A Novel Privacy Preserving Association Rule Mining using Hadoop

Kangsoo Jung, Sehwa Park, Sungyong Cho, Seog Park
Department of Computer Engineering, Sogang University, Seoul, Korea
azure84@sogang.ac.kr sehwapark@sogang.ac.kr Jsy9kr2004@hanmail.net spark@sogang.ac.kr

Abstract— **Hadoop is a popular open source distributed system that can processes large scale data. Meanwhile, data mining is one of the techniques used to find pattern and gain knowledge from data sets, as well as improve massive data processing utility when combined with the Hadoop framework. However, data mining constitutes a possible threat to privacy. Although numerous studies have been conducted to address this problem, such studies were insufficient and had several drawbacks such as privacy-data utility trade-off. In this paper, we focus on privacy preserving data mining algorithm technique, particularly the association rule mining algorithm, which is a representative data mining algorithm. We propose a novel privacy preserving association rule mining algorithm in Hadoop that prevents privacy violation without the loss of data utility. Through the experimental results, the proposed technique is validated to prevent the exposure of sensitive data without degradation of data utilization.**

Keywords-Privacy preserving data mining; Association rule mining; Hadoop.

I. INTRODUCTION

Hadoop [1] is a scalable and stable distributed data processing open-source project that has become a "de facto" standard among big data processing techniques. The development of big data processing techniques such as Hadoop has contributed to the proliferation of massive data analysis, which has not yet been explored in literature. Large size data processing utility is enhanced through its combination with data mining algorithm that is employed in rule mining and pattern recognition. Thus, numerous studies have been conducted to apply existing mining techniques to the MapReduce programming model. Meanwhile, data mining using big data may result in serious privacy violation through the inference of sensitive information. Therefore, research about privacy preserving data mining algorithm of massive datasets is necessary.

Data mining that utilizes big data requires a considerable amount of resources for proper processing. However, constructing this type of environment is burdensome for an individual or a single company. For this reason, cloud platforms, such as Amazon EC2 [2], provides service related to big data mining processes at a lower cost. However, in such platforms, personal data can flow to untrusted cloud service provider during the data mining process. The solutions proposed in literature, which include encryption [3] and privacy preserving data mining [7] algorithm; however, these methods have several disadvantages. Encryption has a too rigorous restriction because of its low computational performance, while privacy preserving data mining algorithm has one weakness on the tradeoff between privacy protection degree and data utility.

In this paper, we propose a novel Privacy Preserving Data Mining (PPDM) algorithm to overcome the limitation of existing methods with the following considerations: (1) under the environment of untrusted external cloud platform and (2) without the loss of data utility while performing privacy preserving data mining. Our method focuses on the association rule mining algorithm which is one of the data mining techniques that have received considerable attention. The proposed technique does not use encryption, but prevents the exposure of data collected under the agreement of the data provider.

The remainder of this paper is organized as follows: Section 2 reviews the association rule mining algorithm, privacy preserving data mining algorithm, and privacy preserving data processing method based on the Hadoop framework. Section 3 describes our preliminary assumption and motivation. Section 4 introduces the proposed privacy preserving association rule mining algorithm. Section 5 presents the results of the performance evaluation. Finally, Section 6 summarizes and concludes the paper.

II. RELATED WORKS

A. Association rule mining

Association rule mining [4] is a data mining algorithm that discovers interesting relations among merchandises based on purchase history. Relations are generally represented as a rule. Meanwhile, support and confidence of an itemset are used as a measure to select interesting rules from the set of all possible rules.

$$\text{Support}(X, Y) = \frac{\#\{customers\ who\ bought\ X\ and\ Y\}}{\#\{customers\}} \quad (1)$$

$$\text{Confidence}(X \rightarrow Y) = \frac{\#\{customers\ who\ bought\ X\ and\ Y\}}{\#\{customers\ who\ bought\ X\}} \quad (2)$$

The well-known apriori algorithm [5] identifies association rules by attempting to select frequent item set that has minimum support value. Frequent item sets are extended from one to the maximum length of transaction until no further extensions are found. Finding frequent itemsets, apriori algorithm determines relations by calculating the confidence value of frequent itemsets.

 ISBN: 978-1-61208-358-2

B. *Privacy preserving data mining*

Privacy preserving data mining technique can be classified into randomization and distributed privacy preserving techniques. For randomization, data perturbation and noise addition are generally used [6]-[8], and k-anonymity [8] and differential privacy [9] have been investigated recently as well. Meanwhile, distributed privacy preserving technique is a method that employs secure multi-party computation and encryption to share mining result in a distributed environment without revealing sensitive information about particular parties.

Privacy preserving mining of association rule [12][15] is an essential part in data mining. A crucial step in privacy preserving association rule is to find the global support value of frequent item set without compromising the privacy of sensitive data. To achieve this, randomization technique [13] is used, through which the data provider sends randomized data to the data miner. Data miner calculates the support value of frequent itemsets using randomized data. The support value of frequent itemset is different from the original data's support value. Hence, randomization technique prevents the exposure of correct frequent items set.

However, Evfimievski et al. [15] points out that traditional privacy preserving association rule mining has problems when it employs uniform randomization technique. Thus, they proposed a novel randomization method to overcome the existing limitation of the technique. However, a randomization technique has several disadvantages that reduce the accuracy of association rule while preserving data privacy. This issue is the privacy-data utility tradeoff problem.

C. *Massive data processing in Hadoop*

Given the growing importance of massive data processing, researchers have explored the application of data mining in a Hadoop environment. Mahout [16] is an open source library that implement machine learning algorithm in the MapReduce programming model. Mahout provides various data mining algorithm such as clustering and classification that can run without additional MapReduce programming in Hadoop.

However, the development of big data mining techniques entails the increased possibility of privacy violation. Ko et al. [17] and Zhang et al. [18] proposed a privacy preserving data processing framework based on Hadoop, and related studies have been conducted in this area. However, privacy preserving data mining algorithm in Hadoop has yet to be explored.

III. PRELIMINARIES

A. *Problem definition*

1) Assumption 1. Service providers want to exploit the external cloud service platform based on Hadoop framework to achieve association rule mining for big data.

To achieve data mining in a large dataset, using a distributed processing framework is advantageous. However, implementing a large-scale distributed environment for an individual or a single company is a rigorous task. In this regard, employing external cloud services such as Amazon EC2 is an efficient solution. For example, The New York Times uses cloud services to convert their news data into digital information. At present, most of these cloud services use Hadoop framework based on the MapReduce programming model. Hence, revising the existing data mining algorithm to correspond to the Hadoop system is necessary.

2) Assumption 2. Data providers want to limit privacy violations when the service provider processes data using external cloud services.

Service providers send data to external cloud services for data processing. However, such external cloud services are not trusted third party. Thus, if the service provider provides raw data to external cloud service without any consideration of privacy, privacy leakage of sensitive information occurs during the data mining process. As indicated in Fig. 1, although data providers agree to provide data to service providers they do not agree with offering data to an external cloud server. As such, the privacy of the data provider should be sufficiently protected.

3) Assumption 3. Service providers want to maximize the accuracy of rule which is extracted through association rule mining.

Existing methods to protect the privacy of data providers add noise to data. However, such techniques limit the performance of data mining. With the goal of service providers to protect the privacy of data providers, and to maximize the data utility, a privacy-preserving data mining algorithm, which considers the privacy-data utility trade-off, is necessary.

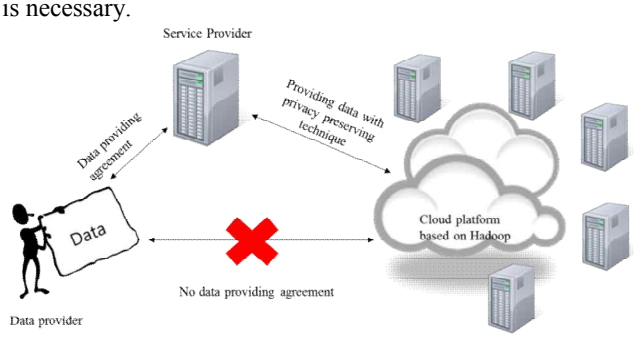

Figure 1. The proposed architecture of association rule mining in cloud platforms.

B. *Motivation*

As mentioned in Assumption 3, the randomization technique that adds noise to data results in a problem in privacy-data utility trade-off. Thus, high privacy protection degree reduces data utility, and the high data utility requirement increases the possibility of privacy violations.

In this paper, we propose a novel privacy-preserving association rule mining algorithm that prevents privacy violation in untrusted external cloud platforms without deteriorating data utility. The proposed technique adopts an identification key using prime number property to filter out

 ISBN: 978-1-61208-358-2

noise, and to process MapReduce programming for privacy-preserving association rule mining. The proposed method has several disadvantages, such as increasing calculation cost with the additional noise. Unlike privacy and data utility tradeoff, calculation cost and privacy relationship are not zero sum; Hadoop can efficiently handle the calculation cost.

IV. PRIVACY PRESERVING ASSOCIATION RULE MINING IN HADOOP

A. Basic scheme

As mentioned in Section 2, the key step of apriori algorithm is exploiting frequent itemsets. Selecting frequent itemsets results in increased calculation cost and privacy violation when performed through an external cloud service. As such, service providers add noise to the transaction data to prevent the exposure of correct frequent itemsets. However, this process results in reduced data utility. The proposed technique prevents data utility degradation by assigning an identification key that can distinguish original data from noise data. With the use of the identification key to filter out noise added by the service provider, the correct association rule can thus be extracted without utility degradation and privacy violation.

We assume that the set of transaction is $T=\{T_1,T_2,\ldots,T_n\}$, the set of item is $I=\{I_1,I_2,\ldots,I_{items}\}$, and the set of noise is $D=\{D_1,D_2,\ldots,D_{dummy}\}$. The identification key is defined as follows:

Definition 1. Identification key

We assume the set of integers $K=\{K_1,K_2,\ldots,K_n\}$, $n \geq items+dummy$. If K_i satisfies the following conditions, we define K_i as the identification key of item i.

$$key(I_i) = K_i \ (1 \leq i \leq items) \quad (3)$$

$$key(D_j) = K_{items+j} (1 \leq j) \quad (4)$$

$$K_i \neq K_j, (\forall i,j, i \neq j \ and \ K_i, K_j \in K) \quad (5)$$

The outline of the proposed method is as follows Fig. 2. The service provider adds a dummy item as noise to the original transaction data collected by the data provider. Subsequently a unique identification key is assigned to the dummy and the original items. The service provider maintains a mapping table for the item and identification key to filter out the dummy item after the selection of frequent itemset in an external cloud platform. Apriori algorithm is then performed in the external cloud platform using data sent by the service provider. The external cloud platform returns the frequent itemset results and count value to the service provider. The service provider filters the frequent itemset that is affected by the dummy item using an identification key, and extracts the correct association rule using frequent itemset without the dummy item. The extraction association rule is not a burden to the service provider, considering that the amount of calculation required for extracting the association rule is not much.

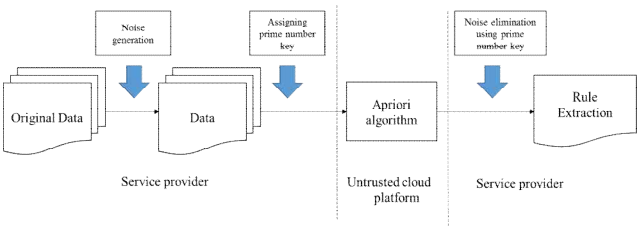

Figure 2. Overview of the proposed association rule mining

Definition 2. Prime number key

We assume the set of prime number $P=\{P_1,P_2,\ldots,P_n\}$. If P_i satisfies the following conditions, we define P_i as the prime number key of item i.

$$key(I_i) = P_i \ (1 \leq i \leq items) \quad (6)$$

$$key(D_j) = P_{items+j} (1 \leq j) \quad (7)$$

$$P_i \neq P_j, (\forall i,j, i \neq j \ and \ P_i, P_j \in P) \quad (8)$$

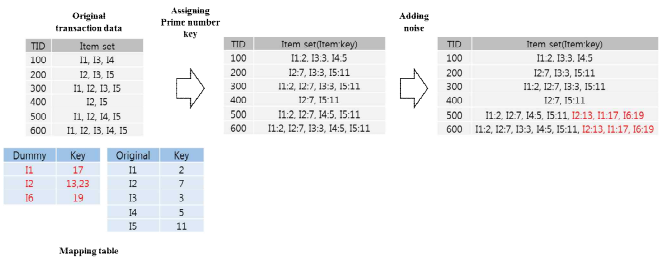

Figure 3. Prime number key assignment

B. Apriori algorithm using prime number indentification key

1) Prime number key

The method that generates dummy item with the prime number key is as shown in Fig. 3: The service provider generates dummy item as noise and assigns a prime number key to each item in the transaction data. In addition, the service provider maintains a mapping table to maintain relations with the prime number key. For example, TID500 transaction in Fig. 3 is {I1,I2,I4,I5}. The service provider assigns the prime number key to TID500, and TID500 becomes an item: prime number key pair transaction, such as {I1:2,I2:7,I4:5,I5:11}. Subsequently, a dummy item is randomly added using a noise generation algorithm, which will be explained in Section 4.3. Finally, TID500 becomes {I1:2,I2:7,I4:5,I5:11,I2:13,I1:17,I6:19}. Prime number key is used to distinguish the original item from the dummy item at the service provider, as well as a key value for MapReduce programming. The dummy item can have two or more prime number keys to enhance the degree of privacy protection.

 ISBN: 978-1-61208-358-2

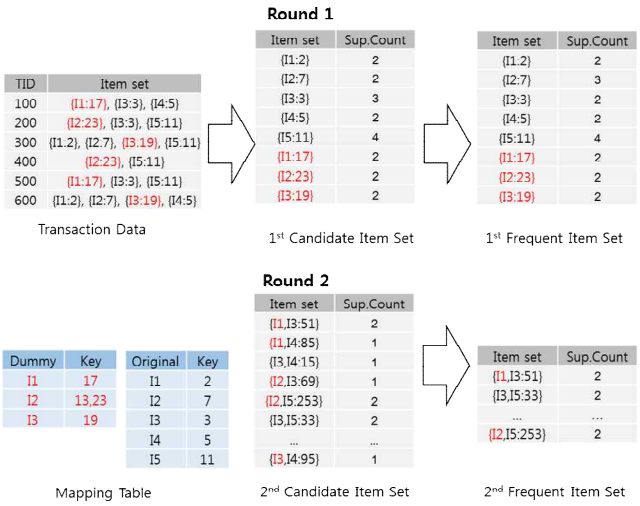

Figure 4. Apriori algorithm using prime number key

2) Apriori algorithm using a prime number key

Apriori algorithm is performed in the external cloud platform using Hadoop framework, which is similar to the traditional apriori algorithm, aside from the use of the item and prime number key pair as a key value. The frequent itemset selection process is shown in Fig. 4.

Each transaction data consists of a prime number key and an item pair. This pair and the number of pairs are regarded as the key and value, respectively, for MapReduce programming. During the processing, a subset of the transaction item's prime number key is multiplied by each item's prime number key. For example, a subset of transaction item second round frequent itemset {I4, I5} consists of I4 and I5, and the prime number key of {I4, I5} is 55, which is assigned by multiplying with I3 and I5's prime number keys, 5 and 11. Prime numbers have a property that makes multiplying with prime numbers unique. Hence, a prime number can be used as a key value for MapReduce programming. The prime number key for the subset of transaction is defined as

$$\text{key}(I') = \prod_{t=1}^{m} key(I_{r_t}) = \prod_{t=1}^{m} P_{r_t} \quad (9)$$

3) Elimination of dummy item

The frequent itemset and the number of frequent itemsets that is performed in an external cloud service platform are returned to the service provider with the prime number key. The proposed technique uses a prime number key to eliminate the frequent itemset that is influenced by the dummy item. If the frequent itemset's prime number key contains the dummy item's prime number key, the former can be divided by the latter. Hence, the service provider performs modular operation using the dummy item's prime number key to filter out frequent itemset that contains the dummy item. The filtering stage eliminates the dummy items from the remaining frequent itemsets. Using this frequent itemset, the service provider can thus extract an accurate association rule.

4) Advantage of the prime number key

The service provider can use other means to filter out noise, such as the hash value. However, this type of indexing method requires a sequential search to check the dummy item, the time complexity of which is O(n*m), (n = number of dummy items; m = number of frequent itemsets). However, the proposed technique only performs modular operation to filter out the dummy item, and time complexity is O(n), (n = number of dummy items). The disadvantage of prime number key is the size growth caused by multiplying prime numbers. We can address this weakness by using the grouping method.

C. Noise generation algorithm

In association rule mining, the possibility of privacy violation is given by the conditional probability f(X|S), which indicates the possibility of being able to infer original data X through noise data S. The proposed technique reflects the sensitivity of the item, which is set by the users to generate a dummy item as noise. That is, the data provider attaches a value from 1 to 10to the sensitivity of each item when providing transaction data. In this scale, 1 means the lowest sensitivity and 10 represents the highest sensitivity value. Item sensitivity is defined as follows:

Definition 3. Item sensitivity

We assume the set of each user's item sensitivity $S_i^m=\{S_1,S_2,\ldots,S_n\}$, and item sensitivity $\hat{S}_i$ is defined as follows:(m = # of data provider, n = # of item)

$$\hat{S}_i = \frac{(\sum_{k=1}^{m}\sum_{i=1}^{n} S_i^m)}{m} \quad (10)$$

The item sensitivity value is used to assign noise generation probability to each item. An item with a higher sensitivity has a larger possibility of generating noise, whereas an item that has lower sensitivity has a smaller possibility of generating noise. The proposed noise generation algorithm is as follows

1. The set of transaction $T_i=(I_1,I_2,..I_n)$, and integer j (j<n) is randomly selected. The probability of selecting j is proportional to the average sensitivity value of each transaction.
2. We perform Bernoulli trials to the item that is not contained in transaction T_i j times. Bernoulli trial probability ρ_i is proportional to each item sensitivity S_i.
3. The selected item is added to transaction T_i as noise.

D. Needles in haystack

Existing privacy-preserving association rule algorithms modify original transaction data through the addition of noise. However, we maintained the original transaction because our goal is to prevent data utility degradation while reducing the risk of privacy violation. Therefore, that an untrusted cloud service provider infers the real frequent itemset remains a possibility in the proposed method.

 ISBN: 978-1-61208-358-2

Despite the risk, we provide enough privacy protection because our privacy-preserving algorithm is based on "the needles in a haystack" concept. This concept is based on the idea that detecting a rare class of data, such as the needles, is hard to find in a haystack, which can be compared to a large size of data. For example, in the case of itemset {I1,I2,I3}, the possible association rules are I1→I2, I1→I3, I2→I3, I1→{I2,I3}, I2→{I1,I3}, I3→{I1,I2}, {I1,I2}→I3, {I1,I3}→I2, {I2,I3}→I1. If every possible association rule has to be extracted by adding noise, privacy protection is completely guaranteed as there is no need to extract association rule.

Existing techniques cannot add noise haphazardly because of the need to consider privacy-data utility trade-off. However, the proposed technique does not take such trade-off into consideration, for we can filter out noise using a prime number key. Nevertheless, the proposed technique incurs additional computation cost in adding noise that will make the "haystack" to hide the "needle." Therefore, we attained a trade-off between privacy and computational cost. The existing trade-off is not considered a serious problem; unlike the privacy-data utility trade-off, the computation cost and privacy protection does not have a zero-sum relationship. The problem of computational cost can be resolved by adding computing resources. Hence, the problem would be easier to be solved with the use of the Hadoop framework in a cloud environment.

V. EXPERIMENT

A. Experimental environment

In this paper, we implemented the Hadoop cluster to validate the performance of the proposed technique. The Hadoop cluster implemented in this study consists of one name node, one secondary name node, and three data nodes. The name node specifications are six-core 2.00 GHz Intex Xeon, 16 GB memory, and Ubuntu 12.10 64 bit. The specifications of the secondary name node and data node are 2-core 1.86GHz Intel CPU, 2 GB memory, and Ubuntu 12.04.3 LTS. We focused on the execution time evaluation because the proposed technique's most important drawback is the execution time degradation. We performed three experimental evaluations, namely, noise size, transaction size, and transaction length.

B. Noise size

The insertion of noise reduces system performance through the increase in amount of noise. We evaluated the performance of proposed technique by changing the noise size. We set the number of transaction as 110, where item type is 10, dummy item type is 9, and the support value is 50%. We increased noise size from 0% to 150% compared with the original transaction. The results show that the execution time increases along with the noise size. The execution time of the original transaction data where noise is not inserted is 126.008 seconds; with 50% noise, 143.578 seconds, 80%, 162.089 seconds; 120%, 172.089 seconds; and 150%, 176.040 seconds. Therefore, the execution time increases linearly in proportion to noise size. This result indicates that the proposed technique does not significantly reduce the performance.

C. Transaction size

In this experiment, we evaluated the system performance by increasing the transaction size. The item type, dummy item type, and support value were set as in the previous experiment. We increased the transaction size from 200 to 400, and set the noise size as 50% of the number of items compared with the original transaction. We can validate that in the 0% case (that is, without noise added) and 50% case (with noise added), the execution time increased because of the transaction data size. When the transaction data size is 200, the execution time of 0% is 73.586 seconds, whereas for 50%, the time is 90.587 seconds. If the transaction size is 300, 0% takes 92.6 seconds, and for 50%, the duration is 114.619 seconds. Finally, when the transaction data size is 600, the execution time of 0% is 113.598 seconds, and for 50%, 130.593 seconds.

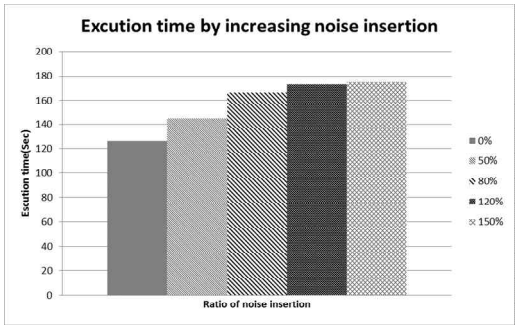

Figure 5. Execution time by increasing noise size

D. Transaction length

In this experiment, we evaluated the system performance in terms of the increasing transaction length. The transaction length is closely related to the iteration phase. If the transaction length increases, the iteration phase increases as well. The number of iteration phase directly affects the execution time. We added the dummy item to the transaction and increased the number of dummy items in the transaction to increase the transaction length. We set the item type, dummy item type, noise size, and support value as in the previous experiments. The transaction size is set as 1000.

The results show that the increase in transaction length results in the linear increase in execution time. Thus, the execution time of the proposed technique is affected by noise size, transaction size, and transaction length. However, this degradation is not a significant issue in the Hadoop framework. As such, privacy-computation cost trade-off can be achieved easily through the Hadoop framework, and the proposed technique can resolve the problem in privacy-data utility trade-off in existing randomization techniques.

 ISBN: 978-1-61208-358-2

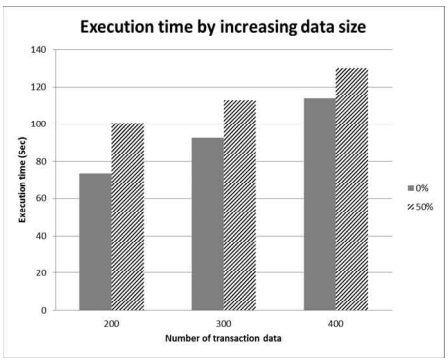

Figure 6. Execution time by data size

Figure 7. Execution time by transaction length

E. The number of Additional Rules

In this experiment, we evaluated the amount of privacy preservation of proposed technique by increasing the proportion of noise data. We increased the noise size from 15% to 120%. We set the number of transaction as 110, where item type is 10, dummy item type is 9, and the support value is 50%. We can validate that in the 0% case (that is, without noise added) and other cases (with noise added), the number of rules increased because of the ratio of noise insertion. Results of these experiments show that privacy would be protected enough.

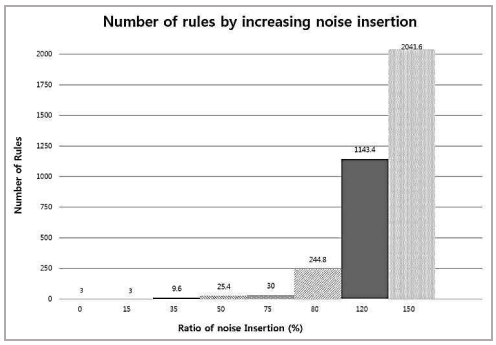

Figure 8. The Number of rules by increasing noise insertion.

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel privacy-preserving association rule mining method based on the concept of "needles in a haystack." In this method, we added a dummy item as noise to the original transaction data to generate the amount of fake frequent itemset to hide the real frequent itemset during association rule mining in untrusted cloud platforms. We used a prime number key to identify the dummy and the original items, while the prime number key was used as a key value to perform MapReduce programming as well. The results showed that the proposed technique does not significantly reduce the performance while sufficiently preventing privacy violation. In future works, we will attempt to formalize the degree of privacy protection of the proposed technique and optimize the noise generation algorithm to prevent performance deterioration.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2013R1A1A2013172).

REFERENCES

[1] Apache Hadoop. http://hadoop.apache.org/.

[2] Amazon Elastic Compute Cloud. http://aws.amazon.com/ec2/.

[3] D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," In Proc. of IEEE Symposium on Security and Privacy, Berkeley, CA, May. 2000, pp. 44-55

[4] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," In Proc. of Conf. Management of Data, ACM SIGMOD, Washington, DC, May. 1993, pp. 207-216.

[5] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," In Proc. of 20th int. conf. on Very Large Data Bases, Santiago, Chile, Sep. 1994, pp. 487-499.

[6] R. Agrawal and R. Srikant, "Privacy-Preserving Data Mining," In Proc. of Conf. on Management of Data, ACM SIGMOD, Dallas, TX, Sep. 2000, pp. 439-450.

[7] C. C. Aggarwal and P. S. Yu, "Privacy-Preserving Data Mining: A Survey," Handbook of Database Security : Application and Trends, Gertz, M. and Jajodia, S. (Eds.), 2008, pp. 431-460, Springer.

[8] K. Chen and L. Liu, "Privacy Preserving Data Classification with Rotation Perturbation," In Proc. of the 5th IEEE Int'l Conf. on Data Minig, Atlanta GA, 2005, pp. 589-592.

[9] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-Preserving Data Publishing: A Survey of Recent Developments," ACM Computing Surveys, 2010, vol. 42, no. 4, pp. 14-53.

[10] A. Friedman and A. Schuster, "Data Mining with Differential Privacy," In Proc. of the 16th ACM Int'l Conf. on Knowledge Discovery and Data Mining, Washington, DC, Jul. 2010, pp. 493-502.

[11] A. C. Yao, "Protocols for Secure Computations," In Proc. of the 23th IEEE Symp. on Foundations of Computer Science, Chicago, Illinois, Nov. 1982, pp. 160-164.

[12] J. Vaidya and C. Clifton, "Privacy Preserving Association Rule Mining in Vertically Partitioned Data," In Proc. of the 8th ACM Int'l Conf. on Knowledge Discovery and Data Mining, Alberta, Canada, Jul. 2002, pp. 639-644.

[13] Charu C. Aggarwal and Philip S. Yu, "A Survey of Randomization Methods for Privacy-Preserving Data Mining," Privacy-Preserving Data Mining: Models and Algorithms, Charu C. Aggarwal and Philip S. Yu, 2008, pp. 137-156, Springer.

 ISBN: 978-1-61208-358-2

[14] W. Du and M. J. Atallah, "Privacy-Preserving Cooperative Statistical Analysis," In Proc. of the 17th Conf. on Annual Computer Security Applications, New Orleans, Louisiana, Dec. 2001, pp. 102-110.

[15] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, "Privacy Preserving Mining of Association Rules," Information Systems, 2004, VOl. 29, pp. 343-364.

[16] Apache (2013) ApacheMahout machine learning library. http://mahout.apache.org/.

[17] Ko SY, Jeon K, and Morales R, "The hybrex model for confidentiality and privacy in cloud computing," In Proceedings of the 3rd USENIX conference on hot topics in cloud computing (HotCloud'11), 2011, pp. 1-5.

[18] X Zhang, C Liu, S Nepal, C Yang, and J Chen, "Privacy Preservation over Big Data in Cloud Systems," Security, Privacy and Trust in Cloud Systems, 2014, pp. 239-257.

 ISBN: 978-1-61208-358-2