Hindawi
Security and Communication Networks
Volume 2017, Article ID 9532163, 10 pages
https://doi.org/10.1155/2017/9532163

Research Article

Practical *m*-*k*-Anonymization for Collaborative Data Publishing without Trusted Third Party

Jingyu Hua,[1] An Tang,[2] Qingyun Pan,[2] Kim-Kwang Raymond Choo,[3] Hong Ding,[4,5] and Yizhi Ren[4,5]

[1]*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China*
[2]*Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China*
[3]*Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, TX 78249-0631, USA*
[4]*School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China*
[5]*Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, China, Hangzhou Dianzi University, Hangzhou 310018, China*

Correspondence should be addressed to Hong Ding; dinghong@hdu.edu.cn and Yizhi Ren; renyizhi@gmail.com

Received 6 October 2016; Revised 9 February 2017; Accepted 27 February 2017; Published 16 March 2017

Academic Editor: Willy Susilo

In collaborative data publishing (CDP), an *m*-adversary attack refers to a scenario where up to *m* malicious data providers collude to infer data records contributed by other providers. Existing solutions either rely on a trusted third party (TTP) or introduce expensive computation and communication overheads. In this paper, we present a practical distributed *k*-anonymization scheme, *m*-*k*-anonymization, designed to defend against *m*-adversary attacks without relying on any TTPs. We then prove its security in the semihonest adversary model and demonstrate how an extension of the scheme can also be proven secure in a stronger adversary model. We also evaluate its efficiency using a commonly used dataset.

1. Introduction

In today's interconnected society, our sensitive personal data are increasingly stored in various databases belonging to different online service providers. Although online service providers have the duty and vested interest to ensure the security and privacy of user data, there are instances where user data are shared or compromised. For instance, a small to medium sized online service provider may wish to mine user purchasing patterns in order to fine-tune their marketing strategy and improve sales. Such (data mining) task is likely to be outsourced to a third-party marketing company; thus, the records in the online service provider's database will be shared with the third-party. In such a scenario, the online service provider requires a privacy-preserving data publishing (i.e., sharing) approach to ensure that the data is shared without breaching user privacy.

If the records to be published are owned by a single provider, the provider can easily run algorithms, such as [1, 2] which implement *k-anonymity* [3] (a widely used privacy protection mechanism), to anonymize the data prior to publishing. *k*-anonymity is proposed to solve such problems, that is, when a data owner or provider wants to publish parts of its data which is related to some specific persons, how can it guarantee that these persons cannot be reidentified while the remaining parts of the data are still practically useful? Consider a table with n rows and m columns; each row of the table represents a record relating to a specific object and each column represents an attribute of each object. Some nonprivate attributes can be considered as quasi identifier (QI). A table satisfying *k*-anonymity means QI of each tuple contained in this table appears at least k times [3].

However, data in the real world is unlikely to originate from a single provider. Solutions seeking to address such a scenario are known as *collaborative privacy-preserving data publishing* (CPPDP). CPPDP has received considerable attention in recent years (e.g., [4–11]). A straightforward solution is for all providers to outsource their data to a

TTP, who will assume control of the data as if the TTP is publishing its own data. An alternative approach uses Secure Multiparty Computation (SMC) [12], which allows providers to collaboratively compute preferred functions upon the complete dataset without revealing private data [4, 5].

Although these schemes could guarantee that anonymized data satisfies k-anonymity against outsider attackers, malicious providers (i.e., insider attackers) or vendors who have access to the provider's systems may collude to invalidate k-anonymity by excluding their own data. Let us now consider the example described in [13], where in Table 1, T_1, T_2, T_3, T_4 are databases of four hospitals. In this example, four hospitals wish to collaboratively publish their dataset without revealing their patients' privacy (e.g., medical diagnosis). Of the nonsensitive attributes ($\{Name, Age, Zip\}$), $\{Age, Zip\}$ can be considered a quasi identifier (QI). QI refers to a set of attributes that are not unique identifiers in themselves but can be corrected to uniquely identify most tuples in the dataset [3]. T_a^* is an anonymized dataset satisfying 2-anonymity. Each QI group includes 3 records; therefore, each QI tuple appears at least 2 times in every group. From $\{Age, Zip\}$, we cannot infer the disease information of different patients. However, if P_1 is an adversary, it could remove all its data from T_a^*. Thus, the only record in this group is provided by P_3, and the remaining data no longer satisfies 2-anonymity. Therefore, the attributes of disease can be achieved easily. For instance, P_1 can link other datasets with the remaining part of first QI group, such as a voter list or another dataset which contains $\{Name, Age, Zip$, etc.$\}$. This would enable us to infer the disease of *Sara* by linking two datasets together. In reality, there could exist more than one adversary. m providers, for instance, might collude to remove their data to infer records contributed by other providers. This is known as the *m-adversary problem* in the literature.

Seeking to address the m-adversary problem, Goryczka et al. [13] introduce the concept of *m-privacy* which is the focus of this paper. More specifically, we focus on m-privacy with respect to k-anonymity, which is referred to as m-k-anonymity in the remainder of this paper. Suppose that the total number of providers participating in the collaborative data publishing is n, and the published data after collaborative anonymization satisfies m-k-anonymity if, and only if, subdata from any $n - m$ providers satisfies k-anonymity. When m adversaries remove all their data, the remaining data in the table still satisfies k-anonymity. The corresponding anonymization process is the m-k-anonymization, and as an example, T_b^* is an anonymized dataset satisfying 1-2-anonymity. From the table, we can see that each QI group in T_b^* contains 3 records. Even if one of the providers is an adversary, that is, P_1 is an adversary and it removes all its data from the first QI group, the remaining two records in the first QI group of T_b^* still satisfy 2-anonymity.

Goryczka et al. present a TTP-dependent CPPDP scheme that achieves m-k-anonymity. However, TTP does not always exist in the real world [14, 15]. This is especially true in aftermath Edward Snowden's revelation that the US Government has been conducting large-scale government surveillance (http://masssurveillance.info/). They then present a SMC variant of this scheme based on a series of cryptographic protocols (e.g., secure sum, secure comparison, and secure size of set union) to remove the need for a TTP. However, this variant is not specially designed for m-k-anonymity, and the constituent cryptographic protocols are too time consuming to be practical for real-world deployment.

Table 1: m-adversary and m-privacy example.

(a) T_1

Name	Age	Zip	Disease
Alice	24	98745	Cancer
Bob	35	12367	Epilepsy
Emily	22	98712	Asthma

(b) T_2

Name	Age	Zip	Disease
Olga	32	98701	Cancer
Mark	37	12389	Flu
John	31	12399	Flu

(c) T_3

Name	Age	Zip	Disease
Sara	20	12300	Epilepsy
Cecilia	39	98708	Flu

(d) T_4

Name	Age	Zip	Disease
Olga	32	98701	Cancer
Frank	33	12388	Asthma

(e) T_a^*

Providers	Name	Age	Zip	Disease
P_1	Alice	[20–30]	* * * * *	Cancer
P_1	Emily	[20–30]	* * * * *	Asthma
P_3	*Sara*	*[20–30]*	* * * * *	*Epilepsy*
P_2	John	[31–34]	* * * * *	Flu
P_2, P_4	Olga	[31–34]	* * * * *	Cancer
P_4	Frank	[31–34]	* * * * *	Asthma
P_1	Bob	[35–40]	* * * * *	Epilepsy
P_2	Mark	[35–40]	* * * * *	Flu
P_3	Cecilia	[35–40]	* * * * *	Flu

(f) T_b^*

Providers	Name	Age	Zip	Disease
P_1	Alice	[20–40]	* * * * *	Cancer
P_2	Mark	[20–40]	* * * * *	Flu
P_3	Sara	[20–40]	* * * * *	Epilepsy
P_1	Emily	[20–40]	987 * *	Asthma
P_2, P_4	Olga	[20–40]	987 * *	Cancer
P_3	Cecilia	[20–40]	987 * *	Flu
P_1	Bob	[20–40]	123 * *	Epilepsy
P_4	Frank	[20–40]	123 * *	Asthma
P_2	John	[20–40]	123 * *	Flu

In this paper, we propose a TTP-independent CPPDP scheme designed to achieve m-k-anonymity in a more efficient manner. We observe that the process of k-anonymity involves no sensitive attributes, and hence, we divide our scheme into two phases. Firstly, we use a centralized server which is not required to be trusted to aggregate nonsensitive components (i.e., QI attributes) of records from each provider and anonymize these components to ensure m-k-anonymity. Secondly, we design a distributed privacy-preserving method to aggregate values of sensitive attributes for each equivalence group without breaching m-k-anonymity. In other words, we present a practical two-phase CPPDP scheme without the need for a TTP, before demonstrating that the proposed scheme achieves m-k-anonymity in the widely accepted semihonest model. In the event that providers are malicious and attempt to modify user data, users are unlikely to find out about the tampering. Therefore, we present an effective sampling-based defense strategy against such an attack. We then evaluate the time efficiency of the proposed scheme with a public dataset including 45222 records. Our evaluations demonstrate that the time overhead increases linearly with n, which is reasonable for an offline scheme.

2. Related Work

Recent trends in big data and cloud computing have partly contributed to renewed interest in privacy-preserving data publishing [16–19]. Existing literature on privacy-preserving data publishing can be broadly classified into the following categories.

Single Provider. Most existing research focus on the scenario involving a single data owner wanting to publish its own data, such as k-anonymity [3], l-diversity [20], and t-closeness [21]. Of these privacy models, k-anonymity has the longest history, and many efficient algorithms have been developed to implement k-anonymity. Examples of a bottom-up generalization approach and a top-down specification approach is Incognito [2] and Mondrian [1], respectively.

Collaborative Data Publishing. In the collaborative data publishing literature, the focus is on privacy-preserving algorithms for distributed setups. For example, Jiang and Clifton [6] propose a protocol implementing k-anonymity on a vertically partitioned dataset. The protocol presented in [7] is designed to extract anonymized data from a set of providers, which is then published to the miner. Jiang and Clifton [8] present a SMC framework for data sharing between two untrusted parties, and Jurczyk and Xiong [9] present several decentralized protocols to ensure the user's privacy during the querying of multiple databases. Mohammed et al. [10] seek to address the privacy-preserving problem in a specific application of data mashup in the web, which is a typical distributed scenario. The authors also present distributed algorithms to integrate healthcare data [11]. The protocol presented in [5] allows protocol participants decide in advance whether its utility is acceptable prior to execution.

Insider Attackers. Goryczka et al. [13] present the m-adversary problem, where data providers are considered as potential attackers. To address such a threat, they propose the m-privacy model and present an efficient and effective TTP-based anonymization scheme. A key limitation of the scheme is the need for a TTP, which is not always available in the real world. Therefore, a SMC-based variant which does not rely on TTP is proposed by Goryczka et al. However, as noted by the authors, the SMC scheme is only a conceptual scheme that is not practical for real-world deployment due to the significant time overhead of the underlying protocols. This is the gap we seek to address in this paper, by presenting a practical SMC-based m-privacy implementation.

3. Problem Definition

Let $P = \{P_1, P_2, \ldots, P_n\}$ be the set of all data providers, who own a set of records, T_i. T, defined as $T = \{T_1, T_2, \ldots, T_n\}$, is the set of all records. Providers aim to collaboratively publish the dataset T while preventing attackers from identifying records of individuals. In such a distributed environment, the providers may not trust each other. In other words, none of the providers can be considered as a TTP, and the publisher (considered as a separate party) is also not a trusted party.

Definition 1 (k-anonymity in CPPDP). Given n providers, $P_1, P_2, \ldots, P_n$, each with a dataset, $T_1, T_2, \ldots, T_n$ (T_i owned by P_i), the result set of CPPDP satisfies k-anonymity if, and only if, each QI group contains at least k records: that is, $\forall i\ (1 \le i \le n_g)$, $|\mathrm{QI}_i| \ge k$, where n_g is the number of QI groups.

k-Anonymity can prevent external adversaries from inferring sensitive attributes with QIs. However, k-anonymity is unable to address the m-adversary problem, as malicious providers (i.e., insiders) may collude to remove their own data to violate the k-anonymity of the remaining data. Targeting this problem, we define a new privacy model which we coined *m-k-anonymity*. The *m-k-anonymity* model is adapted from the *m-privacy model* of Goryczka et al. [13]. We denote the set of records owned by all adversaries $P_A = \{P_{a_1}, P_{a_2}, \ldots, P_{a_m}\}$ to be T_A; that is, $T_A = \bigcup_{P_i \in P_A} T_i$. In other words, $T_A(\mathrm{QI}_i)$ is the set of those records owned by any adversary in group QI_i.

Definition 2 (m-k-anonymity). Given a set of n providers, $P = \{P_1, P_2, \ldots, P_n\}$, m adversaries, and $P_A = \{P_{a_1}, P_{a_2}, \ldots, P_{a_m}\}$, the result satisfies m-k-anonymity if and only if, $\forall P_A \subset P$, $|P_A| = m$, $\forall i\ (1 \le i \le n_g)$, $|\mathrm{QI}_i| - |T_A(\mathrm{QI}_i)| \ge k$.

It means that, for every equivalence group in the anonymized dataset, its size excluding the number of records owned by any m providers must be larger than k. In other words, after m colluding providers have removed their records, each group must contain more than k records.

Design Goal. An efficient and practical CPPDP scheme providing m-k-anonymity without involving a TTP is designed.

4. Two-Phase Scheme Providing m-Privacy

4.1. Semihonest Model. After defining m-k-anonymity in the preceding section (see Definition 2), we will now present our

(1) **if** no allowable multidimensional cut for partition **then**
(2) **return** ϕ: partition $\rightarrow$ statistic summary
(3) **else**
(4) dim $\leftarrow$ chooseDimension()
(5) $\text{fs}_1, \ldots, \text{fs}_{n_g-1} \leftarrow$ frequencySet(partition, dim)
(6) $\text{splitVal}_1, \ldots, \text{splitVal}_{n_g-1} \leftarrow$ findMedian($\text{fs}_1, \ldots, \text{fs}_{n_g-1}$)
(7) $\text{QI}_1 \leftarrow \{t \in \textit{partition} : t.\textit{dim} \leq \text{splitVal}_1\}$
(8) $\text{QI}_2 \leftarrow \{t \in \textit{partition} : \text{splitVal}_1 < t.\textit{dim} \leq \text{splitVal}_2\}$
(9) $\cdots$
(10) $\text{QI}_{n_g} \leftarrow \{t \in \textit{partition} : t.\textit{dim} > \text{splitVal}_{n_g-1}\}$
(11) **return** $\text{QI}_1 \cup \text{QI}_2 \cup \cdots \cup \text{QI}_{n_g}$
(12) **end if**

Algorithm 1: Modified Mondrian algorithm.

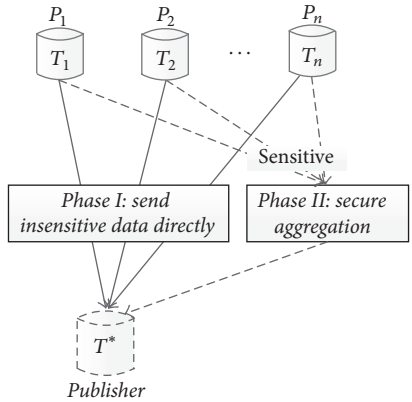


Figure 1: Our proposed two-phase scheme.

two-phase scheme. In our scheme, we have two key assumptions (Assumptions 3 and 4).

Assumption 3. The adversaries are semihonest (i.e., honest but curious), who will faithfully follow the protocol. However, these adversaries will also try to infer user privacy based on the protocol interactions.

Assumption 4. There are at most m colluding providers.

4.2. Two-Phase Scheme. Our scheme, based on Observation 1, consists of two phases. In the first phase, all providers transmit only data with no private attributes to the untrusted publisher, who will carry out an algorithm implementing m-k-anonymity on the received data (see Section 4.2.1). In the second phase, $m + 1$ randomly choose providers to collaboratively aggregate data with private attributes using a suitable cryptographic system (see Section 4.2.2). An illustration of the scheme is depicted in Figure 1.

Observation 5. The anonymization process for k-anonymity does not involve private attributes.

4.2.1. Phase I: k-Anonymization with Insensitive Attributes. In Phase I, the providers remove private attributes from the data prior to sending it to a third-party publisher that they may not truly trust. The publisher will then run a modified Mondrian algorithm on the received data to achieve m-k-anonymity. Mondrian [1], one of the most efficient algorithms implementing k-anonymity, models the dataset using a multidimensional space with each attribute contributing a dimension. k-Anonymization in this multidimensional space is to recursively partition each subspace into two smaller subspaces which do not overlap with each other until the stop condition is satisfied. Each subspace represents a QI group.

To achieve m-k-anonymity, we modify the termination condition of Mondrian based on Conclusion 6.

Conclusion 6. Let P_j^i $(1 \leq j \leq m,\ 1 \leq i \leq n_g)$ be m providers who have the most records in group QI_i. If $\forall \text{QI}_i$ $(1 \leq i \leq n_g)$, $|\text{QI}_i| - \sum_{j=1}^{m} |T_{P_j^i}(\text{QI}_i)| \geq k$, then the result satisfies m-k-anonymity.

Conclusion 6 is straightforward to draw: as the set P_j^i denotes the m providers contributing the most records in QI_i and $|\text{QI}_i| - \sum_{j=1}^{m} |T_{P_j^i}(\text{QI}_i)| \geq k$, it is easy to infer that, after removing any m providers data from QI_i, the number of remained records must be no less than k. This means each group still satisfies k-anonymity.

Based on this conclusion, we modify Mondrian so that the algorithm terminates when, in every subspace, a further partition will result in the number of records, owned by m providers who have most records, being greater than the total number minus k. The output of this algorithm is a number of QIs subject to m-k-anonymity. More details on the algorithm are demonstrated in Algorithm 1. Each iteration in Algorithm 1 is divided into $n_g - 1$ frequency sets according to dimension and values. Then for each frequency set, the split value will be calculated. According to $n_g - 1$ split values, the attributes can be divided into n_g subspaces. Each subspace is a group of quasi identifier. The algorithm returns QIs that consist of $\text{QI}_1, \text{QI}_2, \ldots, \text{QI}_{n_g}$.

4.2.2. Phase II: Private Data Aggregation. In order to ensure the security of the private data, Phase II of our scheme uses a secure public-key cryptographic algorithm of the scheme

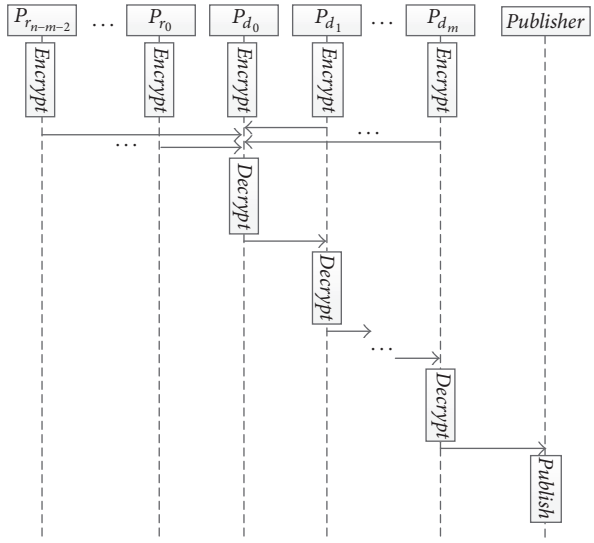


Figure 2: Phase II: private data aggregation.

implementer's choice, such as RSA. Every provider generates its public and private key pair, makes public the public key, and protects the private key.

The publisher will first send QIs obtained in Phase I to every provider. At the same time, the publisher randomly selects $m + 1$ decryption providers, $P_d = \{P_{d_0}, P_{d_1}, \ldots, P_{d_m}\}$, from P and sends their addresses to each provider. We denote the remaining providers as P_{r_i} $(0 \leq i \leq n - m - 2)$. On receiving QIs, every provider iteratively assigns private attributes of every record to the group whose QI contains its value of every nonprivate attribute. The providers encrypt the private data and their group information, using $m + 1$ decryption providers' encryption keys one at a time, in the reverse order of the addresses they received. Note that the encryption scheme used needs to be probabilistic (i.e., basic requirement of a secure encryption scheme). In other words, the encryption scheme introduces randomness in the encryption so that the encryption of the same message will produce a different cipher text each time.

As illustrated in Figure 2, the chain consists of m providers. We choose P_{d_0} as the first decryption provider in the chain. All providers $(P_{r_{n-m-2}}, \ldots, P_{r_0}, P_{d_0}, \ldots, P_{d_m})$ will decrypt their data and send the encrypted data to the first decryption provider P_{d_0}. Upon receiving all encrypted data, P_{d_0} will decrypt and uniformly repermute the decrypted data, prior to sending the decrypted data to the next decryption provider $(P_{d_0}, \ldots, P_{d_m})$. All decryption providers in the chain repeat the same process, sequentially. When the last decryption provider obtains the partially decrypted data, it performs the (last) decryption and submits to the publisher. This repermutation process breaks the linkage between the data and their providers.

In our proposal, providers have to perform decryption operations, which would inevitably bring additional communication and computation overheads. However, considering the fact that the data publishing is usually performed offline, we think higher overheads are affordable so long as they are still within reasonable bounds (e.g., a couple of hours or days). The detailed complexity analysis of our proposal is shown in Section 4.3. Our experiment results on real-world datasets in Section 6 also show that the time overheads are within the acceptable range.

4.3. Security Analysis. We now present the security proof for our proposed scheme based on Theorem 7.

Theorem 7. *The two-phase scheme can correctly implement m-k-anonymity in the semihonest model (described in Section 4.1).*

Proof.

Phase I. It is trivial to observe that data privacy will not be compromised, since the providers transmit only data with no private attributes.

Phase II. Since at most m providers are malicious and there are $m + 1$ or more decryption providers, there must exist at least one honest decryption provider in the decryption chain (which is the worse-case scenario).

Case 1. Adversaries are in front of honest decryption providers in the decryption chain. The adversaries are not able to fully decrypt the cipher texts as they do not have all decryption keys, although knowing the cipher text produced by another provider can be useful in inferring a record's private attribute.

Case 2. Adversaries are after honest decryption providers in the decryption chain. In this case, the adversaries could collaborate to obtain the plain text of the private data, but they are unable to map the data to their owners due to mix operations of the (one or more) honest provider(s).

Case 3. Adversaries are both before and after honest decryption providers in the decryption chain (e.g., $Adversary_1$, $Adversary_2, \ldots, Adversary_x$, $HonestProvider_1$, $Adversary_{x+1}$, $HonestProvider_2$, $Adversary_{x+2}, \ldots$). In this case, before data pass the honest providers, adversaries will not be able to collaborate to fully decrypt the data as they do not have all decryption keys. After the data has passed the honest provider(s), the adversaries could collaborate to decrypt the records. However, since the data has been repermuted by the honest provider(s), the adversaries will not be able to link the records to their owners. Therefore, the proposed scheme is still secure under this case. □

Complexity Analysis. Phase I: the complexity of the modified Mondrian algorithm is $O(N \log N)$, similar to the performance in the original Mondrian algorithm, where N is the total number of records. As the providers submit their insensitive data directly to the publisher, the communication complexity of Phase I is $O(N)$.

Phase II: the major computations are the encryption and decryption of private attributes, but it is easy to find that every record involves $m+1$ encryptions and $m+1$ decryptions. Thus,

the computation complexity of Phase II is $O(mN)$. Since an encrypted record is transmitted for up to $m + 2$ times, the communication complexity of Phase II is also $O(mN)$.

4.4. Discussion. In this part, we will discuss the reliability of our scheme and compare it with TTP-scheme proposed by Goryczka et al. [13].

Goryczka et al. propose an anonymization algorithm based on the Binary Space Partitioning. This algorithm can be implemented in a distributed environment by a trusted third party (TTP), which is considered as a secure anonymization protocol. It consists of two subprotocols. This first one is the provider-aware anonymization protocol. The time complexity of this protocol is determined by the number of records N and the number of attributes $|q|$. The analysis on provider-aware anonymization protocol shows that its time complexity equals $O(N(q + 1)(q^2 + np_s))$, where n is the number of providers and p_s is the maximal number of fake values. The second one is the secure fitness score protocol and its time complexity is $O(n^2 + np_s)$.

However, it is not easy to find a trusted third party in the real world. So, our scheme is TTP-independent which can achieve m-k-anonymity in a more practical manner. Since we assume that there are at most m-adversaries and the total number of providers is more than $m + 1$, there must exist at least one honest provider. We have proved the security of this proposal in Section 4.3. Our scheme is divided into two phases. The time complexity of Phase I is $O(N \log N)$. Phase II's time complexity equals $O(mN)$. The time complexity of our scheme increases linear with n, where n is the number of providers, which is reasonable for an offline scheme. According to these facts, we think that our scheme is more secure in the real-world since it does not rely on any trusted third-party. What is more, the increased computation overheads due to encryptions and decryptions are within acceptable range, which have been demonstrated by our experiments on real datasets.

5. An Extended Scheme Secure against a Stronger Adversary Model

5.1. Fully Malicious Model. The semihonest model, while widely accepted in the literature, may not be practical in the real world. More specifically, in the semihonest model, we are trusting providers not to misbehave. For example, Choo [22, 23] remarked that "there are legitimate concerns about cloud service providers being compelled to hand over user data that reside in the cloud to government agencies without the user's knowledge or consent due to territorial jurisdiction by a foreign government." Similar concerns were raised in [24], which then presented an extended proxy-assisted approach to address the concern of the need to trust the cloud server not to disclose user's proxy keys which is inherent in proxy/mediator assisted user revocation approaches.

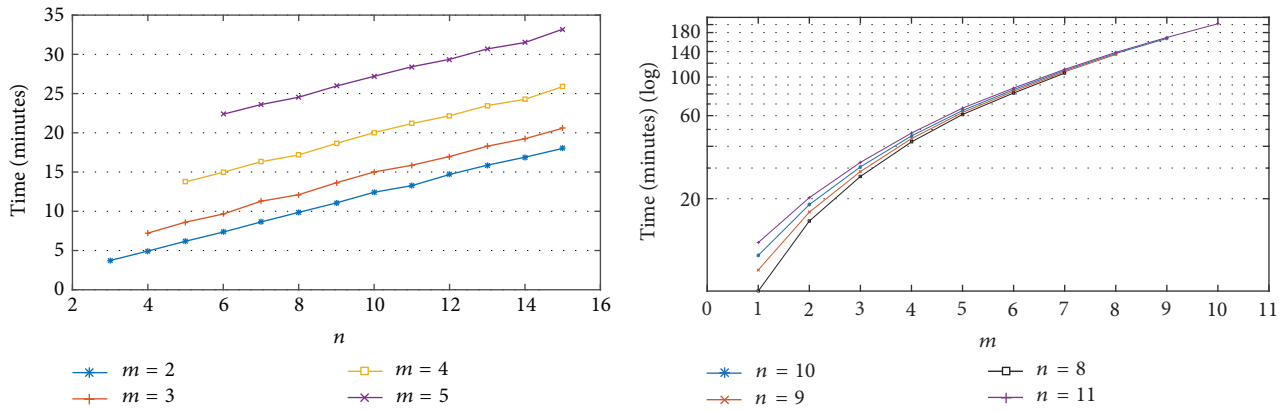
Therefore, in this section, we will present a fully malicious model, which does not require an adversary to follow the protocol. In fact, the adversary's aim is to successfully compromise user privacy. In the remainder of this paper, we will focus on tampering attacks that can be undertaken by (malicious) decryption providers. More specifically, if a decryption provider is malicious, the provider can replace encrypted data belonging to one or more honest providers with fictitious or fabricated data. Consequently, in the published result, the adversary(ies) can remove its/their original data and the fictitious or fabricated data inserted by the decryption provider from a number of QI groups. Hence, these groups will contain fewer than k records.

5.2. Sampling-Based Extension. The sampling-based extension of our scheme is described as follows.

From Section 4.2.2, we know that every provider encrypts the private data and their group information using $m + 1$ decryption providers' public keys one at a time. After encrypting the private data with the respective public keys, every provider generates $m + 1$ special strings, SS_j^i $(0 \le j \le m)$ of P_i, and sends SS_j^i to the decryption provider P_{d_j}. These strings are special because they do not have any group information. Then, each provider P_i $(1 \le i \le n)$ adds s pieces of every SS_j^i $(0 \le j \le m)$ encrypted with $\{P_{d_j}, P_{d_{j-1}}, \ldots, P_{d_0}\}$'s public keys, respectively, and mixes them with the encrypted private data.

Once P_{d_j} has finished the necessary decryption, $n * s$ special strings should have been fully decrypted. P_{d_j} can easily distinguish these fully decrypted strings from previously obtained information. Therefore, these special strings can be removed. Only when the number of every SS_j^i $(1 \le i \le n)$ equals s could the decryption provider sends the remaining decrypted data to the next decryption provider; otherwise, the decryption provider must discard all data and inform the publisher and the other providers. We suppose adversaries remove each record with probability r; then every provider's detection rate is $1 - (1 - r)^s$. For a specific dataset, different s correspond to different detection rate. So we can set a threshold; if the detection rate is greater than the threshold, then we choose the value of s as the number of special strings to be added by each provider. For example, in Section 6 we can choose $s = 300$ when the detection rate is greater than 95%.

Worst-Case Scenario Analysis. There is only one honest decryption provider in the decryption chain. Adversaries appearing after the honest decryption provider in the decryption chain cannot tell the data's owners even if they fully decrypt the cipher texts because the honest provider has faithfully repermuted the data which will break the relation between the cipher texts and their providers. Adversaries appearing before the honest provider in the decryption chain have access to the information which identifies the provider. However, these adversaries will not be able to fully decrypt all data except the special strings for validation. Therefore, regardless of the method used, the adversaries could probably remove some special strings which can be detected by the honest decryption provider. This simple method fulfills our design goal (i.e., an efficient and practical CPPDP scheme providing m-k-anonymity without involving a TTP), without compromising on the quality of the result. Although the



(a) Execution time versus n (n is the number of providers)

(b) Execution time versus m (m is the number of adversaries)

FIGURE 3: Execution time.

additional data do not have group information, it can be easily distinguished and removed from the result.

Suppose that adversaries appearing before the honest provider in the decryption chain remove each private record of P_i independently with probability r; then the detection rate is $1-(1-r)^s$ by P_i. By inserting additional validation strings, we will achieve a higher detection rate. However, this will result in a longer processing time.

6. Experiment Setup and Findings

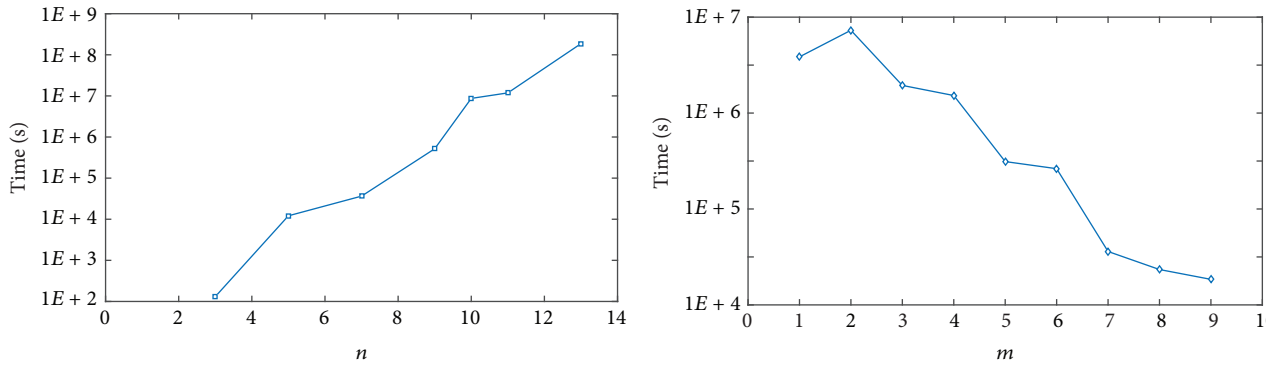
We now describe our experimental setup and findings.

6.1. Setup. We performed the experiments on several machines, each with 2.4 GHz Xeon E5 CPU and 2 G RAM. The operating systems are Ubuntu 12.04 and the implementation was built and run in Java 2 Platform Standard Edition 7.0. We used the Adult dataset (http://archive.ics.uci.edu/ml/datasets/Adult), which is a commonly used benchmark in the literature [25–27]. We combined the training and test sets in the Adult dataset and removed records with missing attributes. Thus, we ended up with a dataset of 45222 records with 14 attributes. We assigned {*age, workclass, fnlwgt, education, education-num, marital-status, relationship, race, sex, capital-gain, capital-loss, hours-per-week, native-country, salary*} as the QI and *occupation* as a private attribute. The total datasets were uniformly distributed among n data providers so that each of them was assigned a subset of similar size. We implemented RSA due to its popularity in commercial applications; each provider generates 1024-bit keys. We considered time as the key efficiency metric in our evaluations.

6.2. Findings. We measured the time between the publisher executing the scheme and receiving the resulting dataset. Findings are illustrated in Figure 3. In Figure 3(a), $n * 3000$ records were chosen from the original dataset and uniformly distributed among n providers; thus, each provider had 3000 records on average. In Figure 3(b), all 45222 records were uniformly distributed among $n = 10$ providers.

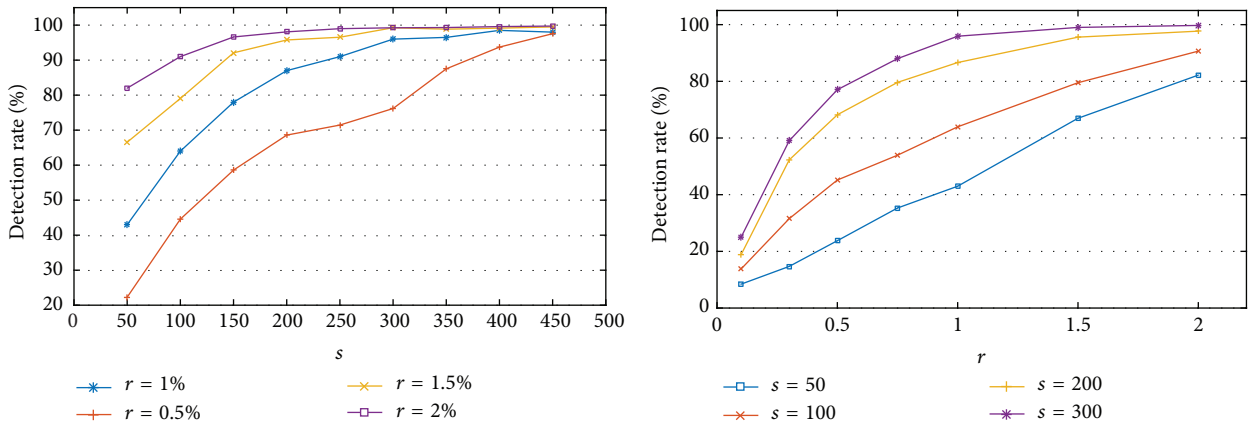
According to the definition of m-k-anonymity, there must exist at least one honest provider among n providers. Thus, n is always greater or equal to $m + 1$. It can be seen from Figure 3(a) that the execution time is approximately linear to n, which represents the size of the complete dataset. The result is also consistent with our guess. Even in the event that there are 15 providers, the time cost is below 20 minutes when $m = 2$. When $n = 15$ and $m = 5$, the time cost is below 35 minutes. This is sufficiently efficient since data publishing is usually performed offline. In addition, from Figure 3(b), we can see that the execution time increases a little faster with m. This is easy to understand since more colluding providers indicate more encryptions and decryptions for each record and these cryptographic operations are very time consuming. Fortunately, the increased rate is still acceptable since the total number of encryptions and decryptions increases linearly with m. If we assume that fewer than half of the providers may collude, the time cost is around one hour, which is reasonable for an offline algorithm.

According to the literature [13], the runtime of TTP-scheme is very high. The computation time of TTP-scheme increases almost exponentially with n which represents the number of providers. So doing experiment on secure m-privacy anonymization which is a subprotocol in TTP-scheme to achieve the computation time is unrealistic. Hence, we take the same approach as the authors mentioned in [13] to estimate the magnitude of computation time on the same dataset. The result is shown in Figure 4. Figure 4(a) shows the estimated time with varying values of n. In Figure 4(a), we can see that the computation time increases exponentially with n. And Figure 4(b) describes the estimated time varying different m. Due to the *provider-aware* anonymization protocol in TTP-scheme, when the number of adversaries increases, the computation time decreases exponentially. *Provider-aware* means that providers will be aware if there exist one or more adversaries among all providers. Thus, as the number of adversaries increases, providers will discover adversaries earlier. So the increasing m will cause the anonymization process to end earlier. In terms of TTP-scheme implementation, the secure protocols can be chosen as different algorithms, such



(a) Estimated execution time versus n (n is the number of providers) (b) Estimated execution time versus m (m is the number of adversaries)

FIGURE 4: Estimated execution time of TTP-scheme.



(a) Detection rate versus s (s is the number of inserted special strings) (b) Detection rate versus r (r is the discarding rate)

FIGURE 5: Detection rate.

as *top − down*, *binary*. The choice of the algorithm will not adversely affect the result of computation time.

We also remark that the security of the TTP-based scheme can be guaranteed, in the sense that the m-privacy anonymization protocol in TTP-scheme is secure as long as the subprotocol in the scheme is secure. We refer interested reader to [13] for the detailed security proof. Both TTP-scheme and our TTP-independent scheme can achieve m-privacy and k-anonymity. However, our algorithm is more efficient and practical. Due to the use of a strong encryption/decryption algorithm, our scheme is more secure in practice. The execution time of our scheme on the real dataset demonstrates that it is practical for deployment, especially for an offline algorithm.

The following experiments illustrate the number of special strings we need to insert in order to obtain an ideal detection rate. We ran our extended scheme 1000 times under different settings (e.g., different adversary discarding rates and numbers of special strings). Suppose that the adversaries appearing before the honest decryption provider in the decryption chain collaboratively remove each record independently with probability r. Figure 5(a) shows the detection rate under different s (the number of inserted validation strings) when $r = 0.01$ and Figure 5(b) shows the detection rates to different discarding rates when $s = 300$.

It can be seen in Figures 5(a) and 5(b) that, when adversaries remove each record with probability 1% independently, inserting 300 special strings yields a detection rate larger than 95%. For our experimental settings, 300 is about 1/15 of the size of each provider's dataset, and the extension will increase the execution time by about 1/15 as the total time is linear to the number of records. In other words, the number of validation strings to be inserted is determined by the desired detection rate and discarding rate of the context.

7. Conclusion

In this paper, we studied the m-adversary problem, where m (geographically dispersed) providers could collude. Existing solutions either depend on a trusted third party (TTP) or have impractical time overheads. In our proposed two-phase scheme, however, we demonstrated how our scheme can be used to implement m-k-anonymity without the need for a TTP. We also proved the security of the scheme in a semihonest adversary model. We then explained how our scheme can be extended so that it is also secure in a stronger

adversary model. Lastly, our experiments demonstrated the practicality of our scheme to be deployed in a real-world context.

Future research include extending the scheme to provide m-privacy with respect to other privacy constraints and generalize the scheme to implement m-privacy with respect to k-anonymity on distributed incremental datasets or collaborative data republishing.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by National 973 Programs (Grant no. 2013CB329102), National Natural Science Foundation of China (Grants nos. 61100194, 61300235, 61300117, and 61272188), Open Foundation of State of State Key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) (Grant no. SKLNST-2013-1-14), National Science Foundation of Zhejiang Province (Grant no. LY12F02005), and Open Foundation of State Key Laboratory for Novel Software Technology of Nanjing University (Grant no. KFKT2014B15).

References

[1] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Mondrian multidimensional K-anonymity," in *Proceedings of the 22nd International Conference on Data Engineering (ICDE '06)*, pp. 25–34, IEEE, Atlanta, Ga, USA, April 2006.

[2] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: efficient full-domain K-anonymity," in *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data (SIGMOD '05)*, pp. 49–60, Baltimore, Md, USA, 2005.

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

[4] Y. Lindell and B. Pinkas, "Pinkas secure multiparty computation for privacy-preserving data mining," *Journal of Privacy and Confidentiality*, vol. 1, no. 1, pp. 59–98, 2009.

[5] M. E. Nergiz, E. Çiçek, T. Pedersen, and Y. Saygin, "A look-ahead approach to secure multiparty protocols," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 7, pp. 1170–1185, 2012.

[6] W. Jiang and C. Clifton, "Privacy-preserving distributed k-anonymity," in *Proceedings of the 19th Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, pp. 166–177, Storrs, Conn, USA, August 2005.

[7] S. Zhong, Z. Yang, and R. N. Wright, "Privacy-enhancing k-anonymization of customer data," in *Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS '05)*, pp. 139–147, June 2005.

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

[9] P. Jurczyk and L. Xiong, "Distributed anonymization: achieving privacy for both data subjects and data providers," in *Proceedings of the 23rd Annual IFIP WG 11.3 Working Conference on Data and Applications Security XXIII*, vol. 5645 of *Lecture Notes in Computer Science*, pp. 191–207, Springer, 2009.

[10] N. Mohammed, B. C. M. Fung, K. Wang, and P. C. K. Hung, "Privacy-preserving data mashup," in *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology (EDBT '09)*, pp. 228–239, ACM, Saint Petersburg, Russia, March 2009.

[11] N. Mohammed, B. C. M. Fung, P. C. K. Hung, and C.-K. Lee, "Centralized and distributed anonymization for high-dimensional healthcare data," *ACM Transactions on Knowledge Discovery from Data*, vol. 4, no. 4, article no. 18, 2010.

[12] A. C. Yao, "Protocols for secure computations," in *Proceedings of the 23rd IEEE Symposium on the Foundation of Computer Science (FOCS '82)*, pp. 160–164, Chicago, Ill, USA, November 1982.

[13] S. Goryczka, L. Xiong, and B. Fung, "m-privacy for collaborative data publishing," in *Proceedings of the 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pp. 1–10, Orlando, Fla, USA, October 2011.

[14] S. I. Ahamed, M. M. Haque, and C. S. Hasan, "A novel location privacy framework without trusted third party based on location anonymity prediction," *ACM SIGAPP Applied Computing Review*, vol. 12, no. 1, pp. 24–34, 2012.

[15] L. L. Win, T. Thomas, and S. Emmanuel, "Privacy enabled digital rights management without trusted third party assumption," *IEEE Transactions on Multimedia*, vol. 14, no. 3, pp. 546–554, 2012.

[16] Z. Fu, X. Wu, C. Guan, X. Sun, and K. Ren, "Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2706–2716, 2016.

[17] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 9, pp. 2546–2559, 2016.

[18] Z. Fu, X. Sun, S. Ji, and G. Xie, "Towards efficient content-aware search over encrypted outsourced data in cloud," in *Proceedings of the 35th Annual IEEE International Conference on Computer Communications (INFOCOM '16)*, pp. 1–9, IEEE, San Francisco, Calif, USA, April 2016.

[19] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 340–352, 2016.

[20] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-diversity: privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, article 3, 2007.

[21] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: privacy beyond k-anonymity and ℓ-diversity," in *Proceedings of the 23rd International Conference on Data Engineering (ICDE '07)*, pp. 106–115, April 2007.

[22] K.-K. R. Choo, "Cloud computing: challenges and future directions," *Trends & Issues in Crime and Criminal*, vol. 400, pp. 1–6, 2010.

[23] K.-K. R. Choo, "Legal issues in the cloud," *IEEE Cloud Computing*, vol. 1, no. 1, pp. 94–96, 2014.

[24] Y. Yang, J. Liu, A. Liang, K.-K. R. Choo, and J. Zhou, "Extended proxy-assisted approach: achieving revocable fine-grained cloud data encryption," in *Proceedings of the 20th European Symposium on Research in Computer Security*, pp. 146–166, Vienna, Austria, September 2015.

[25] G. S. Babu and S. Suresh, "Sequential projection-based metacognitive learning in a radial basis function network for classification problems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 2, pp. 194–206, 2013.

[26] P. G. Ipeirotis, F. Provost, V. S. Sheng, and J. Wang, "Repeated labeling using multiple noisy labelers," *Data Mining and Knowledge Discovery*, vol. 28, no. 2, pp. 402–441, 2014.

[27] K. Subramanian, R. Savitha, and S. Suresh, "A complex-valued neuro-fuzzy inference system and its learning mechanism," *Neurocomputing*, vol. 123, pp. 110–120, 2014.