused on cryptographic techniques which introduce significant cost when applied to healthcare data with high-dimensional sensitive attributes. To address these challenges, we propose a privacy-preserving framework to transit insensitive data to commercial public cloud and the rest to trusted private cloud. Under the framework, we design two protocols to provide personalized privacy protections and defend against potential collusion between the public cloud service provider and the data users. We derive provable privacy guarantees and bounded data distortion to validate the proposed protocols. Extensive experiments over real-world datasets are conducted to demonstrate that the proposed protocols maintain high usability and scale well to large datasets.

Keywords: Healthcare Data, Privacy, Hybrid Cloud

[*] Corresponding author. Tel.: +852 23588766
Email address: gswwang@cse.ust.hk (Wei Wang)

1. Introduction

Gaining access to healthcare data is a vital requirement for medical practitioners and pharmaceutical researchers to study characteristics of diseases. In recent years, the proliferation of cloud computing services enables hospitals and institutions to transit their healthcare data to the cloud, which provides ubiquitous data access and on-demand high quality services at a low cost. On January 25, 2013, the U.S. Department of Health and Human Services (HHS) released the Omnibus Rule [1], which defines cloud service providers (CSPs) as business associates for healthcare data. Currently, many CSPs, including Box, Microsoft, Verizon and Dell, have announced their support for this business associate agreement.

Despite the benefits of healthcare cloud services, the associated privacy issues are widely concerned by individuals and governments [1, 2]. Privacy risks rise when outsourcing personal healthcare records to cloud due to the sensitive nature of health information and the social and legal implications for its disclosure. A natural method is to encrypt healthcare data before transiting them to cloud [3, 4, 5]. However, processing encrypted data is not efficient and is limited to specific operations, and thus is not suitable for healthcare data with versatile usages. An alternative solution is applying existing privacy-preserving data publishing (PPDP) techniques, such as partition-based anonymization [6, 7, 8, 9], and differential privacy [10, 11, 12, 13], to the outsourced healthcare data. However, as we show below, when the following practical requirements are considered, the existing works are not applicable in the context of healthcare data outsourcing.

- **High-dimensional sensitive healthcare attributes.** In real-world s-cenario, hospitals and healthcare institutions often collect and maintain many different healthcare attributes (e.g., blood pressure, heart rate) of their patients. We investigate two real-life healthcare datasets owned by an anonymous hospital in Shenzhen, China, and both contain more than

100 attributes. However, due to limited access of high-dimensional healthcare data, most previous PPDP works have focused on low-dimensional datasets, while the case of high-dimensional data has been overlooked. Partition-based anonymization techniques [6, 7, 8, 9] usually assume that the data only contains a single sensitive attribute, or support only low-dimensional data due to the *curse of dimensionality.* Differential privacy algorithms [10, 11, 12, 13] are designed for data with limited dimensions of sensitive attributes. In the case of high-dimensional sensitive data, differential privacy techniques will inject a huge amount a of noise to results, thus, makes the results useless.

- **Personalized protection at attribute level.** Different individuals may have different privacy preferences. For example, some individuals are sensitive about their blood related records, while others may care about skin related records. Existing personalized protection techniques have focused on personalized access control (e.g., attribute-based encryption [14]) or personalized sensitivities of a single dimension [8], while none has investigated personalized sensitivities over multiple data dimensions.

- **Collusion resistance.** In practice, the outsourced CSP and the data users (DUs), e.g., medical practitioners and pharmaceutical researchers, may collude together due to various incentives [4, 15]. Under such collusion, the whole dataset stored in the cloud as well as the adopted privacy-preserving scheme will be disclosed. Nonetheless, most existing PPDP approaches [16, 6, 7, 8, 10, 11, 12, 13] do not consider this collusion.

To satisfy the above practical requirements, we propose a privacy-preserving framework to outsource healthcare data to a *hybrid cloud.* The hybrid cloud [17] consists of a private cloud that keeps sensitive data within hospitals or institutions, and a public commercial cloud that handles the rest of the dataset. Based on this type of cloud, the proposed framework moves the attributes that are insensitive to any individual to the public cloud while keeping the rest on the private cloud. To answer DUs' queries, the private cloud sends sanitized data to the public cloud to compute results.

In this framework, we propose an optimal sanitization protocol to achieve personalized privacy protections for high-dimensional sensitive data with minimal data distortion. To support high-dimensional data, we exploit the merits of both partition-based anonymization and differential privacy. Instead of directly enforcing differential privacy conditions on each sensitive attribute, we first inject differential privacy in the process of partitioning, and then provide attribute-level protection via partition-based anonymization. As such, the randomness injected to ensure differential privacy is only related to the partition process, which is independent of attribute dimensionality. To achieve such hybrid privacy protection, a partition algorithm with minimal distortion is proposed. In the sanitization protocol, the clouds perform data partition to anonymize the whole dataset based on the amount of distortion required for privacy preservation, which is computed on the private cloud. As such, the public cloud does not have access to the sensitive data stored on the private cloud, but obtains the optimal partition results based on the distortion information. To resist collusion, partition selections are randomized to prevent the public CSP and the DUs from gaining extra knowledge from the partition results. Furthermore, we also propose a greedy protocol to reduce the computational cost. We validate the proposed protocols by formal privacy and usability analyses and evaluate their performance using real-world healthcare datasets.

The main contributions of this paper are threefold. First, we propose a privacy-preserving framework for high-dimensional healthcare data outsourcing. To the best of our knowledge, this is the first framework considering high-dimensional sensitive attributes and personalized privacy requirements over d-ifferent attributes. Second, through formal analytic study, we derive provable privacy guarantees and bounded data distortion achieved by the proposed framework. We show that the proposed framework can defend against the collusion between the public cloud and the DUs while still retaining high usability. Final-ly, for the first time, we conduct experiments on real-world healthcare datasets with high-dimensional sensitive attributes to validate the proposed framework.

The rest of the paper is organized as follows. Section 3 defines the problem. Section 2 reviews related work. Section 4 overviews the privacy-preserving outsourcing framework and two sanitization protocols, followed by privacy and usability analysis in Section 5. Experimental evaluations are reported in Section 6. Section 7 concludes the paper.

2. Related Work

Previous works on privacy-preserving data outsourcing mainly adopt encryption techniques to protect sensitive data [4, 18, 3, 5]. Yuan et al. [4] encrypt the biometric database before outsourcing it to the cloud, which can perform kNN search in the encrypted database. Li et al. [18] leverage Hierarchical Predicate Encryption to establish a scalable framework for authorized private keyword search on cloud data. Cao et al. [3] enables privacy-preserving multi-keyword ranked search over encrypted cloud data. Nonetheless, these solutions are limited to specific operations, which is not suitable for healthcare data outsourcing that supports a variety of queries. Besides, encryption leads to large overhead when answering queries.

Another brand of privacy-preserving approaches are PPDP techniques. Basically, the works on privacy protection in data publishing can be divided into two categories, partition-based approaches and differential privacy. Many partition-based privacy models are proposed to tackle different privacy concerns. k-anonymity [19] is developed to prevent adversaries with Qausi-Identifier (QI) background knowledge from re-identifying an individual with a probability higher than $\frac{1}{k}$. Fragmentation is used in [16] to break sensitive associations among attributes. Other privacy models consider the privacy attack where adversaries associate an individual with a particular sensitive value. ℓ-diversity [20, 6] aims to bound this inference confidence to be no larger than $\frac{1}{\ell}$. (α, k)-anonymity [21] further enhances privacy protection by combining the privacy requirements of k-anonymity and ℓ-diversity. Apart from the QI background knowledge, some works make different adversary assumptions. LKC-privacy [7] is proposed to solve the high dimension problem in healthcare data. LKC-privacy makes an assumption that the background knowledge of adversaries is bounded by L values of QI attributes to limit the power of adversaries. Wong et al. [22] assumes that adversaries know the anonymization algorithm. This extra knowledge can help adversaries breach the privacy, which is called minimality attack. The notion of personalized privacy is proposed by Xiao et al. [8] to allow each individual to specify his/her own privacy preference on a single sensitive attribute. This model makes assumptions that the sensitive attribute has a taxonomy tree and each individual specifies a guarding node in the taxonomy tree as his/her privacy preference. The privacy is violated if the inference confidence on any sensitive value in the subtree of individual's guarding node is higher than the pre-defined threshold. Xiao et al.'s approach specifies different privacy level on a single sensitive attribute, while our work consider individuals' privacy preferences at attribute level. Nevertheless, none of these approaches are designed for high-dimensional data outsourcing with collusion resistance, in which case they lack the consideration of multiple sensitive attributes and personalized privacy concerns at the attribute level.

Recently differential privacy has gained considerable attention as a substitute for partition-based approaches. Numerous approaches [10, 23, 12, 13] are proposed for enforcing ϵ-differential privacy in data publishing. Several works [24, 11, 25] try to handle the multi-dimension issue in differential privacy. Ding et al. [25] present a general framework to release multi-dimensional data cubes by optimally selecting part of a data cube for publication. Peng et al. [24] and Cormode et al. [11] introduce differentially private indices to reduce errors on multi-dimensional datasets. However, the multi-dimensional issues discussed in differential privacy are usually limited to be less than 20, while datasets with higher dimension are not covered. Besides, none of these approaches are designed for data outsourcing with collusion resistance.

Hybrid cloud is a new framework proposed for secure cloud computing. *Sedic* [26] modifies MapReduce's file system to move sanitized data to the public cloud

Table 1: Notational conventions

$\mathcal{A}$	Privacy mechanism
D, D'	Two neighboring datasets
$\tilde{D}$	The anonymized version of D
I	Insensitive attribute
$\lvert I \rvert$	Dimension of insensitive attributes
H	Healthcare attribute
$\lvert H \rvert$	Dimension of healthcare attributes
$\hat{D}$	a partial dataset of D
K	The number of QI partitions
α	Inference confidence threshold
ϵ	Differential privacy parameter
ϵ_k	Differential privacy parameter allocated to the kth partition operation
d_k	The kth partition
d_k^i	A candidate for the kth partition
$E(k, D)$	Minimal distortion of a dataset D with k partitions
$Error(d_k^i)$	Distortion of a partition d_k^i
$E^{d_k^i}(k, D)$	Minimal distortion of a dataset D with k partitions, and the kth partition is set to be d_k^i

and keep sensitive data on the private cloud. Privacy-aware data retrieval on hybrid cloud is investigated in [27]. Different from these works, we consider the salient features of real healthcare data, and provide privacy protection against collusion between the public CSP and the DUs.

3. Problem Definition

3.1. System Model

The notational conventions used in this paper are summarized in Table 1.

System Architecture. We consider the scenario where a hospital needs to transit its healthcare data to the cloud to provide ubiquitous data access and services at low cost. To provide privacy protection of individuals' data, the hospital outsources the healthcare data to a hybrid cloud, which consists of a private cloud that keeps sensitive data within the hospital and a public commercial cloud that handles the rest of the dataset. Fig. 1 illustrates the healthcare outsourcing architecture, where a data holder (e.g., a hospital) outsources a healthcare dataset D to the hybrid cloud, and authorized DUs (e.g., medical practitioners and pharmaceutical researchers) to gain access to the healthcare data from the cloud for medical data analysis. In particular, the data holder first splits the original dataset into insensitive and sensitive parts, and outsources them to the private cloud and public cloud, respectively. Then, to provide privacy protection, a series of operations are performed at the private and public clouds to sanitize the dataset before it can be accessed by DUs. After sanitization, authorized DUs can post queries on the cloud for data analysis.

Healthcare Dataset. The healthcare dataset D contains records of multiple healthcare attributes $\{ID, I^1, ..., I^{|I|}, H^1, ..., H^{|H|}\}$ over different individuals $\{t_1, ..., t_n\}$. ID is the explicit identifier, which should be replaced by pseudo ID (PID). $I = \{I^1, ..., I^{|I|}\}$ is comprised of attributes that are insensitive to all individuals and $H = \{H^1, ..., H^{|H|}\}$ is the union of all individuals' sensitive attributes. Note that I normally consists of non-healthcare attributes such as gender and age that can be obtained by adversaries from other sources like online social networks [6, 7]. As such, the common insensitive attributes are in I while all the differences in personal privacy requirements are left in H. The dataset is divided into $\{PID, H^1, ..., H^{|H|}\}$ and $\{PID, I^1, ..., I^{|I|}\}$, which are outsourced to the private cloud and the public cloud, respectively. The authorized DUs post queries to the private cloud, and the private cloud communicates with the public cloud to generate results. To preserve individual's privacy, the information shared with the public cloud should be carefully sanitized.

Personal Privacy Requirements. An individual has personal privacy requirements on his/her healthcare information. To present personal privacy requirements, *sensitive set* is defined.

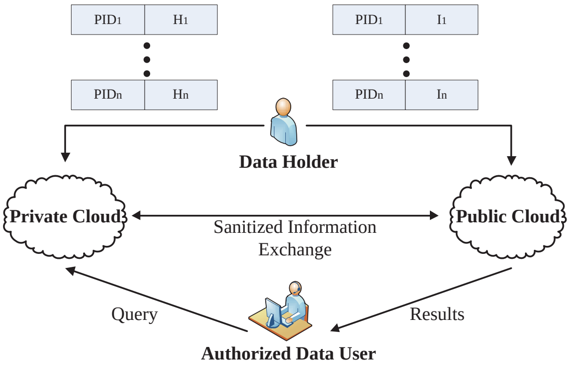


Figure 1: System architecture for healthcare hybrid cloud

Definition 1 (Sensitive Set)**.** *For a tuple $t_i \in D$, its sensitive set $\mathcal{S}_i$ is a subset of H_i, where $H_i \triangleq \{H_i^1, ..., H_i^{|H|}\}$ is the set of all H attributes of t_i.*

$\mathcal{S}_i$ is a subset of healthcare attributes that is private to t_i, while other healthcare attributes $H_i \backslash \mathcal{S}_i$ are non-sensitive and can be published directly. Different tuples (i.e., individuals) in D can specify different sensitive sets.

Threat Model. The CSP that owns the public cloud, referred to as the public CSP, is considered as *honest-but-curious*, that is, the public CSP correctly follows the protocol, yet attempts to learn private information from its received data. Moreover, the DUs may collude with the public CSP to compromise individual's privacy. Note that the private cloud is owned by the data holder and is considered to be trustworthy.

Specifically, the public CSP knows $\mathcal{I}_i = \{I_i^1, ..., I_i^{|I|}\}$ of a target individual t_i, and wants to infer t_i's sensitive values $\mathcal{S}_i$. Based on $\mathcal{I}_i$ and additional information received from the private cloud as well as the DUs, the public CSP can infer the real value r_i^j of t_i's sensitive attribute $S_i^j \in \mathcal{S}_i$ with confidence $\Pr[S_i^j = r_i^j]$.

3.2. Privacy and Data Distortion Measures

Privacy Measure. To quantify the strength of privacy protection, we adopt the most popular privacy models: *confidence bound* [6] and *ϵ-differential* *privacy* [10]. In order to protect the personal sensitive values, anonymization on attributes should be used. To anonymize a dataset D, a *generalization* function is used to map D to a *partition set*, which is defined as follows.

Definition 2 (Partition set)**.** *A partition set* $\{d_1, ..., d_K\}$ *for a dataset* D *is a set of partitions that satisfies the following two conditions: i) a partition set is a set of disjoint partitions such that* $\bigcup_{k=1}^{K} d_k = D$*, and ii) the* I *attribute values of all tuples in a disjoint partition* d_k *are mapped to the same generalized range that has no intersection with other partitions' generalized range.*

For example, if $I = \{age, gender\}$, and there are two tuples $\{PID = 1, age = 18, gender = male\}$ and $\{PID = 2, age = 25, gender = female\}$ in a partition, then generalized range of this partition is $\{PID = \{1, 2\}, age = [18, 25], gender = any\}$.

Based on the notion of partition set, α-confidence bound is defined as follows.

Definition 3 (α-confidence bound)**.** *Let* $\alpha \in [0, 1]$ *be a privacy threshold. We say that a privacy mechanism* $\mathcal{A} : D \to \tilde{D}$ *satisfies* α*-confidence bound if for any* $t_i \in \tilde{D}$ *and any of its sensitive attribute* $S_i^j \in \mathcal{S}_i$*, given* I_i*, an adversary can only infer the real value* r_i^j *of* S_i^j *with confidence* $\Pr[S_i^j = r_i^j] \leq \alpha$*.*

However, according to the notion of *minimality* in anonymization [22], anonymization mechanisms aim to achieve privacy guarantee with minimal data distortion, and this deterministic attempt provides a loophole for attacks. Thus, the anonymized results for queries may leak private information to the DUs. To thwart such privacy breach, differential privacy protection is needed to randomize the anonymization results. The intuition of differential privacy is that the removal or addition of a single record does not significantly affect the outcome of any analysis.

Definition 4 (ϵ-differential privacy)**.** *A randomized mechanism* $\mathcal{A}$ *ensures* ϵ*-differential privacy if for any datasets* D *and* D' *differing on at most one tuple,*

$$\Pr[\mathcal{A}(D) = O] \leq e^{\epsilon} \times \Pr[\mathcal{A}(D') = O], \tag{1}$$

for all $O \in Range(\mathcal{A})$*, where* $Range(\mathcal{A})$ *is the set of possible outputs of* $\mathcal{A}$*.*

Roughly speaking, the parameter ϵ is positive and specified by the data holder. The smaller value of ϵ provides stronger privacy guarantee.

We use a privacy measure, denoted as (α, ϵ)*-privacy*, which combines the above two measures to quantify the strength of privacy protection: i) personalized protection at attribute level: an adversary's inference confidence on a sensitive attribute is bounded by α, and ii) collusion resistance: the partition set is ϵ-differentially private.

Data Distortion Measure. We first define two inference probabilities: the original inference probability $p_{i,j}$ and the approximate inference probability $\tilde{p}_{i,j}$:

- $p_{i,j}$: Before generalization on H, the inference probability of the jth H attribute of a tuple t_i is defined by $p_{i,j} = \frac{v_{i,j}}{n_k}$, where $v_{i,j}$ is the frequency of the real value of H_i^j and n_k is the size of the partition.

- $\tilde{p}_{i,j}$: After generalization on H, the inference probability of the jth H attribute of a tuple t_i is defined by $\tilde{p}_{i,j} = \frac{\tilde{v}_{i,j}}{n_k}$, where $\tilde{v}_{i,j}$ is the frequency of the real value of H_i^j in the partition.

Then, we use *overall sum of error* as the data distortion measure, which is obtained by:

$$Error(D) = \sum_{d_k \in D} \sum_{t_i \in d_k} \sum_{j} |p_{i,j} - \tilde{p}_{i,j}|. \tag{2}$$

3.3. Design Goal

Our goal is to design a privacy-preserving outsourcing framework under the hybrid cloud model. Specifically, the framework has the following objectives: i) The framework should preserve (α, ϵ)-privacy for each personal sensitive set $\mathcal{S}_i$ even when the public CSP and the DUs collude together; ii) The framework should retain the usability of the healthcare data as much as possible by minimizing the data distortion $Error(D)$; iii) The framework should be efficient, that is, the computational and communication costs should be scalable with the size of the dataset and the number of queries.

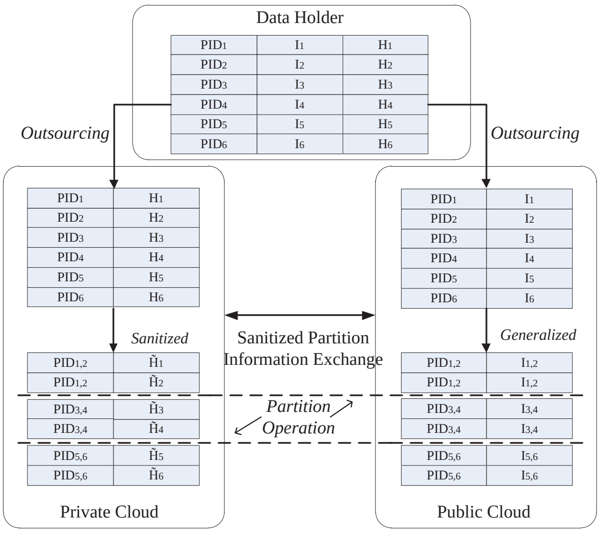


Figure 2: Framework Overview

4. The Privacy-Preserving Outsourcing Framework

4.1. Overview

The core idea of the privacy-preserving outsourcing framework is to share partition strategy between clouds to derive sanitized data while keeping sensitive data on the private cloud. To provide personalized protection on sensitive data, the dataset is divided into multiple partitions and generalization operations are applied on personal sensitive attributes. Partition information is shared between clouds to derive the optimal partition strategy. To ensure ϵ-differential privacy, randomness is injected into each partition selection. Fig. 2 illustrates the framework, where $I_{i,j}$ is the generalized range of I_i, I_j, and $\tilde{H}_i$ is the sanitized version of H_i to protect personal sensitive attributes. As the public CSP is untrusted, the private cloud guides the data partitioning without sharing sensitive data with the public cloud. To achieve this goal, we propose two sanitization protocols, that is, an optimal sanitization protocol with minimal data distortion and a greedy sanitization protocol with higher efficiency. After data partition and generalization, the hybrid cloud answers DUs' queries based on the sanitized data. As such, the DUs can acquire knowledge no more than the sanitized data. Besides, randomization avoids privacy leakage from the partition results. Thus, even when the public CSP and the DUs collude together, the framework can still thwart the privacy breach of the sensitive data. As the sanitized data is in the form of standard anonymized data tables [6, 7, 8], the cloud can easily answer different types of queries.

The protocols are built based on two components. The first component is optimal partitioning, which aims to find a partition set that can satisfy personalized privacy requirements with minimal data distortion. The second component is privacy budget allocation, which optimally allocates different fractions of randomness to each partition operation so that the final partition set is differentially private while the overall data distortion is minimized. The rest of this section elaborates the two components and the sanitization protocols.

4.2. Optimal Partitioning

For a partition set with K partitions, we need $(K-1)$ sequential partition operations. We formulate a dynamic programming problem to find the optimal partition sequence with minimal distortion.

The minimal distortion for a dataset D with k partitions $E^*(k, D)$ is given by:

$$E^*(k, D) = \min_{d_l^i} \sum_{l=1}^{k} Error(d_l^i), \tag{3}$$

where d_l^i is a possible lth partition for D, and $Error(d_l^i)$ is the error for d_l^i computed according to (2).

$E^*(k, D)$ is computed via the following recursive rule:

$$E^*(k, D) = \min_{d_k^i \in D} \left(E^*(k-1, D \backslash d_k^i) + Error(d_k^i) \right), \tag{4}$$

where d_k^i is a possible kth partition d_k, $E^*(k-1, D\backslash d_k^i)$ the minimal error for sanitizing partial dataset $D\backslash d_k^i$ with $k-1$ partitions. Therefore, our problem is to compute $E^*(K, D)$, and keep all intermediate results, i.e. $E^*(k-1, D\backslash d_k^i)$ and $Error(d_k^i)$, for each d_k^i, k.

The complexity of the dynamic programming (4) is $O(|H| \cdot Km^{2|I|}n)$, where n is the number of tuples, m the maximum number of different values in I appearing in D, $|I|$ and $|H|$ the dimensions (i.e., numbers of attributes) of I and H, respectively. In healthcare data, $|I|$ is a small constant that does not grow with the number of tuples n. Thus, the complexity of the dynamic programming problem is polynomial time.

4.3. Privacy Budget Allocation

In order to ensure ϵ-differential privacy, randomness is injected into each partition operation. According to exponential mechanism [23], outputs of higher scores are assigned with exponentially greater probabilities. In our framework, the selection of a partition probability is proportional to $\exp\left(-\frac{\epsilon_k E^{d_k^i}(k,D)}{2\Delta E}\right)$, where ϵ_k is the privacy budget allocated to the kth partition operation, d_k^i a possible kth partition, Δ_E the sensitivity function defined as

$$\Delta E = \max_{\forall k,i,D,d_k^i} |E^{d_k^i}(k,D) - E^{d_k^i}(k,D')|, \tag{5}$$

where d_k^i stands for the ith sample output for the kth partition d_k, k the number of output partitions.

Two parameters ΔE and ϵ_k need to be determined to decide the exponential probability $\exp\left(-\frac{\epsilon_k E(k,d,d_k^i)}{2\Delta E}\right)$. First, we quantify ΔE by the following lemma.

Lemma 1. *Given two neighboring datasets D and D', the difference between the error of the kth partition operation on D and D' is bounded as follows.*

$$|E^{d_k^i}(k,D) - E^{d_k^i}(k,D')| \leq \frac{2|H|}{\alpha}.$$

Proof. Assume that the changed tuple is in partition d_k. n_k denotes the number of tuples in d_k, and $v_{k,j,h}$ denotes the number of jth violated sensitive value h. There are two cases that may change $E^{d_k^i}(k,D)$.

Case I: One tuple is added, and the inference confidence on its sensitive attributes equal to α before the addition. In this case, for each violated H attribute, only one more suppression for the violated value is needed. Then, we derive: $|E^{d_k^i}(k,D) - E^{d_k^i}(k,D')| = \sum_h |((v_{k,j,h}+1) - \lfloor n_i\alpha \rfloor) - (v_{k,j,h} - \lfloor n_i\alpha \rfloor)| = |H|$.

Case II: One tuple is removed, and the removal of its non-sensitive attributes makes sensitive attributes violate the α condition. Similar to Case I, the inference confidence on these violated attributes equal to α before the removal. In this case, only one suppression for each violated sensitive values is needed. This is because $\frac{x-1}{y-1} < \frac{x}{y}, \forall 0 < x < y$. Before removal, for a H attribute, denote the total number of values in the partition by y, the number of a violated value by x. This case can be expressed by $\frac{x}{y} \leq \alpha$ and $\frac{x}{y-1} \geq \alpha$. By one suppression, we can get $\frac{x-1}{y-1} < \frac{x}{y} \leq \alpha$. Then, we derive:

$$\begin{aligned}
&|E^{d_k^i}(k,D) - E^{d_k^i}(k,D')| \\
=&\sum_{j,h} |((v_{k,j,h}+1) - \lfloor n_i\alpha \rfloor) - (v_{k,j,h} - \lfloor n_i\alpha \rfloor)| \\
\leq&|H| \cdot \left\lfloor \frac{n_i}{\lfloor n_i\alpha \rfloor} \right\rfloor \leq |H| \cdot \frac{\lfloor n_i\alpha \rfloor + 1}{\alpha} \cdot \frac{1}{\lfloor n_i\alpha \rfloor} \leq \frac{2|H|}{\alpha}.
\end{aligned}$$

We can see that Case I is no worse than Case II because for each H attribute, at most one sensitive value may violates the α condition, while for Case II there may be multiple. Therefore, the analysis of Case II derives the upper bound in Lemma 1. □

Next, we decide ϵ_k so that the total privacy budget ϵ is carefully allocated to each probabilistic partition operation to minimize the expected error. The expectation of error can be expressed by the optimal error and the expected additional error over the optimal error. We first derive the expected additional error for a single partition operation.

Lemma 2. *Let $\mathbb{E}[\Delta k]$ be the expected additional error over optimal error $E^*(k,D)$ when executing Line 8 in Algorithm 1, i.e., $\mathbb{E}[\Delta k] \triangleq \mathbb{E}[E^{d_k}(k,D) - E^*(k,D)]$.*

We have

$$\mathbb{E}[\Delta k] = \frac{\sum_{d_k^i} \left(E^{d_k^i}(k,D) - E^*(k,D)\right) \exp\left(-\frac{\epsilon_k(E^{d_k^i}(k,D)-E^*(k,D))}{2\Delta E}\right)}{\sum_{d_k^i} \exp\left(-\frac{\epsilon_k(E^{d_k^i}(k,D)-E^*(k,D))}{2\Delta E}\right)}.$$

Due to space limitation, we omit the proof of the above lemma. The problem of error minimization with respect to ϵ can be formulated as follows.

$$\min_{\epsilon_k} \quad \sum_{k=1}^{K-1} \mathbb{E}[\Delta k] \tag{6a}$$

$$\text{subject to} \quad \sum_{k=1}^{K-1} \epsilon_k = \epsilon, \tag{6b}$$

$$\epsilon_k \geq 0, \forall k, \tag{6c}$$

where K is the number of partitions. The constraints ensure the ϵ condition. The objective is to minimize the summation of the expected additional error over the optimal error for each partition, so as to minimize the overall error.

Problem (6) is a convex problem shown by the following theorem.

Theorem 1. *Problem* (6) *is convex, which has an optimal solution and can be solved in time complexity of* $O(K^3)$ *by standard convex solver [28].*

Proof. Denote $E^{d_k^i}(k,D) - E^*(k,D)$, $\epsilon'_k = \frac{\epsilon_k}{2\Delta E}$. We can rewrite $\mathbb{E}[\Delta k]$ as $\mathbb{E}[\triangle k] = \sum_i \triangle E_i \cdot \frac{\exp\{-\epsilon'_k \triangle E_i\}}{\sum_i \exp\{-\epsilon'_k \triangle E_i\}}$. To show the convexity of Problem (6), we need to show that the objective function is twice differentiable, that is, its *Hessian* or second-order partial derivative matrix exists at each point in the domain. Then the problem is convex if and only if its domain is convex and its Hessian matrix is *positive semi-definite* [28]. The domain here is defined by the linear constraints, which is a convex region. Now we study the Hessian of the objective function. The first order derivative is given as:

$$\begin{aligned}&\frac{\partial \sum_{k=1}^{K-1} \mathbb{E}[\triangle k]}{\partial \epsilon'_k} \\ =&\frac{\left(\sum_i \triangle E_i e^{-\epsilon'_k \triangle E_i}\right)^2 - \left(\sum_i \triangle {E_i}^2 e^{-\epsilon'_k \triangle E_i}\right)\left(\sum_i e^{-\epsilon'_k \triangle E_i}\right)}{\left(\sum_i e^{-\epsilon'_k \triangle E_i}\right)^2} \leq 0.\end{aligned}$$

For second-order partial derivative with respect to ϵ_k and ϵ'_k, we have $\frac{\partial^2 \sum_{k=1}^{K-1} \mathbb{E}[\triangle k]}{\partial \epsilon'_k \partial \epsilon'_{k'}} = 0, \forall k, k' \in 1, ..., K-1, k \neq k'$. And for all $k' = k$, we have $\frac{\partial^2 \sum_{k=1}^{K-1} \mathbb{E}[\triangle k]}{\partial {\epsilon'_k}^2} \geq 0$ by using the *Cauchy-Schwarz* inequalities. Note that the equality stands if and only if $\triangle E_i = 1, \forall \triangle E_i$.

Based on the above analyses, it is easy to prove that its Hessian is positive semi-definite. Therefore, we conclude that Problem (6) is convex.

Then, we show the time complexity of Problem (6). Since we can derive the closed-form for the first and second order derivatives, according to [28], convex optimization problems can be solved with time complexity of $O(\log m \cdot \max(n^3, n^2 m))$, where n is the number of variables and m is the number of constraints. In our case, $n = K$ and $m = 2$. Thus, the time complexity for solving our problem is $O(K^3)$. □

The optimal ϵ_k in Problem (6) can be obtained via standard convex optimization tools such as gradient search or simplex methods. Furthermore, notice that the time complexity $O(K^3)$ is a constant given the partition parameter K and does not grow with the size of the input dataset.

4.4. Optimal Sanitization Protocol

After presenting the two core components in designing the sanitization protocols, we show Algorithm 1, which illustrates the optimal sanitization protocol *OptPer*. In this algorithm, dataset D is expressed as the full set of PIDs, and partitions $\{d_k^i\}$ are expressed as sets of PIDs. The kth partitions of H and I are denoted as H_{d_k}, I_{d_k}, respectively. As such, the only information exchange between clouds is the partition strategy expressed as groups of PIDs. OptPer consists of the following two stages.

Initialization (Line 1-3). First, the public could computes all possible partition sets and sends them (groups of PIDs) to the private cloud. For all possible partition sets, the private cloud computes $\{Error(d_k^i)\}$, $\{E^*(k, \hat{D})\}$, where $Error(d_k^i)$ is the distortion for a possible partition d_k^i and $E^*(k, \hat{D})$ is the minimal distortion for a partial dataset $\hat{D}$ with k partitions. The distortions

Algorithm 1 Optimal Sanitization Protocol (OptPer)

```
1: The public cloud computes all possible $k$th partitions $\{d_k^i\}$, and sends $\{d_k^i\}$ to the private cloud;
2: The private cloud computes $\{Error(d_k^i)\}$, $\{E^*(k, \hat{D})\}, \forall k, d_k^i, \hat{D} \subseteq D$ according to (2), (4);
3: The private cloud computes the optimal $\epsilon_k$ for the $k$th partition operation by solving (6);

4: for each $k$ from $K$ to 2 do
5:     for each possible $k$th partition $d_k^i$ do
6:         The private cloud computes $E^{d_k^i}(k, D) = E^*(k-1, D \backslash d_k^i) + Error(d_k^i)$;
7:     end for
8:     The private cloud selects $d_k \leftarrow d_k^i$ with probability $\propto \exp\left(-\frac{\epsilon_k E^{d_k^i}(k,D)}{2\triangle E}\right)$, and updates $D$: $D \leftarrow D \backslash d_k$;
9:     The private cloud suppresses minimal number of sensitive values in $H_{d_k}$ to satisfy personalized privacy requirements;
10:    The private cloud sends $d_k$ to the public cloud for its generalization on on $I_{d_k}$;
11: end for
```

are computed according to (2), (4). Based on distortions, the private cloud allocates the privacy budget ϵ_k to each partition operation by solving Problem (6).

Sequential Partitioning (Line 4-11). After computing the errors and privacy budget of partition operations, the private cloud decides the partitions one by one. The private cloud randomizes each partition operation via exponential mechanism, where the selection probability of a partition operation is proportional to $\exp\left(-\frac{\epsilon_k E^{d_k^i}(k,D)}{2\Delta E}\right)$ (Line 8). After selecting the kth partition, the private cloud shares this partition strategy with the public cloud to perform generalization and suppression.

In the protocol, we generalize the insensitive attribute in a partition to the same generalized range (Line 10). If a partition still violates the privacy requirements by Definition 3, we remove the violation by suppressing some sensitive values from the partition. The suppression scheme is to remove the minimal number of the violating sensitive values that satisfies the privacy requirements. The generalization and suppression operations generalize values to a coarser

Algorithm 2 Greedy Sanitization Protocol (GrePer)

```
1: Initialize $D$ as a single partition;
2: repeat
3:     The public cloud enumerates all possible partition operation that divides an existing partition, and sends them to the private cloud;
4:     for each possible partition operation $p_i$ do
5:         The private cloud computes $E(p_i)$: the partition distortion if applying $p_i$;
6:     end for
7:     The private cloud selects $p_i$ with probability $\propto \exp\left(-\frac{\epsilon_k E(D,p_i)}{2(K-1)\Delta E}\right)$, and updates $D$;
8: until There are $K$ partitions
9: The private cloud suppresses minimal number of sensitive values in $H$ to satisfy personalized privacy requirements;
10: The private cloud sends the selected partition strategy to the public cloud for its generalization on on $I$;
```

range that is consistent with the original values.

4.5. Greedy Sanitization Protocol

OptPer achieves minimal data distortion and has good scalability in the case of large dataset with high-dimensional sensitive attributes. Specifically, OptPer has linear complexity with respect to the dimension of senstive attributes and number of users. Detailed complexity analysis can be found in Section 5.2.2. However, as for data with high-dimensional insensitive attributes, the dynamic programming adopted in OptPer still requires relatively high computational costs and communication overhead. To cope with this case, we propose a more efficient algorithm called *GrePer*, as described in Algorithm 2. Instead of using dynamic programming in OptPer, we greedily choose a single dimensional partition operation p_s based on the distortion after applying the operation. The computation for partition error is the same as OptPer, where minimal suppressions on sensitive values in each partition is applied to satisfy personalized privacy requirements. The detailed costs analysis is provided in the next section.

As described in Algorithm 2, GrePer iteratively applies a selected partition operation until the dataset is cut into K partitions (Line 2-8). In each iteration, GrePer computes the overall distortion of each possible cut, and selects a cut with a probability proportional to $\exp\left(-\frac{\epsilon_k E(d_j, p_s)}{2(K-1)\Delta E}\right)$. Similar to Algorithm 1, personalized privacy protections are guaranteed by generalization and suppression (Line 9,10).

4.6. Handling New Healthcare Data

Note that there may be new healthcare records generated in hospital over time. In this case, hospitals need periodically (e.g., annually or quarterly) outsource the new records to update the dataset in the cloud. Our framework treats the new records in one period as an independent dataset, and sanitizes the dataset to ensure the same (α, ϵ)-privacy. Note that there are no modifications to the old records that have been outsourced to the cloud. As there is no overlap between the check-in timestamps of the new data and the old data, the partitions of the new dataset and the old dataset in the cloud are disjoint. Therefore, the confidence bounding condition of the old dataset in the cloud is not affected by the newly-outsourced dataset. Based on the parallel composition theory [29], the computations operate on disjoint subsets of the dataset, the overall privacy guarantee depends only on the worst of the guarantees of each computation. If we apply partitioning independently to disjoint datasets with the same ϵ-differential privacy guarantee, the overall dataset paritioning still preserves ϵ-differential privacy. Thus, the overall dataset outsourced to the cloud preserves (α, ϵ)-privacy.

5. Privacy and Usability Analysis

5.1. Privacy Analysis

To prove the privacy validity of the protocols, we first show the privacy gained by each partition operation.

Lemma 3. *The selection of the kth partition operation in Algorithm 1 (Line 8) and Algorithm 2 (Line 7) ensures ϵ_k-differential privacy.*

Proof. w.l.o.g., we take the notations in Algorithm 1 for illustration. Given two neighboring datasets D and D', from Theorem 1, we have $|E^{d_k^i}(k, D) -$

$E^{d_k^i}(k, D')| \le \Delta E$. Then, for the kth partition d_k, the probability of selecting d_k^i from D, denoted as $\Pr[d_k \leftarrow d_k^i|D] = \exp\left(-\frac{\epsilon_k E^{d_k^i}(k,D)}{2\Delta E}\right) \Big/ \sum_i \exp\left(-\frac{\epsilon_k E^{d_k^i}(k,D)}{2\Delta E}\right)$, enjoys

$$\Pr[d_k \leftarrow d_k^i|D] \le \frac{\exp\left(-\frac{\epsilon_k E(k,D',d_k^i)-\Delta E}{2\Delta E}\right)}{\sum_i \exp\left(-\frac{\epsilon_k E(k,D',d_k^i)+\Delta E}{2\Delta E}\right)} = e^{\epsilon_k}\Pr[d_k \leftarrow d_k^i|D],$$

which proves this lemma according to Definition 4. □

Based on Lemma 3, we can prove the correctness of the protocols by the following theorem.

Theorem 2. *OptPer and GrePer ensure (α, ϵ)-privacy even when the public CSP and the DUs collude together.*

Proof. As CSPs are honest-but-curious, the sanitization protocols are correctly executed. In both protocols, the sanitized data is generated by the probabilistic partition operations. After partitioning, the protocols resolve the violations of α-confidence bound by suppression and generalization, which guarantee the α condition. The ϵ condition is ensured by the sequence of probabilistic partition operations. The sequential partition operations are conducted on the same dataset. According to sequential composition [29], the partitioning result achieves $\sum_k \epsilon_k$-differential privacy, i.e. ϵ-differential privacy, which satisfies the ϵ condition. As the private cloud answers queries according to sanitized data, there is no extra privacy breach when answering queries posed by the DUs. Thus, the protocols achieve (α, ϵ)-privacy when the public CSP and the DUs collude together. □

Note that the threat model considered in this paper is curious-but-honest, meaning that both CSPs exactly follow the proposed protocols, yet attempts to learn private information. While how to defend malicious CSPs is rather a security issue, which is beyond the scope of this paper.

5.2. Usability Analysis

5.2.1. Data Distortion

To provide accurate answers for queries, OptPer involves minimal data distortion which is bounded by the following theorems. We use $E(K, D)$ to denote the overall distortion of the sanitized data output by OptPer, where K is the number of partitions and D is the original data.

Theorem 3. *OptPer minimizes the expected* $E(K, D)$*.*

Proof. According to the property of expectation, we have

$$\mathbb{E}[E(K,D)] = \sum_{k=1}^{K-1} (\mathbb{E}[\Delta k + E^*(k,D))] = \sum_{k=1}^{K-1} \left(\mathbb{E}[\Delta k] + \sum_{k=1}^{K-1} E^*(k,D) \right). \quad (7)$$

Since $E^*(k, D)$ is constant, and Problem (6) minimizes $\sum_{k=1}^{K-1} \mathbb{E}[\Delta k]$, for all $k \in \{1, ..., K-1\}$. Thus, the expected $E(K, D)$ is minimized. □

Next, we analyze the upper bound of the expected distortion incurred by OptPer.

Theorem 4. *Let* $E^*(K, D)$ *be the minimal distortion without injecting randomness. The expected error of the sanitized data output by OptPer is up-bounded by*

$$E^*(K,D) + \frac{2|H| \cdot \left(e^{\frac{\epsilon}{2}} + K - 2\right)}{\alpha}.$$

Proof. To prove the upper bound of $\mathbb{E}[E(K, D)]$, we first derive the upper bound for the expected error of a partition operation. For any error of a partition operation $E^{d_k^i}(k, D)$, we have $E^{d_k^i}(k, D) - E^*(k, D) \geq 0$ and $E^{d_k^i}(k, D) - E^*(k, D) \leq \Delta E$. Then, based on Lemma 2, the expected error of the kth partition $\mathbb{E}[\Delta k]$ over the optimal error $E^*(k, D)$ is bounded by the following inequalities.

$$\begin{aligned} \mathbb{E}[\Delta k] =& \frac{\sum_i \left(E^{d_k^i}(k,D) - E^*(k,D)\right) \exp\left(-\frac{\epsilon_k (E^{d_k^i}(k,D) - E^*(k,D))}{2\Delta E}\right)}{\sum_i \exp\left(-\frac{\epsilon_k (E^{d_k^i}(k,D) - E^*(k,D))}{2\Delta E}\right)} \\ \leq& \frac{\sum_i \Delta E \exp\left(-\frac{\epsilon_k (E^{d_k^i}(k,D) - E^*(k,D))}{2\Delta E}\right)}{\sum_i \exp\left(-\frac{\epsilon_k}{2}\right)} \leq \Delta E \cdot e^{\frac{\epsilon_k}{2}}. \end{aligned}$$

The following inequalities use the fact that $e^{x_1} + e^{x_2} \leq e^{x_1+x_2} + 1$ for any non-negative real number x_1, x_2.

$$\begin{aligned}\mathbb{E}[E(K,D)] \leq & E^*(K,D) + \Delta E \cdot \left(e^{\frac{\sum_k \epsilon_k}{2}} + K - 2\right) \\ \leq & E^*(K,D) + \frac{2|H| \cdot \left(e^{\frac{\epsilon}{2}} + K - 2\right)}{\alpha}.\end{aligned}$$

□

From the upper bound derived by Theorem 4, we have the following observation:

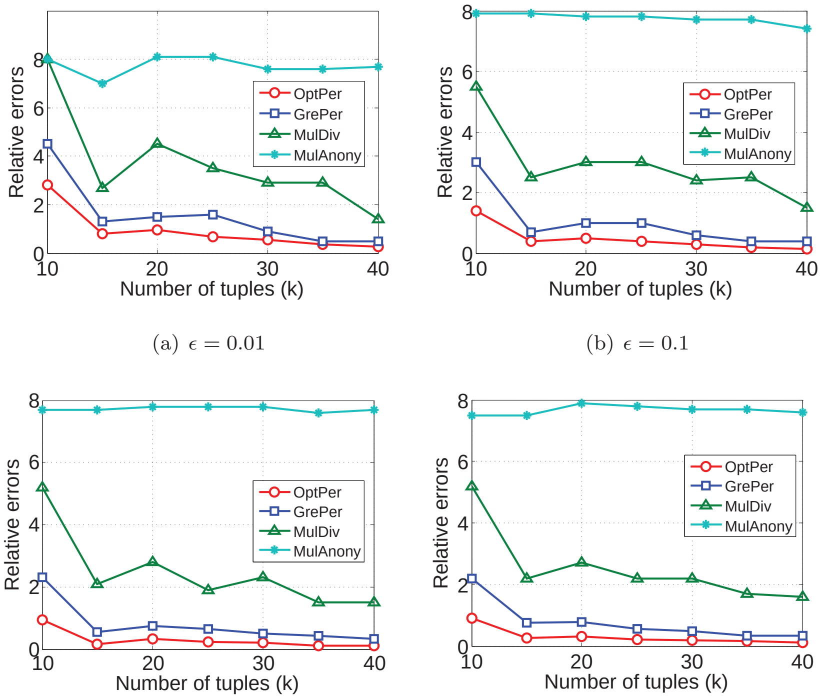
Corollary 1. *The gap between the minimal error without randomness and the error incurred by OptPer remains the same when the number of tuples grows.*

The error gap is caused by randomization in OptPer to guarantee ϵ condition. The constancy property of the error gap stated in Corollary 1 indicates that OptPer maintains good utility scalability on large datasets.

5.2.2. Computational Complexity and Communication Overhead

The overall complexity of OptPer is $O(|H|(1 + Km^{2|I|})n)$. m is up-bounded by the maximum number of different values in I, which is a constant given I and is much smaller than the number of individuals n. In healthcare data, I is rather small while $|H|$ is large, and $|I|$, $|H|$ are constant that do not grow with n. Thus, given all the attributes contained in the data, the complexity is linear with respect to the number of tuples n. The overall complexity of GrePer is $O(|I| \cdot |H| \cdot Kmn)$, which largely reduces computational cost and is scalable with $|I|$, compared with OptPer. Note that these computations are one-time computations that are required for initial outsourcing. Thus, no extra computations are added to answering queries.

The information exchanges between clouds only contain partition strategies, which are quite lightweight. The amounts of communication overhead incurred by OptPer and GrePer are $O(nm)$ and $O(km)$, respectively. To answer queries, the private cloud sends the PIDs that matches queries to the public cloud to



(a) $\epsilon = 0.01$

(b) $\epsilon = 0.1$

(c) $\epsilon = 1$

(d) $\epsilon = 2$

Figure 3: Errors vs. number of tuples (in thousands)

generate results, where the communication overhead is usually much less than $O(n)$.

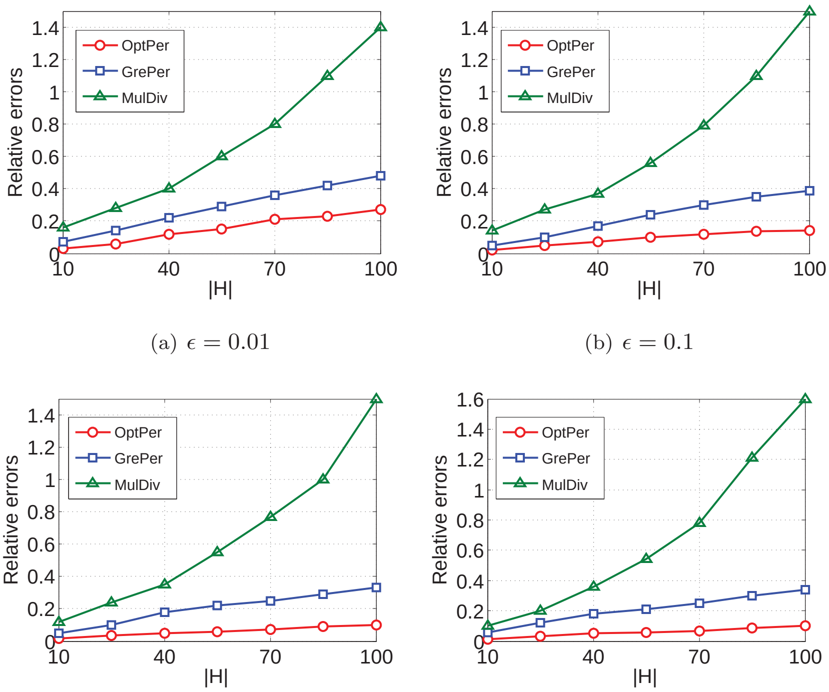
6. Evaluation

In this section, we evaluate the performance of OptPer and GrePer on real-world healthcare datasets.

6.1. Experimental Setup

Hardware. All the experiments were conducted on 3.00 GHz Intel Core 2 E8400 PC with 2GB RAM, and all algorithms were implemented using C++.

Datasets. We employ two real-life healthcare checkup datasets, *Checkuplist1* and *Checkuplist2*, which are owned by an anonymous hospital in Shen-



(a) $\epsilon = 0.01$

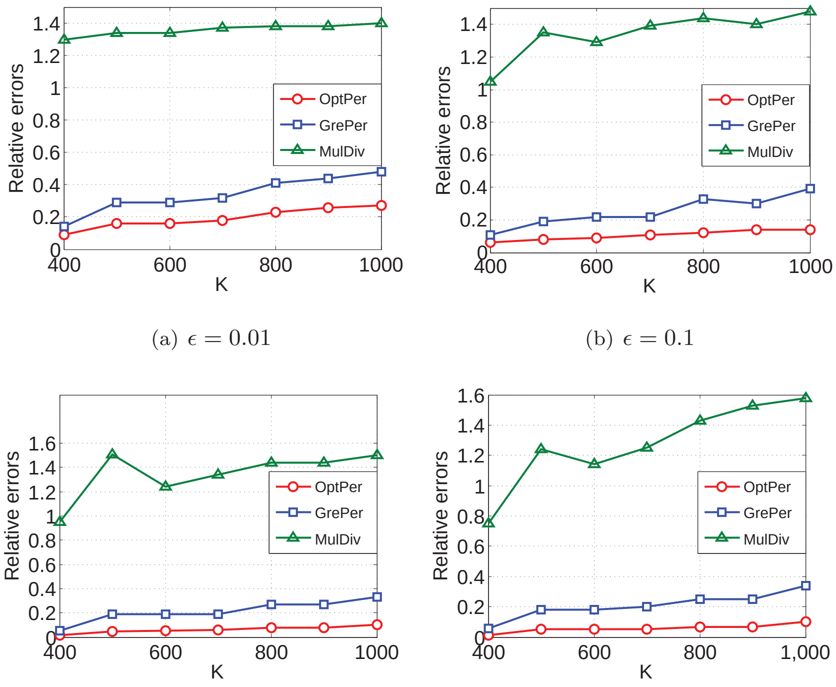
(b) $\epsilon = 0.1$

(c) $\epsilon = 1$

(d) $\epsilon = 2$

Figure 4: Errors vs. the dimension of sensitive attribute

zhen, China. Both datasets contain checkup records of 40,332 individuals from the year 2006 to 2011. Each record contains personal information and all health checkup items of an individual. Normally, personal information is considered insensitive while health checkup items are potentially sensitive [7]. After removing explicit identifiers, Checkuplist1 and Checkuplist2 have 116 and 111 attributes, respectively. In both datasets, personal information includes *age*, *gender* and *checkup time*, which are considered to be the I attributes. Values of I attributes are either numerical (age and checkup time) or categorical (gender). The remaining attributes are H attributes corresponding to different checkup items. The values of the healthcare attributes are the diagnostic results on a certain checkup item, which are categorical.



(a) $\epsilon = 0.01$

(b) $\epsilon = 0.1$

(c) $\epsilon = 1$

(d) $\epsilon = 2$

Figure 5: Errors vs. number of partitions

Personalization settings. We generate personalization settings based on previous personalization privacy protection work [8]. Tuples are randomly divided into three levels: 10% tuples in level 1, 60% tuples in level 2, and 30% tuples in level 3. There are no sensitive attributes for level 1 tuples. For each level 2 tuple, 30% of its H attributes are randomly selected as sensitive attributes, while for each level 3 tuple, 60% of its H attributes are marked as sensitive. The selection of sensitive attributes is independent for each tuple.

Baselines. We compare the performance of our protocols with traditional anonymization [6] and differential privacy [10] approaches. For fair comparison, we extend these approaches to be applicable to high-dimensional dataset. The anonymization approach, denoted as *MulAnony*, first decomposes the dataset into disjoint partial datasets that each partial dataset contains the tuples with the same privacy requirements, and then applies anonymization on each partial dataset independently. The comparison with MulAnony demonstrates the merits of the proposed optimal partition strategy and budget allocation. The differential privacy approach, denoted as *MulDiff*, treats the union of sensitive attributes as sensitive attributes, and applies randomized (differentially private) multi-dimensional partitioning. The comparison between MulDiff and our algorithms shows the benefits of considering personal requirements in the partitioning procedure.

6.2. Utility

To measure the utility of the protocols, we use relative error, which is defined by the value of overall distortion divided by number of tuples. We test all methods with three commonly used values of ϵ: 0.01, 0.1, 1, 2, and α is set to 0.01.

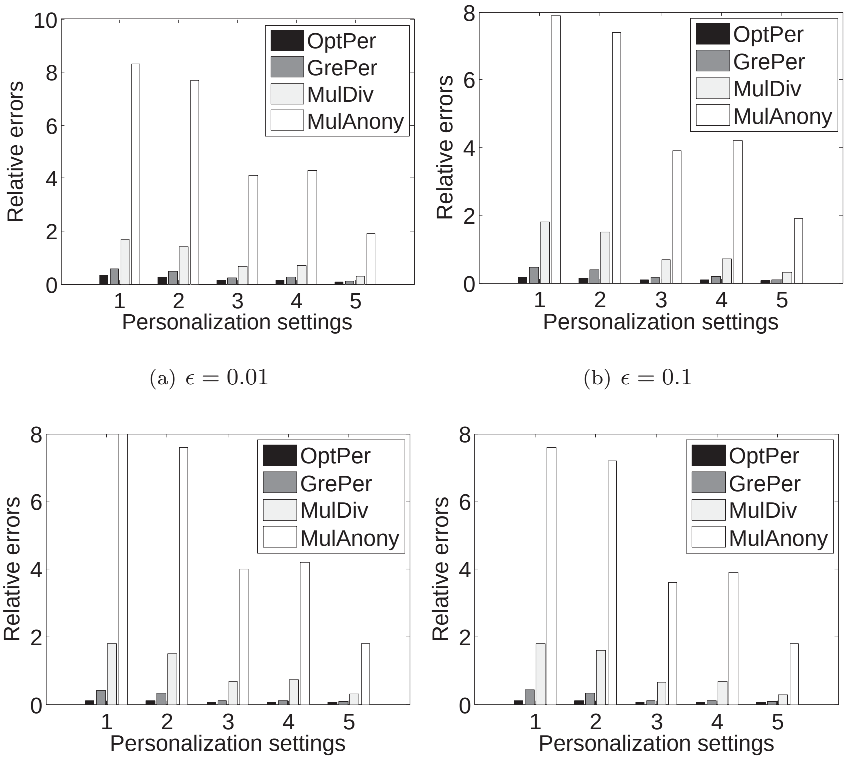
We first vary the number of tuples from 10k to 40k, and keep $|H| = 100$ and $K = 1000$. Fig. 3 shows that the relative error decreases when number of tuples increases. This is because more tuples are included in a partition and average inference probability is lower. The error of MulAnony is significantly larger than that of other algorithms in almost all cases. This is because very few tuples contain the same sensitive attribute set, and sensitive values of these tuples can be the same, making a large number of suppressions to satisfy the α condition. The errors of OptPer and GrePer stay lower than that of MulDiff in all cases, which is caused by the differences in handling personal privacy requirements. Also note that errors of all algorithms decrease slightly when ϵ increases. This is because when ϵ is larger, the privacy offered is weaker, and the probability of selecting the optimal operation is larger.

Fig. 4 illustrates the errors under different dimensions of H, ranging from 10 to 100. Since the error of MulAnony is much higher than other algorithms, we do not depict the error of MulAnony in the figures. In all cases, the number of tuples is set to 40k, and K is set to 1000. As depicted in Fig. 4, OptPer and GrePer outperform MulDiff significantly in high H dimension cases. The reason is that OptPer and GrePer leverage the extra information of personalized sensitive attributes to guide partition operations, while MulDiff treats all tuples with a uniform privacy setting, which involves more extra sensitive attributes when the dimension of H is larger. The errors of all algorithms increase with the H dimension. This is because when dimension is higher, there are more sensitive values contained in the dataset.

We also investigate the utility of the three algorithms over the number of partitions K, with number of tuples set to 40k and H dimension set to 100. Fig. 5 depicts that OptPer and GrePer produce much less errors than that of MulDiff. And the errors of all algorithms increase with K. This is because when K is larger, there are more partition operations, making randomized algorithms more likely to select an operation with large error.

Note that the errors of OptPer and GrePer increase linearly with K (Fig. 5) and $|H|$ (Fig. 4). In our utility analysis, we derive the upper bound of the expected error of OptPer in Theorem 4, which consists of the optimal error $E^*(K, D)$ and the error introduced by randomization. Theorem 4 shows that the error introduced by randomization grows linearly with K and $|H|$. The experimental results are consistent with the analytical result and further demonstrate that the error of OptPer and GrePer scale linearly with K and H dimension.

To validate our protocols under different personalized settings, we vary the personalized settings as shown in Table 2. Fig. 6 depicts the errors of different algorithms under various personalized settings. The results demonstrate that our protocols outperform baseline schemes in all the five personalized settings. We also observe that the errors of all algorithms under distribution 1 and 2 are larger than other distributions. This is because the total number of sensitive attributes under distribution 1 and 2 are larger than those under other distributions.



(a) $\epsilon = 0.01$

(b) $\epsilon = 0.1$

(c) $\epsilon = 1$

(d) $\epsilon = 2$

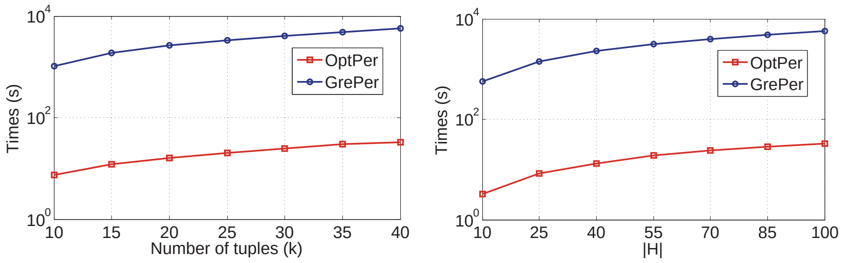
Figure 6: Errors vs. different personalized settings.

Table 2: Personalized settings

Distributions	1	2	3	4	5
Level 1	30%	10%	30%	60%	60%
Level 2	10%	60%	60%	10%	30%
Level 3	60%	30%	10%	30%	10%

6.3. Scalability

We test the scalability of OptPer and GrePer by varying the number of tuples from 10k to 40k, with $\epsilon = 0.1$, $|H| = 1000$ and $K = 100$. Fig. 7.(a) depicts



(a) Running time vs. number of tuples (in thousands)

(b) Running time vs. the dimension of sensitive attribute

Figure 7: Running time

the results. As expected, the running time of each algorithm grows linearly with the number of tuples, and GrePer runs much faster than OptPer in the same settings. In Fig. 7.(b), we run both protocols on datasets with different dimensions of sensitive attribute. The results demonstrate that both OptPer and GrePer achieve linear complexity with respect to the dimension of sensitive attribute.

The results demonstrated in Fig. 7 are consistent with our complexity analyses. As stated in previous sections, the complexities of OptPer and GrePer are $O(|H|(1 + Km^{2|I|})n)$ and $O(|I| \cdot |H| \cdot Kmn)$, respectively. Since m stays relatively the same when the number of tuple n increases, both algorithms have linear computation costs. Considering the m part of the complexities, GrePer reduces $O(m^{2|I|})$ to $O(|I|m)$, compared with GrePer.

7. Conclusion

This paper studied the problem of privacy-preserving healthcare data outsourcing. A framework based on hybrid cloud was proposed to provide personalized privacy protection over high-dimensional healthcare data. Under the framework, we devised two sanitization protocols to anonymize the dataset on the private and public clouds based on randomized data partitioning. The protocols are proved to be resistant to collusion between the public CSP and the

DUs. Analytical results are derived to verify the usability and efficiency of the protocols. Experiments on real-life datasets validate the superiority of our approaches over a number of baseline techniques.

Acknowledgement

The research was supported in part by grants from 973 project 2013CB329006, China NSFC under Grant 61173156, RGC under the contracts CERG 622613, 16212714, HKUST6/CRF/12R, and M-HKUST609/13, as well as the grant from Huawei-HKUST joint lab.

References

[1] Omnibus HIPAA rule in the Federal Register, 2013, http://www.gpo.gov/fdsys/pkg/FR-2013-01-25/pdf/2013-01073.pdf.

[2] W. Wang, Q. Zhang, Towards long-term privacy preservation: A context-aware perspective, IEEE Wireless Commun.

[3] N. Cao, C. Wang, M. Li, K. Ren, W. Lou, Privacy-preserving multi-keyword ranked search over encrypted cloud data, in: Proc. IEEE INFOCOM, 2011.

[4] J. Yuan, S. Yu, Efficient privacy-preserving biometric identification in cloud computing, in: Proc. IEEE INFOCOM, 2013.

[5] C. Wang, K. Ren, S. Yu, K. M. R. Urs, Achieving usable and privacy-assured similarity search over outsourced cloud data, in: Proc. IEEE INFOCOM, 2012.

[6] J. Liu, K. Wang, On optimal anonymization for l+-diversity, in: Proc. IEEE ICDE, 2010.

[7] N. Mohammed, B. Fung, P. Hung, C. Lee, Anonymizing healthcare data: a case study on the blood transfusion service, in: Proc. ACM SIGKDD, 2009.

[8] X. Xiao, Y. Tao, Personalized privacy preservation, in: Proc. ACM SIGMOD, 2006.

[9] W. Wang, Q. Zhang, Privacy-preserving collaborative spectrum sensing with multiple service providers, IEEE Trans. Wireless Commun. 14 (2) (2015) 1011–1019.

[10] C. Dwork, F. McSherry, K. Nissim, A. Smith, Calibrating noise to sensitivity in private data analysis, Springer Theory of Cryptography (2006) 265–284.

[11] G. Cormode, M. Procopiuc, E. Shen, D. Srivastava, T. Yu, Differentially private spatial decompositions, in: Proc. IEEE ICDE, 2012.

[12] F. McSherry, R. Mahajan, Differentially-private network trace analysis, in: Proc. ACM SIGCOMM, 2010.

[13] S. Lee, E. L. Wong, D. Goel, M. Dahlin, V. Shmatikov, πbox: a platform for privacy-preserving apps, in: NSDI, 2013.

[14] M. Li, S. Yu, Y. Zheng, K. Ren, W. Lou, Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption, IEEE Trans. Parallel Distrib. Syst 24 (1) (2013) 131–143.

[15] T. Jung, X.-Y. Li, Z. Wan, P. Li, M. Wan, Privacy preserving cloud data access with multi-authorities, in: Proc. IEEE INFOCOM, 2013.

[16] V. Ciriani, et al., Fragmentation design for efficient query execution over sensitive distributed databases, in: Proc. IEEE ICDCS, 2009.

[17] B. Furht, A. Escalante, Handbook of cloud computing, Springer, 2010.

[18] M. Li, S. Yu, N. Cao, W. Lou, Authorized private keyword search over encrypted data in cloud computing, in: Proc. IEEE ICDCS, 2011.

[19] P. Samarati, Protecting respondents identities in microdata release, IEEE Trans. Knowl. Data Eng. 13 (6) (2001) 1010–1027.

[20] A. Machanavajjhala, D. Kifer, J. Gehrke, M. Venkitasubramaniam, l-diversity: Privacy beyond k-anonymity, ACM Trans. Knowl. Discov. Data 1 (1) (2007) 1–52.

[21] R. Wong, J. Li, A. Fu, K. Wang, (α,k) - anonymity: an enhanced k-anonymity model for privacy preserving data publishing, in: SIGKDD, 2006.

[22] R. Wong, A. Fu, K. Wang, J. Pei, Minimality attack in privacy preserving data publishing, in: VLDB, 2007.

[23] F. McSherry, K. Talwar, Mechanism design via differential privacy, in: Proc. IEEE FOCS, 2007.

[24] S. Peng, Y. Yang, Z. Zhang, M. Winslett, Y. Yu, Dp-tree: indexing multi-dimensional data under differential privacy, in: Proc. ACM SIGMOD (poster), 2012.

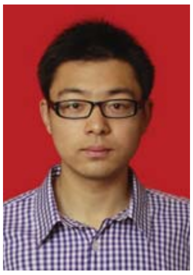
[25] B. Ding, M. Winslett, J. Han, Z. Li, Differentially private data cubes: optimizing noise sources and consistency, in: Proc. ACM SIGMOD, 2011.

[26] K. Zhang, X. Zhou, Y. Chen, X. Wang, Y. Ruan, Sedic: privacy-aware data intensive computing on hybrid clouds, in: Proc. ACM CCS, 2011.

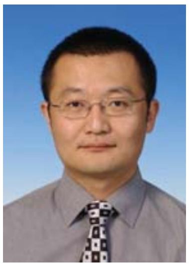
[27] Z. Zhou, H. Zhang, X. Du, P. Li, X. Yu, Prometheus: Privacy-aware data retrieval on hybrid cloud, in: Proc. IEEE INFOCOM, 2013.

[28] S. Boyd, L. Vandenberghe, Convex optimization, Cambridge Univ Pr, 2004.

[29] F. McSherry, Privacy integrated queries: an extensible platform for privacy-preserving data analysis, in: Proc. ACM SIGMOD, 2009.



Wei Wang is currently a Research Assistant Professor in Fok Ying Tung Graduate School, Hong Kong University of Science and Technology (HKUST). He received his Ph.D. degree in Department of Computer Science and Engineering from HKUST. Before he joined HKUST, he received his bachelor degree in Electronics and Information Engineering from Huazhong University of Science and Technology, Hubei, China, in June 2010. His research interests include privacy and fault management in wireless networks.



Lei Chen is currently an associate professor in the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. His research interests include social networks, probabilistic and uncertain data, cloud data processing, and graph data. He is a member of the IEEE and the IEEE Computer Society.

Qian Zhang joined Hong Kong University of Science and Technology in Sept. 2005 where she is a full Professor in the Department of Computer Science and Engineering. Before that, she was in Microsoft Research Asia, Beijing, from July 1999, where she was the research manager of the Wireless and Networking Group. Dr. Zhang has published about 300 refereed papers in international leading journals and key conferences in the areas of wireless/Internet multimedia networking, wireless communications and networking, wireless sensor networks, and overlay networking. She is a Fellow of IEEE for “contribution to the mobility and spectrum management of wireless networks and mobile communications”. Dr. Zhang has received MIT TR100 (MIT Technology Review) worlds top young innovator award. She also received the Best Asia Pacific (AP) Young Researcher Award elected by IEEE Communication Society in year 2004. Her current research is on cognitive and cooperative networks, dynamic spectrum access and management, as well as wireless sensor networks. Dr. Zhang received the B.S., M.S., and Ph.D. degrees from Wuhan University, China, in 1994, 1996, and 1999, respectively, all in computer science.