Distributed Privacy Preserving Data Collection using Cryptographic Techniques

Mingqiang Xue[1], Panagiotis Papadimitriou[2], Chedy Raïssi[1], Panos Kalnis[3] and Hung Keng Pung[1]

[1]*Computer Science Department, National University of Singapore*
{xuemingq, raissi, punghk}@comp.nus.edu.sg

[2]*Stanford University*
papadimitriou@stanford.edu

[3] *King Abdullah University of Science and Technology*
panos.kalnis@kaust.edu.sa

***Abstract*— We study the distributed k-anonymous data collection problem: a data collector (e.g., a medical research institute) wishes to collect data (e.g., medical records) from a group of respondents (e.g., patients). Each respondent owns a multi-attributed record which contains both non-sensitive (e.g., quasi-identifiers) and sensitive information (e.g., a particular disease), and submits it to the data collector. Assuming T is the table formed by all the respondent data records, we say that the data collection process is k-anonymous if it allows the data collector to obtain a k-anonymized version of T without revealing the original records to any adversary. In contrast to most k-anonymization approaches which trust the data collector, our work assumes that the adversary can be any third party, including the data collector and the other responders.**

We propose a distributed data collection protocol that outputs a k-anonymized table by generalization of quasi-identifier attributes. The protocol employs cryptographic techniques such as homomorphic encryption, private information retrieval and secure multiparty computation to ensure the privacy goal in the process of data collection. Meanwhile, the protocol is designed to leak limited but non-critical information (mainly statistical information about the non-sensitive attributes of the data respondents) to achieve practicability and efficiency. Experiments show that the utility of the k-anonymized table derived by our protocol is in par with the utility achieved by traditional k-anonymization techniques that trust the data collector.

I. Introduction

In the data collection problem a third party collects data from a set of individuals who are concerned about their privacy. Specifically, we consider a setting in which there is a set of data *respondents*, each of whom has a row of a table, and a data *collector*, who wants to collect all the rows of the table. For example, a medical researcher may request from some patients that each of them provides him with a health record that consists of three attributes: ⟨*age*, *weight* and *disease*⟩.

Each patient is willing to share his record with the researcher or other patients provided that none of them learns his identity. Although the health record contains no explicit identifiers such as name and phone numbers, an adversarial medical researcher may be able to retrieve a patient's identity using the combination of *age* and *weight* with external information. For instance, in the data records of Figure 1(a), we see that there is only one patient with age 45 and weight 60 and this patient suffers from Gastritis (the third row). If the researcher knows a particular patient with the same age and weight values, after collecting all the data records he learns that this patient suffers from Gastritis. In this case the attributes *age* and *weight* serve as a quasi-identifier.

The patients feel comfortable to provide the researcher with medical records only if there is a guarantee that the researcher can only form a k-anonymous table with their records, i.e., each record has at least $k-1$ other records whose values are indistinct over the quasi-identifier attributes [1]. The patients may achieve this by generalizing the values that correspond to the quasi-identifiers [2]. In Figure 1(b), observe that if each patient discloses only some appropriate range of his age and weight instead of the actual values, then the medical researcher receives a 4-anonymous table. In this case, the researcher can only determine with probability 1/4 the disease of the 45-year old patient.

In the k-anonymous data collection problem the data respondents look for the minimum possible generalization of the quasi-identifier values so that the collector receives a k-anonymous table. The constraint of the problem is that although the respondents can communicate with each other and with the collector, no single participant can leak any

Age	*Weight*	*Disease*
35	50	Gastritis
40	55	Diabetes
45	60	Gastritis
45	65	Pneumonia
55	65	Gastritis
60	60	Diabetes
60	55	Diabetes
65	50	Alzheimer
55	75	Diabetes
60	75	Flu
65	85	Flu
70	80	Alzheimer

(a) Original

Age	*Weight*	*Disease*
[35, 45]	[50, 65]	Gastritis
[35, 45]	[50, 65]	Diabetes
[35, 45]	[50, 65]	Gastritis
[35, 45]	[50, 65]	Pneumonia
[55, 65]	[50, 65]	Gastritis
[55, 65]	[50, 65]	Diabetes
[55, 65]	[50, 65]	Diabetes
[55, 65]	[50, 65]	Alzheimer
[55, 70]	[75, 85]	Diabetes
[55, 70]	[75, 85]	Flu
[55, 70]	[75, 85]	Flu
[55, 70]	[75, 85]	Alzheimer

(b) k-Anonymous

Fig. 1. Distributed medical records table. Each patient owns a row of the table.

information to the others except from his final anonymous record.

Traditional table k-anonymization techniques [1] are not applicable to our problem, since they assume that there is a single trusted party that has access to all the table records. The shortcoming of such an approach is that: if the trusted party is compromised then the privacy of all respondents is compromised as well. In our approach, each respondent owns his own record and does not convey its information to any other party prior to its k-anonymization.

Our setting is similar to the distributed data collection scenario studied by Zhon et al [3]. The difference is that in their work the respondents create a k-anonymous table for the collector by suppressing quasi-identifier attribute values. We use generalization instead of suppression, which makes the problem not only more general but also much more practical and challenging. Our problem is more general because suppression is considered as a special case of generalization: a suppressed attribute value is equivalent to the value generalization to the higher level of abstraction. The problem is also more practical because generalized attribute values have greater utility than suppressed values, since they convey more information to the data collector without compromising the respondents' privacy. Finally our problem is more challenging, because the required cryprographic techniques do not support directly value generalization, forcing us to develop novel solutions. For example, see the problem of partitioning the respondents' records into subsets with more than k records. In case of data suppression the respondents have to select the appropriate attributes to suppress so that all the quasi-identifiers in one subset are exactly the same. To test the similarity of values that they cannot disclose, the respondents can simply hash them using the same cryptographic hash function and then compare the digests that arise. In case of value generalization the respondents have to partition records with similar but not necessarily identical quasi-identifers. Such a partitioning without disclosing information about the quasi-identifiers is challenging. In addition, value generalization requires a mechanism for the definition of a common abstraction level in each partition that is not needed in the value suppression case.

In this paper we propose an efficient protocol for k-anonymous data collection. The protocol has the following four stages: (i) In the *preparation stage*, each respondent maps the quasi-identifiers of the data record to a private 1D integer using some public mapping function. The secrecy of the 1D integer is as important as the quasi-identifiers. (ii) In the *first stage*, each respondent secretly maps the private 1D integer to a new 1D value in a public space. In the new public space, the localities of the private 1D integers from all respondents are preserved, but the values are statistically hidden. (iii) In the *second stage*, k-anonymization is performed in the new public 1D space to determine a set of k-partitions (i.e. a group with a least k respondents). (iv) In the *third stage*, the same set of k-partitions is used to privately k-anonymize the data records of the respondents. Finally the respondents submit their generalized data records to the collector.

Our contributions are the following:

- We formally define the problem of distributed k-anonymous data collection with respondents that can generalize attribute values.
- We present an efficient and privacy-preserving protocol for k-anonymous data collection.
- We show theoretically that the information leakage that our protocol yields is quantifiable and can be limited under our security parameters.
- We provide a detailed complexity analysis of the protocol to illustrate its efficiency.
- We evaluate our protocol experimentally to show that it achieves similar utility preservation as the state-of-the art non-distributed k-anonymity algorithm [4] that trusts the data collector.

The rest of the paper is organized as follows: in Section II, we review the related work. In Section III, we describe the system, model the adversaries and define our privacy goals. Sections IV and V explain the rationale behind our solution and summarize the protocol. In Section VI, we analyze the privacy guarantee of our approach, and the complexity metrics. In Section VII we show experimentally that our distributed k-anonymous data collection algorithm preserves both the privacy and the utility of the data. Lastly, we conclude in Section VIII.

II. Related Work

A. Secure Multiparty Computation

In theory, our problem is essentially an instance of the Secure Multiparty Computation (*SMPC*) [5] problem. In *SMPC*, different parties who have their own private data wish to jointly compute the value of a public function on them without revealing their private data to other parties. A classic example of *SMPC* is the millionaire problem: Alice and Bob are two millionaires who want to find out who is the richer without revealing the precise amount of their wealth.

An *SMPC* protocol provides security, if by the end of its execution, each party does not learn extra information besides of the information from the description of the public function, the result of computation and any information deduced from himself. It has been shown that there exists a polynomial time generic solution [6], [7] that achieves the same functionality for any polynomial time algorithm by representing the problem as a boolean circuit. However, when the size of the input is large, it is computationally impractical to use the pure circuit based generic solution.

If our problem is treated as *SMPC*, the input is the data respondent records and the output is the k-anonymized table. Unlike generally studied two-party or three-party computation problems, our problem may involve several thousands of respondents. Therefore, scalability is an important requirement. The solution that we propose in this paper does not strictly conform to *SMPC* security requirements. This is because in order to achieve efficiency and high utility, we leak certain statistical information about the data respondents' non-sensitive

values. The amount of information is quantifiable and can be limited below a predefined threshold.

B. Private Entries Suppression

As we discussed in Section I, the most relevant work to ours is found in [3]. In this paper, the authors proposed a distributed, privacy-preserving version of the Meyerson and Williams's algorithm (*MW*) [8], which is an $O(k \log k)$ approximation to optimal k-anonymity based on entry suppression; in contrast, our algorithm supports generalization. Similar to our scheme, in order to achieve efficient distributed anonymization the distributed MW algorithm reveals information about the relative distance between different data record pairs. In [3], the distance between two records is the *number* of differences in the attribute values. For example, in Figure 1(a), the distance between the first two records is 2, since age 35 is different from age 40 and weight 50 is different from weight 55. In our approach, the distance between two records depends on the *distance* between the corresponding attribute values, which is more difficult to evaluate securely.

C. Horizontally and Vertically Partitioned Table

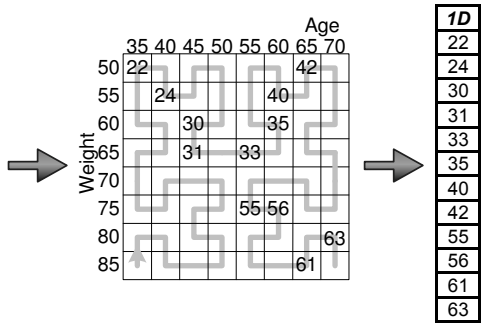
There are also existing works on distributed k-anonymity which only consider two-party computation. In [9], the authors introduced a technique that supports joint computation of a k-anonymized table from the data of two parties, where each party owns a vertical partition of the table. In [10], the authors considered the case where each party owns a horizontal partition of a table and they attempt to jointly form a k-anonymized table. Our problem is more complicated than these approaches because we do not pose a limit on the number of parties that share the tabular data. Moreover, each party in our approach has very little information about the table, since he owns only a single data record versus a big portion of the table in case of two-party computation.

D. FALL

The k-anonymization algorithm that we present in this paper is based on the the Fast data Anonymization with Low information Loss (*FALL*) algrithm proposed by Ghinita *et al* [4]. In their work, efficient k-anonymization is achieved in two steps. The first step includes the transformation of u-dimensional to 1-dimensional data, in which a multi-attributed data record is converted to an integer using a space filling curve (e.g. Hilbert curve [11]). An important characteristic of such space filling curves is that if two data records are close to each other in the uD space, they also tend to be close to each other in the 1D space. For example, Figure 2 shows a Hilbert walk that visits each cell in the two dimensional space (*Weight* $\times$ *Age*) and assigns each cell with an integer in increasing order along the walk. Using the Hilbert mapping, each data record in the original *Weight* $\times$ *Age* table is mapped to a unique 1D integral data value. The authors show that both uD categorical data and numerical data can be converted to 1D data using this method. In the second step, an optimal 1D k-anonymization is performed over the set of integers obtained in the first step using an efficient algorithm based on dynamic programming. The authors show that the optimal k-anonymity for 1D data can be achieved over optimal non-overlapping partitions of sorted data. The same partitions will be used for forming the *equivalence classes* of data records. Although the final k-anonymized table may not be an optimal one, the authors show that, in practice, their approach outperforms the state-of-the-art technique Mondrian [12] in terms of both execution time and utility loss.

The utility loss metric used by the authors is called *Global Certainty Penalty* (*GCP*). The *GCP* is derived from the *Normalized Certainty Penalty* (*NCP*) [13] which only measures the utility loss of a single *equivalence class*. However, the *GCP* measures the utility loss of the entire anonymized table. The *GCP* value is between 0 and 1. Value 0 indicates no utility loss, i.e., the anonymized table is exactly the same as the original one and value 1 indicates complete utility loss, i.e., all records are anonymized into a single *equivalence class*. In this paper, we adopt the same utility metric to compare the utility of our approach with the *FALL* algorithm.

Age	Weight
35	50
40	55
45	60
45	65
55	65
60	60
60	55
65	50
55	75
60	75
65	85
70	80



1D
22
24
30
31
33
35
40
42
55
56
61
63

Fig. 2. Mapping 2D to 1D points using Hilbert curve.

E. Other Related Work

It has been pointed out that k-anonymity does not always guarantee the privacy. For example, the authors in [14] show that in a k-anonymized table, it is possible that the sensitive attribute values are the same for all the data records within the same *equivalence class*. Thus, the adversary can find the sensitive value of a victim with 100% confidence. To solve the problem, they proposed l-diversity, in which the sensitive attribute values are guaranteed to be diverse in an *equivalence class*. In a fine-grained analysis [15], the authors proposed a new privacy metric, i.e. t-closeness, to measure the change of believes in the distribution of sensitive values before and after releasing the table. Nevertheless, our paper focuses on achieving k-anonymous data collection.

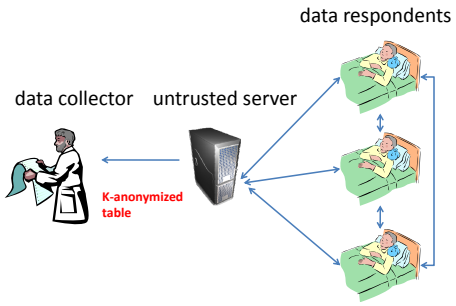


Fig. 3. System structure and participants

III. Problem Formulation

A. The System and the adversaries

The system employs the Client-Server architecture. Each respondent runs a client. There is an untrusted server that facilitates the communication and computation in the system on behalf of the collector. We assume that all messages are encrypted, and secure communication channels exist between any pair of communicating parties. By the end of the protocol execution, a k-anonymized table, generalized from the data records of the respondents, is created at the server side (i.e., the collector). Figure 3 shows the system structure and the participants in the data collection process.

The adversaries can either be the respondents or the server. We assume that the adversaries follow the semi-honest model, which means that they always correctly follow the protocol but are curious in gaining additional information during the execution of the protocol. In addition, we assume that the adversarial respondents can collaborate with each other to gain additional information. However, the server, which is considered to be adversarial, does not collaborate with the adversaries. We assume there can be up to $t_{ss}-1$ adversaries among the respondents, where t_{ss} is a security parameter.

B. Notion of Privacy

Initially, there are x number of respondents each running an instance of the client. We denote the set of non-sensitive attributes of the data records $A = \{a_1, a_2, \ldots, a_u\}$ and the set of sensitive attributes $\{s_1, s_2, \ldots, s_v\}$. The data record for the i^{th} respondent is represented as $t_i = \{a_1^i, \ldots, a_u^i, s_1^i, \ldots, s_v^i\}$ and $T = \{t_1, t_2, \ldots, t_x\}$ is the table formed by the original data records of the respondents. $t_i.A$ represents the non-sensitive attribute values for the data record t_i. Similarly, $T.A$ represents the non-sensitive attributes columns of table T. Let $\mathcal{K}(T)$ denote the final output of the protocol, which is a k-anonymized table generalized from T. Let $\mathcal{L}_i$ and $\mathcal{L}_{svr}$ denote the amount of information leaked in the process of protocol execution to the respondent i and the server, respectively.

During the execution of the protocol, the view of a party uniquely consists of four objects: (i) the data owned by the party, (ii) the assigned key shares, (iii) the set of received messages and (iv) all the random coin flips picked by this party. Let $\mathsf{view}_i(T)$ (respectively $\mathsf{view}_{svr}(T)$) denote the view of the respondent i (respectively the view of server) and let $\equiv_c$ denote the *computational indistinguishability of probability ensembles*. We adopt a similar privacy notion as in [3]:

Definition 1: A protocol for k-anonymous data collection leaks only $\mathcal{L}_i$ for the respondent i and $\mathcal{L}_{svr}$ for the server if there exist probabilistic polynomial-time simulators M_{svr} and $M_1, M_2, \ldots, M_x$ such that:

$$\{M_{svr}(\text{keys}_{svr}, \mathcal{K}(T), \mathcal{L}_{svr})\}_T \equiv_c \{\mathsf{view}_{svr}(T)\}_T \quad (1)$$

and for each $i \in [1, x]$,

$$\{M_i(\text{keys}_i, \mathcal{K}(T), \mathcal{L}_i)\}_T \equiv_c \{\mathsf{view}_i(T)\}_T \quad (2)$$

The contents of $\mathcal{L}_{svr}$ and $\mathcal{L}_i$ are mainly statistical information about the respondent's quasi-identifiers. More details will be given on these values along with the descriptions of our proposed solution. Later in this paper, we prove that the execution of our proposed protocol respects the previous definition by only leaking $\mathcal{L}_{svr}$ and $\mathcal{L}_i$ for each respondent i.

C. Using Secret Sharing

To conquer up to $t_{ss}-1$ collaborating adversaries among the respondents, we initially assume that there is a global private key SK shared by all the respondents and the server using a $(t_{ss}, x+1)$ threshold secret sharing scheme [16]. The shares owned by the respondents and the server are denoted as sk_1, sk_2, ..., sk_x, and sk_{svr}, respectively. With a $(t_{ss}, x+1)$ secret sharing scheme, t_{ss} or more key shares are necessary in order to successfully reconstruct the decryption function with the secret key SK, while less than t_{ss} key shares give absolutely no information about SK. The corresponding public key of the private key SK is denoted as PK. The public key encryption algorithm that we use in this paper is the Paillier's cryptosystem [17] because of its useful additive homomorphic property. This very important property will be discussed in the next section. The security of Paillier's cryptosystem relies on the Composite Residuosity (CR) assumption. In order to support threshold secret sharing, we use a threshold version of Paillier's encryption as described in [18] based on Asmuth-Bloom secret sharing [19]. We use $E()$ (respectively $D()$) to represent the encryption function with PK (respectively the collaborative decryption function with SK).

IV. Towards the Solution

In this section, we first illustrate the ideas. Then we introduce and describe the new notions. Last, we elaborate on the principle of our solution and its precise and necessary steps.

A pure *SMPC* solution which leaks strictly no information is too expensive to realize. Alternatively, we can design a protocol that leaks certain information but satisfies the following two requirements simultaneously: (i) Both computation and communication costs are greatly reduced. (ii) The information leakage is accurately quantifiable and can be controlled. Our solution is designed and based exactly on this principle. In the following, we explain the proposed solution using a top-down approach: we first give a sketch to the main stages of

the proposed solution together with notations, definitions and requirements for each stage of the protocol. Second, we give the technical details of each stage of the protocol.

A. A Sketch of the Solution

Preparation stage: The main goal of this stage is to map the uD records to 1D integers. In this stage, each respondent independently performs uD to 1D mapping using a space filling curve, e.g., the Hilbert curve. The purpose of performing uD to 1D mapping is to reduce data dimensionality for efficient k-anonymization in a later stage. Symbolically, the mapping for $t_i.A$ is denoted as $c_i = \mathcal{S}(t_i.A)$. Each integer c_i is in the range $[1, c_{max}]$, where c_{max} denotes the maximum possible value that the mapping function can yield. For example, if we use the Hilbert curve of Figure 2, then $c_{max} = 64$. The set of mapped values for all the respondents is denoted as $S = \{c_1, c_2, \ldots, c_x\}$. As $\mathcal{S}()$ is a public one-to-one function, revealing c_i is equivalent to revealing $t_i.A$ for the respondent i. Therefore, the value of c_i should be kept secret by the i^{th} respondent after mapping.

Without loss of generality, we assume that the values in S are already sorted in ascending order for the ease of subsequent discussion. If the actual S is not sorted, we simply need to reassign the IDs of the respondents to make it sorted.

Stage 1: The main aim of this stage is to achieve *probabilistic locality preserving* mapping. Symbolically, the i^{th} respondent maps the secret integer c_i to a real number $r^+_{c_i}$ using function $\mathcal{F}()$, i.e. $r^+_{c_i} = \mathcal{F}(c_i)$. The set of mapped values for all the respondents is represented as $\mathcal{F}(S) = \{r^+_{c_1}, r^+_{c_2}, \ldots, r^+_{c_x}\}$. We require that the mapping from each c_i to $r^+_{c_i}$ by $\mathcal{F}()$ preserves certain order and distance relations for the integers in S for utility efficient k-anonymization. In this paper, this property is known as *probabilistic locality preserving*, which is formally described by the following definition:

Definition 2: Given any two pre-images c_{i_1}, c_{i_2}, a mapping function $\mathcal{F}()$ is *order preserving* if:

$$c_{i_1} \leq c_{i_2} \Rightarrow \mathcal{F}(c_{i_1}) \leq \mathcal{F}(c_{i_2}) \tag{3}$$

Given any three pre-images $c_{i_1}, c_{i_2}, c_{i_3}$, and the distances $dist_1 = |c_{i_1} - c_{i_2}|$, $dist_2 = |c_{i_2} - c_{i_3}|$, a mapping function $\mathcal{F}()$ is *probabilistic distance preserving* if:

$$dist_1 \leq dist_2 \Rightarrow \Pr(fdist_1 \leq fdist_2) \geq \frac{1}{2} \tag{4}$$

and it increases with $dist_2$, where $fdist_1 = |\mathcal{F}(c_{i_1}) - \mathcal{F}(c_{i_2})|$ and $fdist_2 = |\mathcal{F}(c_{i_2}) - \mathcal{F}(c_{i_3})|$.

A mapping function $\mathcal{F}()$ is *probabilistic locality preserving* if it is both *order preserving* and *probabilistic distance preserving*.

In addition to the requirement of *probabilistic locality preserving*, we also require that the mapping from c_i to $r^+_{c_i}$ does not reveal too much information about c_i. This property is known as γ-concealing which is formally defined as follows:

Definition 3: Given the pre-image c_i and $r^+_{c_i} = \mathcal{F}(c_i)$, the function $\mathcal{F}()$ is γ-concealing if $\Pr(c_{mle} = c_i | r^+_{c_i}) \leq 1 - \gamma$ for the Maximum Likelihood Estimation (*MLE*) c_{mle} of c_i.

Achieving both *probabilistic locality preserving* and γ-*concealing* seem to be two contradicting goals, as one aims to reveal information and the other aims to conceal information. However, in practice, both goals are realizable by using appropriate parameters. The set of values in $\mathcal{F}(S)$ is uploaded to the server for k-anonymization in the next stage.

Stage 2: The goal of this stage is to determine a set of k-partitions of respondents based on the set of values in $\mathcal{F}(S)$. The utility efficient k-partitions can be formed by using 1D optimal k-anonymization algorithm as proposed in *FALL*. Alternatively, we can adopt the polynomial 1D optimal k-anonymization algorithm proposed in [20]. The authors have shown that the optimal 1D k-anonymization is equivalent to finding the shortest path on a specially constructed graph. The readers may refer to [4], [20] for the details of the two algorithms.

Stage 3: The goal of this stage is to privately anonymize the respondent data records based on the k-partitions from *Stage 2*. This stage involves secure computation of *equivalence classes* for the respondents in the same k-partition. As $\mathcal{F}(S)$ is *probabilistic locality preserving* for the data values in S, if we use the same k-partitions created on $\mathcal{F}(S)$ to anonymize T, we expect that the k-anonymized table $\mathcal{K}(T)$ preserves the utility well. In the following, we describe the technical details in *Stages 1*, *2*, and *3*.

B. Technical Details

Stage 1. Probabilistic Locality Preserving Mapping: The challenge of performing *probabilistic locality preserving* mapping in this application is that all the data values in S are distributed, and we must ensure the secrecy of c_i for respondent i in the mapping process. Building *directly* an encryption scheme respecting the notions of distance and order among respondents' data is difficult. Instead, in our approach we build a large encrypted index $E(R+) = \{E(r^+_1), \ldots, E(r^+_{c_{max}})\}$ on the server side containing c_{max} randomly generated numbers that correspond to all integers in the range $[1, c_{max}]$ of the mapping function $\mathcal{S}$. For example, if the the mapping function uses the Hilbert curve of Figure 2, then set $E(R+)$ will contain 64 numbers, one for each cell of the grid. Each respondent i retrieves then the c_i^{th} item in the encrypted index, i.e., the item $E(r^+_{c_i})$, in a private manner and can jointly and safely decrypt it with other respondents in order to build the k-anonymized data.

In the proposed solution, four steps are needed in order to achieve *probabilistic locality preserving mapping*. These steps are briefly sketched as follows:

- *Step 1:* Two sets of encrypted real numbers are created at the server side: $E(R_{init})$ and $E(R_p)$. It is required that the plaintexts values of the real numbers are not known to any party in the protocol.
- *Step 2:* The set of encrypted real numbers $E(R+)$, i.e., the encrypted index, is created in a recursive way using the two sets of encrypted real numbers from *Step 1*: the set $E(R_{init})$ is used to define the value of the first encrypted number $E(r^+_1)$ and the set $E(R_p)$ is

used to define number $E(r_i^+)$ in terms of $E(r_{i-1}^+)$. The construction procedure of the encrypted numbers in $E(R+)$ gurantees that the corresponding plainttext values are sorted in ascending order.

- *Step 3:* Respondent i retrieves the c_i^{th} item from index $E(R+)$ created in *Step 2* using a *private information retrieval* scheme.
- *Step 4:* The retrieved encrypted item is jointly decrypted by t_{ss} parties, and uploaded to the server. Its plaintext is defined as $r_{c_i}^+$, i.e., the image of c_i under $\mathcal{F}()$.

In the following, we describe the above four steps in detail.

In *Step 1*, we first describe how to create one encrypted random real number whose plaintext value is not known by any parties. The creation of two sets of encrypted real numbers is just a simple repetition of this process.

In order to hide the value of a random number, each of these is jointly created by both a respondent and the server. We call such a random number a *joint random number*. To compute an encrypted joint random number $E(r)$, the respondent randomly selects a real number r_{dr} from a uniform distribution in the interval $[\rho_{min}, \rho_{max}]$ (the uniform distribution and the bounded interval are required for the proof of Theorem 1 that comes later). Then the respondent sends the encrypted number $E(r_{dr})$ to the server. The server independently chooses another random real number r_{svr} from the same interval $[\rho_{min}, \rho_{max}]$ and encrypts it to obtain $E(r_{svr})$. The join of the two encrypted real numbers is computed as $E(r) = E(r_{dr}) \cdot E(r_{svr})$. From the additive homomorphic property of the Paillier's encryption [1], it holds that:

$$E(r_{dr}) \cdot E(r_{svr}) = E(r_{dr} + r_{svr}). \tag{5}$$

Therefore we have $E(r) = E(r_{dr} + r_{svr})$. As the value of r is the sum of the random number from respondent i and the server who do not collaborate in our model, no party knows the exact value of r. However, we are aware that both the respondent i and the server knows the range information about r. We denote such range knowledge about the joint random numbers for respondent i and the server as $\mathcal{RG}_i$ and $\mathcal{RG}_{svr}$, respectively. Recall that $\mathcal{L}_{svr}$ and $\mathcal{L}_i$ are the information leakage for the server and the data respectively. Therefore, we have that $\mathcal{RG}_{svr} \in \mathcal{L}_{svr}$ and $\mathcal{RG}_i \in \mathcal{L}_i$. In practice, knowing the range is insufficient for the adversaries to determine the values of the joint random numbers, thus our privacy goal (i.e. hide the exact values) is met.

With the above technique, the first encrypted set of joint random numbers that we create is $E(R_{init}) = \{E(\iota_1), E(\iota_2), E(\ldots), E(\iota_b)\}$, where the size b is a security parameter of the system. Each of the encrypted joint random numbers is created by the server and a randomly selected respondent.

The second set of encrypted joint random numbers that we create on the server side is $E(R_p) = \{E(r_1), E(r_2), \ldots, E(r_{c_{max}})\}$. To create $E(R_p)$, each respondent needs to generate $\lfloor \frac{c_{max}}{x} \rfloor$ or $\lceil \frac{c_{max}}{x} \rceil$ encrypted joint random numbers with the server, if we distribute this task evenly among all the respondents.

In *Step 2*, to build an encrypted set of real numbers $E(R^+) = \{E(r_1^+), E(r_2^+), \ldots, E(r_{c_{max}}^+)\}$ whose plaintexts values are in ascending order based on $E(R_{init})$ and $E(R_p)$, we once again use the additive homomorphic property of Paillier's encryption:

$$\begin{cases} E(r_i^+) = E(r_i) \cdot \prod_{j=1}^{b} E(\iota_j) & i = 1 \\ E(r_i^+) = E(r_{i-1}^+) \cdot E(r_i) & i = 2, \ldots, c_{max} \end{cases} \tag{6}$$

The first element $E(r_1^+)$ is initialized by the product of $E(r_1)$ and the encryption of the sum of all ι_j for $j \in [1, b]$. Each subsequent $E(r_i^+)$ is the product of $E(r_{i-1}^+)$ and $E(r_i)$. Due to the additive homomorphic property, it is clear that the plaintexts values are sorted in ascending order.

In *Step 3*, the encrypted number $E(r_{c_i}^+)$ is retrieved from the server by the respondent i who owns the secret c_i using Private Information Retrieval (*PIR*). In cryptography, *PIR* is a technique that allows a user to retrieve an item from a database server without revealing which item is retrieved. Therefore, the respondent can keep the value of c_i secret while retrieving $E(r_{c_i}^+)$ using a *PIR* scheme. Various *PIR* schemes have been proposed, and in this paper we adopt the single database *PIR* scheme developed in [21] which supports the retrieval of a block of bits with constant communication rate. This *PIR* scheme is proven to be secure based on a simple variant of the Φ-hiding assumption. To hide the complexity of the *PIR* communications, we use the $\mathcal{PIR}(c_i, E(R^+))$ to represent the sub-protocol that privately retrieves the ${c_i}^{th}$ item in the set $E(R^+)$ by the i^{th} respondent, and the result of retrieval is $E(r_{c_i}^+)$. The secrecy of $E(r_{c_i}^+)$ is as important as c_i, as the server can try to discover the value of c_i, if he knows the value of $E(R_{c_i}^+)$ by searching through $E(R^+)$.

In *Step 4*, after the respondent i has retrieved $E(r_{c_i}^+)$, he partially decrypts $E(r_{c_i}^+)$ and send the partially decrypted cipher to $t_{ss} - 2$ other respondents for further decryption. The last partial decryption is done by the server, after which the server obtains the plaintext $r_{c_i}^+$. Note that the server cannot identify the value of c_i by re-encrypting the $r_{c_i}^+$ and search through $E(R^+)$, as the Paillier's encryption is a randomized algorithm in which the output ciphers are different for the same plaintext with different random inputs. Finally, we have achieved the mapping from the c_i to $r_{c_i}^+$. The server obtains the set $\mathcal{F}(S)$ by the end of this step.

We illustrate these four steps in the Figure 4. The first column describes the respondents 1D data. The second column represents the 33^{rd} to 40^{th} entries in $E(R^+)$. The third column represents 33^{rd} to 40^{th} entries in $E(R_p)$. The i^{th} entry of $E(R^+)$ is computed based on the product of the $(i-1)^{th}$ entry of $E(R^+)$ and the i^{th} entry of $E(R_p)$. For example, $E(r_{34}^+) = E(r_{33}^+) \cdot E(r_{34})$ by the additive homomorphic property, $E(r_{34}^+) = E(r_{33}^+ + r_{34})$ which translated in terms of real values gives $E(304.7) = E(293.5) \cdot E(11.2) = E(293.5 + 11.2)$.

[1] Assuming a large modulus N is used so that round up does not take place.

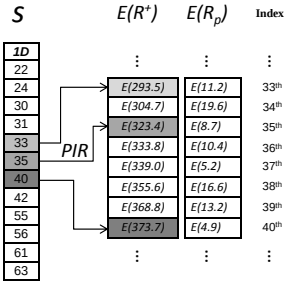


Fig. 4. Example of the probabilistic locality preserving mapping construction.

Theorem 1: The mapping function $\mathcal{F}()$ is *probabilistic locality preserving*.

Proof: Since R_p^+ is a set of ascending real numbers, we have $r_{c_{i_1}}^+ \leq r_{c_{i_2}}^+$, if $c_{i_1} \leq c_{i_2}$. Therefore, $\mathcal{F}()$ is order preserving by Equation 3. To prove that it is also *probabilistic distance preserving*, let c_{i_1}, c_{i_2}, c_{i_3} be any randomly selected pre-images, and $dist_1$, $dist_2$, $fdist_1$ and $fdist_2$ follow the definitions in Definition 2 Equation 4. Assume that $c_{i_1} \leq c_{i_2} \leq c_{i_3}$ and $dist_1 \leq dist_2$. The exact form of the distributions of $fdist_1$ and $fdist_2$ are difficult to estimate. However, since $fdist_1$ ($fdist_2$ *resp.*) is the sum of $dist_1$ ($dist_2$ *resp.*) number of joint random numbers, where each joint random number is the sum of two random uniformly selected real numbers in the interval $[\rho_{min}, \rho_{max}]$, $fdist_1$ and $fdist_2$ can be unbiasedly approximated by continuous normal distribution according to the *central limit theorem*. Let $\mu = \frac{\rho_{min}+\rho_{max}}{2}$ and $\sigma^2 = \frac{(\rho_{min}-\rho_{max})^2}{12}$ be the mean and variance of the uniform distribution respectively, and without ambiguity, $fdist_1$ and $fdist_2$ be the continuous random variables. From the *central limit theorem*, we have $fdist_1 \sim N(dist_1 \cdot 2\mu, dist_1 \cdot 2\sigma^2)$ and $fdist_2 \sim N(dist_2 \cdot 2\mu, dist_2 \cdot 2\sigma^2)$. Therefore, $fdist_1 - fdist_2 \sim N((dist_1 - dist_2) \cdot 2\mu, (dist_1 + dist_2) \cdot 2\sigma^2)$. From the property of continuous normal distribution, $\Pr(fdist_1 - fdist_2 \leq 0) = \Pr(fdist_1 \leq fdist_2) \geq \frac{1}{2}$ when $dist_1 \leq dist_2$ and it increases with $dist_2$. Hence, by Equation 4, $\mathcal{F}()$ is also *probabilistic distance preserving*. Therefore, by Definition 2, $\mathcal{F}()$ is *probabilistic locality preserving*. ■

In terms of information leakage, the server gains knowledge of $\mathcal{F}(S)$ by the end of this stage. Therefore, $\mathcal{F}(S) \in \mathcal{L}_{svr}$.

Stage 2. k-anonymization in the mapped space: Suppose the 1D optimal k-anonymization algorithm in *FALL* is used by the server to form optimal k-anonymization. Let $Z = \{z_1, z_2, \ldots, z_\pi\}$ be the result of the 1D optimal k-anonymization, where the i^{th} element in Z is the ending index of the i^{th} k-partition of respondents and there are π number of k-partitions. We assume the indices in Z are sorted in ascending order, as the optimal k-partition is always found on 1D sorted data. For example, the first k-partition is formed by the respondents $1, 2, \ldots, z_1$, and the second k-partition is formed by the respondents $z_1 + 1, z_1 + 2, \ldots, z_2 - 1$ and so on.

Stage 3. Secure computation of equivalence classes: In this stage, the quasi-identifiers of respondents in the same k-partition defined by Z form an equivalence class in $\mathcal{K}(T)$. Consider the i^{th} k-partition defined by Z, which is formed by the $z_{i+1} - z_i$ number of respondents with IDs $z_i, z_i + 1, \ldots, z_{i+1} - 1$, where $k \leq z_{i+1} - z_i \leq 2k - 1$. Note that each non-sensitive attribute in the k-partition will be generalized to an interval in the $\mathcal{K}(T)$. Moreover, the interval for a particular attribute is the same for all the data records in this k-partition. We use $lep(a_j, i)$ and $rep(a_j, i)$ to represent the left endpoint and right endpoint of the interval, for the attribute a_j ($1 \leq j \leq u$) in the i^{th} partition in the $\mathcal{K}(T)$, respectively. From the k-anonymization algorithm, we have:

$$\begin{aligned} lep(a_j, i) &= \min(a_j^{z_i}, a_j^{z_i+1}, \ldots, a_j^{z_{i+1}-1}) \\ rep(a_j, i) &= \max(a_j^{z_i}, a_j^{z_i+1}, \ldots, a_j^{z_{i+1}-1}) \end{aligned} \tag{7}$$

To find the minimum and maximum values of the set $\{a_j^{z_i}, a_j^{z_i+1}, \ldots, a_j^{z_{i+1}-1}\}$ by the $z_{i+1} - z_i$ respondents, we employ the unconditionally secure constant-rounds *SMPC* scheme in [22]. This *SMPC* scheme provides a set of protocols that compute the shares of a function of the shared values.

Based on the result of [22], we can define a primitive comparison function $\overset{?}{<} : \mathbb{F}_\delta \times \mathbb{F}_\delta \rightarrow \mathbb{F}_\delta$ for some prime δ, such that $(\alpha \overset{?}{<} \beta) \in \{0, 1\}$ and $(\alpha \overset{?}{<} \beta) = 1$ iff $\alpha < \beta$. This function securely compares two numbers α and β, and outputs if α is less than β. With this function, the maximum and minimum numbers in a set are easily found based on a series of pairwise comparisons.

To implement the primitive $\overset{?}{<}$, the owner of α (β *resp.*) creates a set of t_{ss} shares of α (β *resp.*) which is represented as $[\alpha]_\delta$ ($[\beta]_\delta$ *resp.*) based on a (t_{ss}, t_{ss}) *linear secret sharing* scheme, such as Shamir's. A set of t_{ss} respondents $DR_<$ (including the owners of α and β) are selected to be share recipients. Then the owner of α (β *resp.*) acts as a dealer and distribute t_{ss} shares of α (β *resp.*) to the respondents in $DR_<$ so that each respondent in $DR_<$ owns a share of α (β *resp.*) in $[\alpha]_\delta$ ($[\beta]_\delta$ *resp.*). With the shares of α (β *resp.*), the next step is to create shares of bits of α (β *resp.*) so that each bit of α (β *resp.*) is shared by the t_{ss} respondents in $DR_<$. The shares of bits can be computed with the bit-decomposition function in [22], and the shares of bits for α and β are represented as $[\alpha]_{bits}$=$[\alpha_0]_\delta, \ldots, [\alpha_\eta]_\delta$ and $[\beta]_{bits}$=$[\beta_0]_\delta, \ldots, [\beta_\eta]_\delta$, respectively, where η is the number of bits in α and β in their binary forms. Lastly, with the shares of bits, the shares of $(\alpha \overset{?}{<} \beta)$ can be computed with the bitwise less-than function in [22]. The set of shares of $(\alpha \overset{?}{<} \beta)$ is represented as $[\alpha \overset{?}{<} \beta]_\delta$, where each respondent in $DR_<$ owns one share in it. Therefore, the t_{ss} respondents in $DR_<$ can jointly compute the result of $(\alpha \overset{?}{<} \beta)$ without revealing extra information about the values of α and β.

The sub-protocol that uses the primitive comparison function $\overset{?}{<}$ to find the maximum and minimum values for the attribute a_j in the i^{th} k-partition is called $\mathcal{M}(a_j, i)$ with

Set Up

1. Choose two secure primes p and q, where $p = 2p' + 1$ and $q = 2q' + 1$ for random large primes p' and q' and $\gcd(N, \phi(N))$ for $N = p \cdot q$.

2. Let $g = (1+N)^e h^N \mod N^2$ for random e, h from $\mathbb{Z}_N^*$. Let ε be a random number in $\mathbb{Z}_N^*$, and $\theta = e\varepsilon\lambda$ for $\lambda = \text{lcm}(p-1, q-1)$.

3. Let $PK = (N, g, \theta)$ and $SK = \lambda$. sk_1, sk_2, ..., sk_x are shares of the $\varepsilon\lambda$ using (t_{ss}, x) Asmuth-Bloom secret sharing scheme.

Encryption function $E()$

For the j^{th} random number generated by the i^{th} respondent r_j^i, the ciphertext c is computed as:

$$c = E(r_j^i) = g^{r_j^i} r^N$$

for a random number $r \in \mathbb{Z}_N$.

Decryption function $D()$

Assume a set of t respondents DR_t form a coalition who jointly decrypt the ciphertext c. The t respondents jointly obtain the plaintext:

$$w = \frac{L(c^{\varepsilon\lambda} \mod N^2)}{\theta} \mod N \text{ (for } w \in \mathbb{Z}_N)$$

for the ciphertext c, where $L(\varsigma) = \frac{\varsigma - 1}{N}$ for $\varsigma \equiv 1 (\mod N)$. The readers may refer to [18] for the details of partial decryption by each respondent in DR_t.

TABLE I

THRESHOLD PAILLIER'S CRYPTOSYSTEM

the output $< lep(a_j, i), rep(a_j, i) >$. This sub-protocol is described as follows: first, each value in this set is shared using Shamir's (t_{ss}, t_{ss}) secret sharing. The shares are distributed via an *anonymous protocol* so that the identities of the shares' owners are not revealed. Second, with the shares, the pairwise comparison of values based on $\overset{?}{<}$ can be successfully constructed. The maximum and minimum values in $\{a_j^{z_i}, a_j^{z_i+1}, \ldots, a_j^{z_{i+1}-1}\}$ can be found with maximally $\left\lceil \frac{3 \cdot (z_{i+1} - z_i)}{2} \right\rceil - 2$ number of pairwise comparisons. Finally, the owners of the maximum value and minimum value publish their values of a_j anonymously and each respondent in the k-partition assigns the values of $lep(a_j, i)$ and $rep(a_j, i)$ accordingly.

For each non-sensitive attribute a_j ($1 \leq j \leq u$) and each k-partition i ($1 \leq i \leq \pi$), $\mathcal{M}(a_j, i)$ is run once. Therefore, the $\mathcal{M}$ sub-protocol runs for $\pi \cdot u$ rounds. Since the $\mathcal{M}$ sub-protocol runs independently within each k-partition, the sub-protocol can run simultaneously for each k-partition. By the end, the respondent j in the i^{th} k-partition submits the anonymized data record $\mathcal{K}(t_j)$=$\{[lep(a_1, i), rep(a_1, i)], \ldots, [lep(a_u, i), rep(a_u, i)], s_1, \ldots, s_v\}$ to the server. After collecting $\mathcal{K}(t_1)$, $\mathcal{K}(t_2)$, ..., $\mathcal{K}(t_x)$ from all x respondents, the final k-anonymized table $\mathcal{K}(T)$ is created and is returned to the collector.

V. THE PROTOCOL

In this section, we summarize the proposed k-anonymous data collection protocol. The presentation of the protocol follows the same order used in the last section. In addition, we present the key set up phase in the *preparation stage*. Table I shows the main steps in the threshold Paillier's cryptosystem [18], in which the $E()$ and $D()$ functions used in this paper are properly defined. The protocol is described as follows:

(s_0) **Key and data preparation:**

($s_{0.1}$) The public key PK, and the secret key SK are created following the setup procedure in Table I. sk_1, sk_2, ..., sk_x, sk_{svr} are shares of the private key SK based on the $(t_{ss}, x+1)$ Asmuth-Bloom secret sharing scheme.

($s_{0.2}$) *Input*:$\{t_1.A, t_2.A, \ldots, t_x.A\}$
Output:$\{c_1, c_2, \ldots, c_x\}$
Description: Each respondent maps his quasi-identifiers to an integer in $[1, c_{max}]$ using space filling curve, $c_i = \mathcal{S}(t_i.A)$. c_i is kept secret by the respondent i.

(s_1) **Probabilistic locality preserving mapping:**

($s_{1.1}$) *Input*: Random numbers from both the respondents and the server.
Output: $E(R_{init})$, $E(R_p)$
Description: The respondents and server jointly create two set of encrypted joint random numbers $E(R_{init})$ and $E(R_p)$.

($s_{1.2}$) *Input*: $E(R_p) = \{E(r_1), E(r_2), \ldots, E(r_{c_{max}})\}$
$E(R_{init}) = \{E(\iota_1), E(\iota_2), \ldots, E(\iota_b)\}$
Output: $E(R^+) = \{E(r_1^+), E(r_2^+), \ldots, E(r_{c_{max}}^+)\}$
Description: A set of encrypted random numbers is created based on Equation 6. The plaintexts of the encrypted numbers are sorted in ascending order according to the additive homomorphic property of Paillier's encryption.

($s_{1.3}$) *Input*: $E(R^+) = \{E(r_1^+), E(r_2^+), \ldots, E(r_{c_{max}}^+)\}$,
$S = \{c_1, c_2, \ldots, c_x\}$
Output: $E(\mathcal{F}(S)) = \{E(r_{c_1}^+), E(r_{c_2}^+), \ldots, E(r_{c_x}^+)\}$
Description: The respondent i retrieves the c_i^{th} item from the server's encrypted database $E(R^+)$ using private information retrieval, i.e. $E(r_{c_i}^+) = \mathcal{PIR}(c_i, E(R^+))$.

($s_{1.4}$) *Input*: $E(\mathcal{F}(S)) = \{E(r_{c_1}^+), E(r_{c_2}^+), \ldots, E(r_{c_x}^+)\}$
Output: $\mathcal{F}(S) = \{r_{c_1}^+, r_{c_2}^+, \ldots, r_{c_x}^+\}$
Description: Respondent i partially decrypts $E(r_{c_i}^+)$ with $t_{ss} - 2$ other respondents, and sends the partially decrypted cipher to the server for final decryption. The server obtains the value of $r_{c_i}^+$.

(s_2) **1D k-anonymization:**

($s_{2.1}$) *Input*: $\mathcal{F}(S) = \{r_{c_1}^+, r_{c_2}^+, \ldots, r_{c_x}^+\}$
Output: $Z = \{z_1, z_2, \ldots, z_\pi\}$
Description: 1D optimal k-anonymization algorithm is performed over $\mathcal{F}(S)$, and a description of k-partitions Z is created. z_i is the ending index of

the i^{th} k-partition.

(s_3) ***SMPC* of equivalence classes:**

($s_{3.1}$) *Input*:$T = \{t_1, t_2, \ldots, t_x\}$, $Z = \{z_1, z_2, \ldots, z_\pi\}$
Output: $\mathcal{K}(T) = \{\mathcal{K}(t_1), \mathcal{K}(t_2), \ldots, \mathcal{K}(t_x)\}$
Description:The same k-partitions Z is used for k-anonymization of T. Each respondent in the same k-partition use $\mathcal{M}$ sub-protocol to determine the generalized interval for each non-sensitive attribute. The k-anonymized data record $\mathcal{K}(t_i)$ from respondent i is submitted to the server anonymously to form $\mathcal{T}$.

VI. Analysis

In this section, we first analyze the information leakage during the execution of the protocol. We show that, the information leakage is equivalent $\mathcal{L}_{svr}$ for the server and $\mathcal{L}_i$ for respondent i. Second, we analyze the probability of correctly guessing the quasi-identifiers of a victim given its mapped image in $\mathcal{F}(S)$, which is described by the γ-concealing in this paper. Third, we analyze the time complexity of each stage of the protocol. Last, we present a complexity metric identifying the required number of online respondents during the execution of the protocol, which shows the flexibility of the proposed protocol.

A. Information Leakage

Theorem 2: The k-anonymous data collection protocol only leaks $\mathcal{L}_{svr}$ for the server and $\mathcal{L}_i$ for the respondent i, where $\mathcal{L}_{svr} = \{\mathcal{RG}_{svr}, \mathcal{F}(S)\}$ and $\mathcal{L}_i = \{\mathcal{RG}_i\}$.

Proof: We first construct the simulator M_{svr} for the server. In stage $s_{1.1}$, the knowledge of the server is described by $\mathcal{RG}_{svr}$, in which the server knows the range of each of the random numbers in $E(R)$ and $E(R_{init})$. Each joint encrypted random number in $E(R)$ and $E(R_{init})$ in the view of the server can be simulated by M_{svr} by multiplying an encrypted random number in the range of $[\rho_{min}, \rho_{max}]$ to the encrypted random number contributed by the server. In stage $s_{1.2}$, the $E(R^+)$ is constructed based on $E(R)$ and $E(R_{init})$, where no information is leaked during the computation based on the semantic security of the additive homomorphic property of the Paillier's encryption. Therefore, M_{svr} simulates $E(R^+)$ based on the simulations of $E(R)$ and $E(R_{init})$. In stage $s_{1.3}$, the server gains no information about the retrieved item which is guaranteed by the property of $\mathcal{PIR}()$ function. The decrypted value in stage $s_{1.4}$ is $\mathcal{F}(S)$, which is part of the knowledge of the server. In stage $s_{2.1}$, the input is based on $\mathcal{F}(S)$, therefore the server does not gain any additional information. In stage $s_{3.1}$, the server receives the k-anonymized tuples from the respondents, the received data records are equivalent to the knowledge of the server $\mathcal{K}(T)$.

Now, we construct the simulator M_i for the respondent i. In stage $s_{1.1}$, the knowledge of respondent i is described by $\mathcal{RG}_i$, in which he knows the range of joint random numbers which are jointly created by him and the server. The respondent is not participating in stage $s_{1.2}$. In stage $s_{1.3}$, M_i simulates the retrieved ciphertext by a random ciphertext. In stage $s_{1.5}$, M_i simulates the partially decrypted message by partially decrypted the random ciphertext. The respondent is not participating in stage $s_{2.1}$. In stage $s_{3.1}$, the secret shares and messages can be simulated by M_i using random ciphers, guaranteed by the function sharing algorithm in [22]. The output is equivalent to the knowledge of the respondent $\mathcal{K}(T)$. ■

B. γ-concealing Property

A property explaining how well the mapped value $r^+_{c_i}$ hides the value c_i is described by the notion of γ-concealing. In this part, we analyze the relation of γ-concealing property with other parameters.

Suppose that the adversary is targeting respondent i (victim), and wants to guess the value of c_i based on the value of $r^+_{c_i}$. The value of $1-\gamma$ (the probability the adversary can guess c_i correctly based on $r^+_{c_i}$) can be approximated as follows: with $r^+_{c_i}$, the Maximum Likelihood Estimation of c_i is $c_{mle} = \left\| r^+_{c_i}/\mu \right\| - b$ (i.e. c_{mle} =roundup($r^+_{c_i}/\mu$)$-b$). The adversary can find the value of c_i with the Maximum Likelihood Estimation successfully only when $c_i = c_{mle}$. However, the condition for $c_i = \left\| r^+_{c_i}/\mu \right\| - b$ is equivalent to the condition for $r^+_{c_i}$ to be in the range of $[(c_i - \frac{1}{2} - b)\mu, (c_i + \frac{1}{2} - b)\mu]$. Therefore, we can establish the following equivalence:

$$\Pr(c_{mle} = c_i | r^+_{c_i}) = \Pr(r^+_{c_i} \in [(c_i - \frac{1}{2} - b)\mu, (c_i + \frac{1}{2} - b)\mu]) \tag{8}$$

The probability value on the r.h.s of the above equation can be approximated using the *central limit theorem*. According to the *central limit theorem*, $r^+_{c_i}$ is approximately normally distributed with $r^+_{c_i} \sim N((c_i + b)\mu, (c_i + b)\sigma^2)$. Thus, the following approximation holds:

$$\begin{aligned} 1 - \gamma \approx \quad & \Phi_{(c_i+b)\mu,(c_i+b)\sigma^2}[(c_i + \tfrac{1}{2} - b)\mu] \\ & -\Phi_{(c_i+b)\mu,(c_i+b)\sigma^2}[(c_i - \tfrac{1}{2} - b)\mu] \end{aligned} \tag{9}$$

In the above equation, $\Phi_{(c_i+b)\mu,(c_i+b)\sigma^2}$ is the distribution function of a normal distribution with mean $(c_i + b)\mu$, and variance $(c_i+b)\sigma^2$. The equation shows that, the value of $1-\gamma$ relies on the values of μ, σ^2, b and c_i. Moreover, according to the property of continuous normal distribution, the value $1-\gamma$ increases with increasing μ and decreases with increasing σ^2, b and c_i. Hence, the protocol tends to be secure when large σ^2, b, and c_i values, and small μ value are used.

While the μ, σ^2 and b are the system parameters, c_i is the parameter of respondents which are different among respondents. Since $1-\gamma$ decreases with increasing c_i, by setting the c_i value to be minimum (i.e. $c_i = 0$) we can find the maximum value of $1-\gamma$. Therefore, the maximum value of $1-\gamma$ is:

$$(1-\gamma)_{max} \approx \Phi_{b\mu,b\sigma^2}[(\frac{1}{2} - b)\mu] - \Phi_{b\mu,b\sigma^2}[(\frac{1}{2} - b)\mu] \tag{10}$$

The value of $(1-\gamma)_{max}$ can be viewed as a system-wide security metric of the protocol.

C. Complexity Analysis

Now, we analyze the time complexity of the proposed protocol for both the respondents and the server. We assume that each Paillier's encryption operation or each partial decryption operation (with a secret share) takes a single unit time. The analysis follows the stages of the protocol execution as described in Section V: in stage $s_{1.1}$, to generate the set of joint encrypted random numbers $E(R_p)$, each respondent needs to generate $\frac{c_{max}}{x}$ encrypted random numbers, which takes $\mathcal{O}\left(2\frac{c_{max}}{x}\right)$ time for each respondent. When generating $E(R_{init})$, in the worst case, all the b random encrypted numbers are generated by a single respondent. Therefore, the time complexity for the respondent is $\mathcal{O}(2b)$. On the server side, for each random number generated by a respondent, the server needs to generate a random number, perform an encryption, and perform a multiplication of cipertexts. Therefore, the server side time complexity for creating $E(R_p)$ and $E(R_{init})$ is $\mathcal{O}(3(c_{max}+b))$. The homomorphic addition of ciphertexts in stage $s_{1.2}$ takes $\mathcal{O}(c_{max})$ time for the server. In stage $s_{1.3}$, the communication complexity of retrieving ϕ bits of data from the server is $O(t_{pir}+\phi)$, where t_{pir} is a security parameter that satisfies $t_{pir} \geq \log(\phi c_{max})$. If the retrievals by all the respondents are executed sequentially, the total time is in $\mathcal{O}(x(t_{pir}+\phi))$. In stage $s_{1.4}$, the total number ciphertexts to be decrypted is x, where each ciphertext requires $t_{ss}-1$ number of partial decryptions from the respondents. Therefore, the time complexity of all the decryption operations for a respondent is $\mathcal{O}(t_{ss}-1)$. The server needs to perform $\mathcal{O}(x)$ number of partial decryptions in this stage. In stage $s_{2.1}$, the time complexity for optimal 1D k-anonymization based on *FALL* is $\mathcal{O}(k^2x)$. If the 1D optimal k-anonymization algorithm based on the graph shortest path [20] is used, the time complexity is $\mathcal{O}(\max(x\log(x), k^2x))$. Lastly, in stage $s_{3.1}$, the time complexity for each respondent can be computed by the total number of $\mathcal{M}$ execution multiplies the complexity of an execution of $\mathcal{M}$ and divide by the total number of respondents. The total number of $\mathcal{M}$ execution is πu. For each $\mathcal{M}$ execution, the complexity can be computed by the $\mathcal{O}(k)$ number of comparisons times t_{ss} number of partial decryptions of each comparison. Combining all, the time complexity for each respondent in this stage is $\mathcal{O}(\frac{\pi u k t_{ss}}{x})$. Since $\frac{\pi k}{x} \approx 1$, the complexity can be simplified to $\mathcal{O}(ut_{ss})$.

D. The Required Number of Online respondents

In the proposed protocol, only three operations require collaboration from multiple respondents, while other operations are independently done by each respondent. These operations include joint decryption of a ciphertext, joint computation of a pairwise comparison in $\mathcal{M}$ algorithm, and finding the minimum and maximum values for a particular attribute within a k-partition. Among these three operations, the first two operations require $\mathcal{O}(t_{ss})$ number of respondents to be online simultaneously, and the last operation requires $\mathcal{O}(k)$ respondents to be online simultaneously. Therefore, the complexity is $\mathcal{O}(\max(k, t_{ss}))$.

ρ_{min}	ρ_{max}	μ	σ^2	$1-\gamma$
100	200	150	833.333	0.119235
50	250	150	3333.33	0.0597853
0	300	150	7500	0.0398776
100	400	250	7500	0.0664135
150	450	300	7500	0.0796557
200	500	350	7500	0.0928758

TABLE II

γ-CONCEALING PROPERTY WITH $b = 200$ AND $c_i = 100$

b	c_i	$1-\gamma$
200	100	0.0796557
200	200	0.0690126
200	300	0.0617421
200	0	0.0974767
300	0	0.0796557
400	0	0.0690126

TABLE III

γ-CONCEALING PROPERTY WITH $\rho_{min} = 100$ AND $\rho_{max} = 300$

VII. Experimental Evaluation

In this section, we carry out several experiments to evaluate the performance of the proposed k-anonymous data collection protocol. The experiments are divided into three parts: in the first part, we evaluate the γ-concealing property of the proposed protocol. In the second part, we evaluate the *probabilistic distance preserving* property in the proposed protocol due to its importance in utility preservation. In the third part, we evaluated the performance of the protocol in utility preservation. In order to compare with *FALL* – the k-anonymization algorithm that the proposed protocol is based on, we employ the utility metric *GCP*. For the details of how *GCP* is defined, the readers may refer to [4].

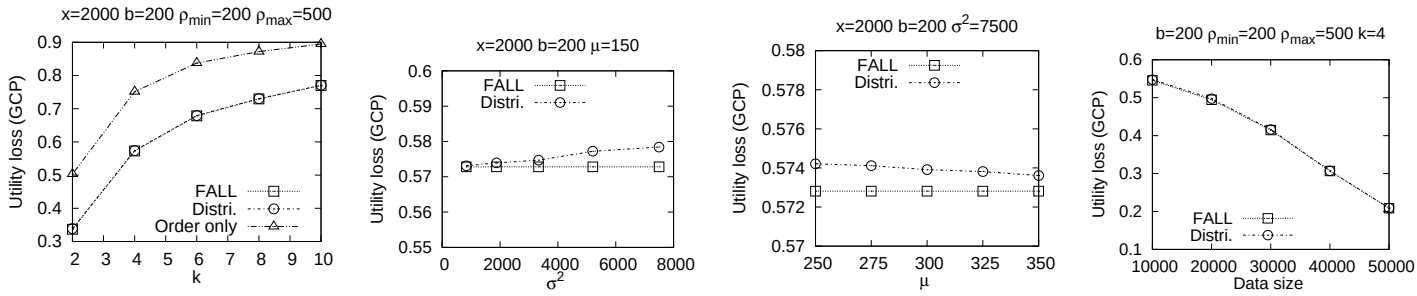
The dataset that we use for the experiments is from the website of Minnesota Population Center (*MPC*)[2], which provides census data over various locations through different time periods. For the experiments, we have extracted 1% sample USA population records with attributes *age*, *sex*, *marital status*, *race*, *occupation* and *salary* for the year 2000. The dataset contains $2,808,457$ number of data records, however, we only use a subset of these records. Among the six attributes,

[2]http://www.ipums.org/

ρ_{min}	ρ_{max}	μ	σ^2	*DPR*
100	200	150	833.333	0.999525
50	250	150	3333.33	0.999174
0	300	150	7500	0.998411
100	400	250	7500	0.999223
150	450	300	7500	0.999438
200	500	350	7500	0.999536

TABLE IV

DISTANCE PRESERVING RATIO WITH $b = 200$



(a) Change k (b) Change μ (c) Change σ^2 (d) Change data size

Fig. 5. Utility preservation evaluation

the age is numerical data while others are categorical data. For the categorical data, we can use taxonomy trees (e.g. [23], [24]) to convert a categorical data to numerical data for generalization purposes. Among all the seven attributes, the *salary* is considered as the sensitive attribute, while others are non-sensitive and are considered as quasi-identifiers. The domain sizes for *age*, *sex*, *marital status*, *race* and *occupation* are 80, 2, 6, 9, and 50, respectively. The programs for the experiments are implemented in Java and run on Windows XP PC with $4.00\ GB$ memory and Intel(R) Core(TM)2 Duo CPU each at $2.53\ GHz$.

A. Evaluation of γ-concealing Property

In this part of experiments, we compute some real values of $1-\gamma$ with some predefined parameters based on the formulas in Equation 9, to show that the proposed protocol is privacy preserving. The Table II shows the result of how the value of $1-\gamma$ changes with the value of μ and σ^2 (respectively the mean and variance of the uniform distribution). In the first three rows of Table II, we keep the value of μ constant ($\mu = 150$) while increasing the value of σ^2. Notice that the value of $1-\gamma$ decreases with increasing σ^2. In the last three rows of Table II, we keep the values of σ^2 constant ($\sigma^2 = 7,500$) instead, and increase the values of μ. Notice in this case that the value of $1-\gamma$ increases with increasing μ. In Table III, we experimented how the value of $1-\gamma$ changes with the value of b and c_i. In the first three rows of Table III, we keep the value of b constant ($b = 200$) and increase the value of c_i. We find that the value of $1-\gamma$ decreases with increasing c_i. In the last three rows of Table III, we keep the value of c_i constant ($c_i = 0$) and increase b. It is true that the value of $1-\gamma$ decreases with increasing b. Since the minimum c_i is 0, the last three rows of Table III shows the maximum values of $1-\gamma$ (following Equation 10) under different values of b.

In this set of experiments, the values of $1-\gamma$ are all below 0.1 which supports the level of privacy that a respondent can hide his quasi-identifiers with probability at least 90% in the process of data collection. For stronger privacy protection, we can further lower the value of $1-\gamma$, by either decreasing the value of μ or increasing the value of σ^2 or b.

B. Evaluation of Distance Preserving Mapping

The property of *probabilistic distance preserving* of the mapping function $\mathcal{F}()$ is very critical to utility preservation. For the purpose of hiding the quasi-identifiers of respondents, in the proposed $\mathcal{F}()$, we do not achieve strict *relative distance preserving*. However, in this part of experiments, we show that the proposed mapping function $\mathcal{F}()$ can quite well preserve the *relative distance*. For this purpose, we propose the *Distance Preserving Ratio* (DPR) metric, which measures the quality of *relative distance preserving* mapping. Given a set of pre-images $\{c_1, c_2, \ldots, c_x\}$, and the set of images $\{\mathcal{F}(c_1), \mathcal{F}(c_2), \ldots, \mathcal{F}(c_x)\}$. A *relative distance preserving triple* (*RDPT*), is a combination of three pre-images $< c_{i_1}, c_{i_2}, c_{i_3} >$ whose images $< \mathcal{F}(c_{i_1}), \mathcal{F}(c_{i_2}), \mathcal{F}(c_{i_1}) >$ preserve their relative distances. The DPR is defined as follows:

$$DPR = \frac{\text{total no. of } \textit{RDPT} \ < c_{i_1}, c_{i_2}, c_{i_3} >}{\text{total no. of triples } C(x, 3)} \qquad (11)$$

Naturally, the DPR describes the ratio between the number of triples of pre-images whose mapping preserve relative distances and the total number of triples in the set of pre-images. The computation of exact value of DPR requires the enumeration of all triples of pre-images and images, which is feasible when the size of pre-images (or images) is relatively small. However, when the size of the pre-images (or images) is large (e.g. millions or above), we can use sampling techniques to estimate the value of DPR by randomly selecting a fixed number (supposed to be large) of samples of triples and then compute the DPR based on the samples.

In the experiments, we randomly select 2,000 data records from the dataset. We convert the non-sensitive attributes of selected data records into a set of integers using Hilbert curve, and input it to $\mathcal{F}()$ as the set of pre-images. The set of parameters used is the same as the one used in the experiments for γ-concealing property. In Table IV, we see that when μ is fixed to 150, the value of DPR decreases with increasing of σ^2. On the other hand, when we fix the value of σ^2 to be $7,500$, the value of DPR increases with μ. In other words, large μ and small σ^2 has positive impacts on *relative distance preserving*. In all cases, the values of DPR are extremely high (almost close to 1), which clearly show that the mapping function $\mathcal{F}()$ achieves excellent *relative distance preserving*.

C. Evaluation of Utility Preservation

Lastly, we evaluate the utility preservation property of the proposed protocol by measuring the utility loss (the *GCP* metric) against several parameters. The set of data records used in the first three experiments is the same set of 2,000 data records used in the last part of the experiments.

In the first experiment, we measure the *GCP* value against increasing k. The parameters that we use are $b = 200$, $\rho_{min} = 200$ and $\rho_{max} = 500$. Figure 5.a shows that the value of *GCP* increases with increasing k (as expected). Moreover, the *GCP* value computed based on table created by *FALL* (as labeled) and the proposed protocol (labeled as *Distr.*) are almost the same, showing that our approach can achieve almost the same level of utility preservation as the *FALL*. A naive method (labeled as *Order only*), which only sorts the respondents in 1D space and group every consecutive k respondents, results in much higher *GCP* values compared to *FALL* and our approach.

Figure 5.b shows the utility loss for both *FALL* and the proposed protocol with increasing σ^2. Though from Figure 5.a, the curve of utility loss for *FALL* and the proposed protocol appear to be overlapping, when we focus the *GCP* values in the interval of [0.55, 0.6] in Figure 5.b, we indeed observe that the performance of the proposed protocol in utility preservation is slightly less optimal compare to *FALL*. Moreover, the Figure 5.b shows that the *GCP* value based on the proposed approach increases with increasing σ^2 at relatively slow rate. Similarly, Figure 5.c shows that increasing μ value helps to reduce the *GCP* value.

In Figure 5.d, in order to evaluate how the *GCP* value changes with the data size, we increase the data size from 10,000 to 50,000. It shows that the *GCP* value for both *FALL* and the proposed approach decreases at similar rate with increasing data size. The decreasing of *GCP* value is due to the fact that when data size increases, the density of data also increases. To conclude this part, these experiments show that with appropriate parameters, the proposed approach achieves almost the same utility preservation performance as *FALL*. Large σ^2 and large μ have negative and positive impacts over the utility, respectively.

VIII. Conclusions

In this paper, we proposed a k-anonymous data collection protocol under the assumption that the data collector is not trustworthy. With the protocol, the collector receives a k-anonymized table generalized from the data records of the respondents without seeing the original data records. The protocol is designed to leak certain information in order to reduce the communication and computation cost that otherwise are intractable. However, we show that the privacy threat caused by the information leakage is limited and guaranteed by the γ-concealing property. Moreover, we show that the utility of the k-anonymized table produced via the proposed protocol is almost as good as in the case of a trustworthy collector. In the future, we plan to extend our protocol to l-diversity and t-closeness.

References

[1] L. Sweeney, "k-anonymity: A model for protecting privacy," *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[2] P. Samarati and L. Sweeney, "Generalizing data to provide anonymity when disclosing information (abstract)," in *Proc. of ACM PODS*, 1998, p. 188.

[3] S. Zhong, Z. Yang, and R. N. Wright, "Privacy-enhancing k-anonymization of customer data," in *PODS '05: Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. New York, NY, USA: ACM, 2005, pp. 139–147.

[4] G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis, "Fast data anonymization with low information loss," in *Proc. of VLDB*, 2007, pp. 758–769.

[5] A. C. Yao, "Protocols for secure computations," in *SFCS '82: Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*. Washington, DC, USA: IEEE Computer Society, 1982, pp. 160–164.

[6] A. C.-C. Yao, "How to generate and exchange secrets," in *SFCS '86: Proceedings of the 27th Annual Symposium on Foundations of Computer Science*. Washington, DC, USA: IEEE Computer Society, 1986, pp. 162–167.

[7] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game," in *STOC '87: Proceedings of the nineteenth annual ACM symposium on Theory of computing*. New York, NY, USA: ACM, 1987, pp. 218–229.

[8] A. Meyerson and R. Williams, "On the complexity of optimal k-anonymity," in *PODS '04: Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. New York, NY, USA: ACM, 2004, pp. 223–228.

[9] W. Jiang and C. Clifton, "A secure distributed framework for achieving k-anonymity," *The VLDB Journal*, vol. 15, no. 4, pp. 316–333, 2006.

[10] P. Jurczyk and L. Xiong, "Privacy-preserving data publishing for horizontally partitioned databases," in *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*. New York, NY, USA: ACM, 2008, pp. 1321–1322.

[11] B. Moon, H. v. Jagadish, C. Faloutsos, and J. H. Saltz, "Analysis of the clustering properties of the hilbert space-filling curve," *IEEE Trans. on Knowl. and Data Eng.*, vol. 13, no. 1, pp. 124–141, 2001.

[12] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Mondrian multidimensional k-anonymity," in *Proc. of ICDE*, 2006.

[13] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W.-C. Fu, "Utility-based anonymization using local recoding," in *KDD '06*. New York, NY, USA: ACM, 2006, pp. 785–790.

[14] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," in *Proc. of ICDE*, 2006.

[15] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *Proc. of ICDE*, 2007, pp. 106–115.

[16] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[17] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," 1999, pp. 223–238.

[18] K. Kaya and A. A. Selçuk, "Threshold cryptography based on asmuth-bloom secret sharing," *Inf. Sci.*, vol. 177, no. 19, pp. 4148–4160, 2007.

[19] C. Asmuth and J. Bloom, "A modular approach to key safeguarding," *IEEE Trans. Information Theory*, pp. 29(2):208–210, 1983.

[20] S. L. Member-Hansen and S. Member-Mukherjee, "A polynomial algorithm for optimal univariate microaggregation," *IEEE Trans. on Knowl. and Data Eng.*, vol. 15, no. 4, pp. 1043–1044, 2003.

[21] C. Gentry and Z. Ramzan, "Single-database private information retrieval with constant communication rate," 2005, pp. 803–815.

[22] I. Damgard, M. Fitzi, E. Kiltz, J. Nielsen, and T. Toft, "Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation," 2006, pp. 285–304.

[23] R. Bayardo and R. Agrawal, "Data privacy through optimal k-anonymization," in *Proc. of ICDE*, 2005, pp. 217–228.

[24] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: Efficient full-domain k-anonymity," in *Proc. of ACM SIGMOD*, 2005, pp. 49–60.