IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 4, June 2014.

www.ijiset.com

ISSN 2348 - 7968

Classification of Information as datasets and Encrypting them for Privacy Protection in Cloud

Vinutha N
M.Tech Department of CS&E
VTU RC, Mysore

Abstract

Cloud provides data, resources, and storage capacity and computation power as services to the user which enables the user to perform some applications or functions without infrastructure investment. While performing such applications large volume information will be generated and that information is stored in cloud server to save the recomputing cost of them. And that information is classified into datasets. But third parties may recover the privacy sensitive information. Therefore, in existing system all the datasets are encrypted, but it is very time consuming and costly. Therefore in proposed system only few datasets are selected and encrypted using the novel approach. As a result cost of privacy protection of datasets can be reduced by satisfying the privacy requirement of data holders.

Keywords :*Privacy protection, Anonymization, Datasets.*

1 Introduction

The US National Institute of Standards and Technology (NIST) define cloud computing as "a model for user convenience, on demand network access contribute the computing resources (e.g. networks, storage, applications, servers, and services) that can be rapidly implemented with minimal management effort or service provider interference" Cloud computing can also be defined as it is a new service, which are the collection of technologies and a means of supporting the use of large scale Internet services for the remote applications with good quality of service (QoS) levels. Cloud computing is has many technologies such as Saa i.e. "Software as a Service", Paas i.e. "Platform as a Service", IaaS i.e. Infrastructure as a Service". Cloud Computing is a paradigm that focuses on sharing data and computations over a scalable network of nodes. Examples of such nodes include end user computers, data centers, and Cloud Services.

Cloud service delivery is divided among three archetypal models and various derivative combinations. The three fundamental classifications are often referred to as the -SPI Model where SPI' refers to Software, Platform or Infrastructure (as a Service), respectively defined.

1.1 Cloud Service Models.

• Cloud Software as a Service (SaaS): The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser (e.g., web-based email)

• Cloud Platform as a Service (PaaS): The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider (e.g., configurations)

• Cloud Infrastructure as a Service (IaaS): The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications.(e.g., host fire walls).

Cloud computing is technically regarded as an ingenious combination of a series of technologies, establishing a novel business model by offering IT services and using economies of scale. Participants in the business chain of cloud computing can benefit from this novel model. Cloud customers can save huge capital investment of IT infrastructure, and concentrate on their own core business [1]. Therefore, many companies or organizations have been migrating or building their business into cloud. However, numerous potential customers are still hesitant to take advantage of cloud due to security and privacy concerns [2].

The privacy concerns caused by retaining data sets in cloud are important. Storage and computation services in cloud are equivalent from an economical perspective because they are charged in proportion to their usage. Thus, cloud users can store valuable data sets selectively when processing original data sets in data intensive applications like medical diagnosis, in order to curtail the overall expenses by avoiding frequent re-computation to obtain these data sets. Usually, intermediate data sets

ISSN 2348 - 7968

in cloud are accessed and processed by multiple parties, but rarely controlled by original data set holders. This enables an adversary to collect intermediate data sets together and menace privacy-sensitive information from them, bringing considerable economic loss or severe social reputation impairment to data owners.

Existing technical approaches for preserving the privacy of data sets stored in cloud mainly include encryption and anonymization. Encrypting all data sets, a straightforward and effective approach, is widely adopted in current research [3], [4], [5]. Processing on encrypted data sets efficiently is a challenging task, because most existing applications only run on unencrypted data sets. Although recent progress has been made in homomorphic encryption which theoretically allows performing computation on encrypted data sets, applying current algorithms are rather expensive due to their inefficiency. On the other hand, partial information of data sets, e.g., aggregate information, is required to expose to data users in most cloud applications like data mining and analytics. In such cases, data sets are anonymized rather than encrypted to ensure both data utility and privacy preserving. Current privacy-preserving techniques can withstand most privacy attacks on one single data set, while preserving privacy for multiple data sets is still a challenging problem. Thus, for preserving privacy of multiple data sets, anonymize all data sets first and then encrypt them before storing or sharing them in cloud. The volume of intermediate data sets is huge. Hence, that encrypting all intermediate data sets will lead to high overhead and low efficiency when they are frequently accessed or processed. Therefore, only part of datasets is encrypted rather than all for reducing privacy-protection cost.

In this paper, a novel approach is used to identify which intermediate data sets need to be encrypted while others do not, in order to satisfy privacy requirements given by data holders. A tree structure is modeled from generation relationships of data sets to analyze privacy propagation of data sets. As quantifying joint privacy leakage of multiple data sets efficiently is challenging, an upper bound constraint is exploit to confine privacy disclosure. Based on such a constraint, the problem of saving privacy-protection cost as a constrained optimization problem is modeled. This problem is then divided into a series of sub-problems by decomposing privacy leakage constraints. Finally, design a practical heuristic algorithm is designed accordingly to identify the data sets that need to be encrypted. Experimental results on real-world and extensive data sets demonstrate that privacy-protection cost of intermediate data sets can be significantly reduced with our approach over existing ones where all data sets are encrypted.

There are three major contributions. First, we formally demonstrate the possibility of ensuring privacy leakage requirements without encrypting all data sets when encryption is incorporated with anonymization to preserve privacy. Second, a practical heuristic algorithm is designed to identify which data sets need to be encrypted for privacy protection while the rest of them do not. Third, experiment results demonstrate that our approach can significantly reduce privacy-protection cost over existing approaches, which is quite beneficial for the cloud users who utilize cloud services in a pay-as-you-go fashion.

2 Related Work

Encryption is currently exploited by most existing research to ensure the data privacy in cloud [3], [4], [5]. Although encryption works well for data privacy in these approaches, it is necessary to encrypt and decrypt data sets frequently in many applications. Encryption is usually integrated with other methods to achieve cost reduction, high data usability and privacy protection. Roy et al. [8] investigated the data privacy problem caused by MapReduce and presented a system named Airavat which incorporates mandatory access control with differential privacy. Puttaswamy et al. [9] described a set of tools called Silverline that identifies all functionally encryptable data and then encrypts them to protect privacy. Zhang et al. [10] proposed a system named Sedic which partitions MapReduce computing jobs in terms of the security labels of data they work on and then assigns the computation without sensitive data to a public cloud. The sensitivity of data is required to be labeled in advance to make the above approaches available. Ciriani et al. proposed an approach that combines encryption and data fragmentation to achieve privacy protection for distributed data storage with encrypting only part of data sets. But integrate data anonymization and encryption together to fulfill cost-effective privacy protection.

The importance of retaining data sets in cloud has been widely recognized, but the research on privacy issues incurred by such data sets just commences. Davidson et al. [11], studied the privacy issues in workflow provenance, and proposed to achieve module privacy preserving and high utility of

provenance information via carefully hiding a subset of intermediate data. This general idea is similar to focuses on data privacy protection from an economical cost perspective while theirs concentrates majorly on functionality privacy of workflow modules rather than data privacy and also differs from theirs in several aspects such as data hiding techniques, privacy quantification and cost models.

The PPDP research community has investigated extensively on privacy-protection issues and made fruitful progress with a variety of privacy models and protection methods. Privacy principles such as k-anonymity [12] and l-diversity are put forth to model and quantify privacy, yet most of them are only applied to one single data set. Privacy principles for multiple data sets are also proposed, but they aim at specific scenarios such as continuous data publishing or sequential data releasing. The research in [13], exploits information theory to quantify the privacy via utilizing the maximum entropy principle.

3 Motivating Example

A motivating scenario is illustrated in Fig. 1 where an online health service provider, e.g., Microsoft HealthVault, has moved data storage into cloud for economical benefits. Original data sets are encrypted for confidentiality. Data users like governments or research centers access or process part of original data sets after anonymization. Data sets generated during data access or process are retained for data reuse and cost saving. Two independently generated intermediate data sets (Fig. 1a) and (Fig. 1b) in Fig. 1 are anonymized to satisfy 2-diversity, i.e., at least two individuals own the same quasi-identifier and each quasi-identifier corresponds to at least two sensitive values. Knowing that a lady aged 25 living in 21,400 (corresponding quasiidentifier is h214_; female; youngi) is in both data sets, an adversary can infer that this individual suffers from HIV with high confidence if Fig. 1a and Fig. 1b are collected together. Hiding Fig. 1a or Fig. 1b by encryption is a promising way to prevent such a privacy breach. Assume Fig. 1a and Fig. 1b are of the same size, the frequency of accessing Fig. 1a is 10 and that of Fig. 1b is 100. We hide Fig. 1a to protect privacy because this can incur less expense than hiding Fig. 1b. In most real-world applications, a large number of data sets are involved. Hence, it is challenging to identify which data sets should be encrypted to ensure that privacy leakage requirements are satisfied while keeping the hiding expenses as low as possible.

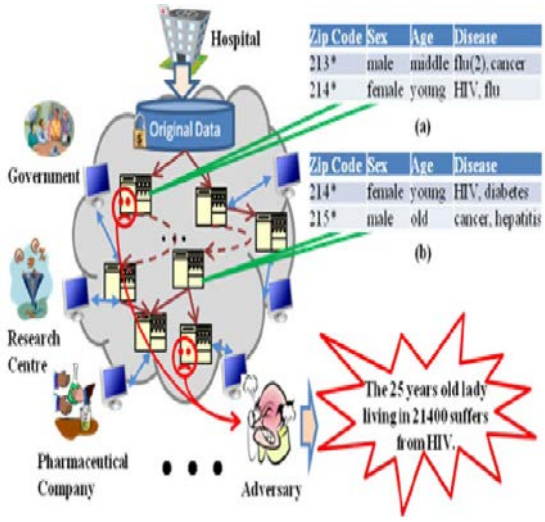

Fig.1: A scenario showing privacy threats due to data sets.

3.1 Problem Analysis

3.1.1 Sensitive Intermediate Data Set Management

Data provenance is employed to manage intermediate data sets in this research. Provenance is commonly defined as the origin, source or history of derivation of some objects and data, which can be reckoned as the information upon how data were generated. Reproducibility of data provenance can help to regenerate a data set from its nearest existing predecessor data sets rather than from scratch. Here, the information recorded in data provenance is leveraged to build up the generation relationships of data sets.

There are several basic notations below. Let do be a privacy-sensitive original data set. We use D = {d1, d2, . . . , dn} to denote a group of data sets generated from do where n is the number of data sets. Note that the notion of data herein refers to both intermediate and resultant data. Directed Acyclic Graph (DAG) is exploited to capture the topological structure of generation relationships among these data sets.

Definition 1 (Sensitive intermediate data set graph)

A DAG representing the generation relationships of data sets D from do is defined as a Sensitive Intermediate data set Graph, denoted as SIG. Formally, SIG = {V ,E}, where V = {do] U D, E is a set of directed edges. A directed edge {dp, dc} in E

ISSN 2348 - 7968

means that part or all of dc is generated from dp, where dp, dc ε {do} U D.

Definition 2(Sensitive intermediate data set tree (SIT))

An SIG is defined as a Sensitive data set Tree if it is a tree structure. The root of the tree is do. An SIG or SIT not only represents the generation relationships of an original data set and its intermediate data sets, but also captures the propagation of privacy-sensitive information among such data sets. Generally, the privacy sensitive information in do is scattered into its offspring data sets. Hence, an SIG or SIT can be employed to analyze privacy disclosure of multiple data sets. In this paper, an SIT is first developed, and then extends it to an SIG with minor modifications.

An data set is assumed to have been anonymized to satisfy certain privacy requirements. However, putting multiple data sets together may still invoke a high risk of revealing privacy-sensitive information, resulting in violating the privacy requirements. Privacy leakage of a data set d is denoted as PLs(d), meaning the privacy-sensitive information obtained by an adversary after d is observed. The value of PLs(d) can be deduced directly from d. Similarly, privacy leakage of multiple data sets in D is denoted as PLm(D), meaning the privacy-sensitive information obtained by an adversary after all data sets in D are observed. It is challenging to acquire the exact value of PLm(D) due to the inference channels among multiple data sets [13].

3.2.2 Privacy-Protection Cost Problem

Privacy protection cost of data sets stems from frequent en/decryption with charged cloud services. Cloud service venders have set up various pricing models to support the pay-as-you-go model, e.g., Amazon Web Services pricing model. Practically, en/decryption needs computation power, data storage, and other cloud services. To avoid pricing details and focus on the discussion of our core ideas, combine the prices of various services required by en/decryption into one. This combined price is denoted as PR. PR indicates the overhead of en/decryption on per GB data per execution.

An attribute vector is employed to frame several important properties of the data set di. The vector is denoted as {Si, Flagi, fi, PLi}. The term Si represents the size of di. The term Flagi, a dichotomy label, signifies whether di is hidden. The term fi indicates the frequency of accessing or processing di. If di is labeled as hidden, it will be en/decrypted every time when accessed or processed. Thus, the larger fi is, the more cost will be incurred if di is hidden. Usually, fi is estimated from the data provenance. The term PLi is the privacy leakage through di, and is computed by PLs(di).

Data Sets in D can be divided into two sets. One is for encrypted data sets, denoted as Denc. The other is for unencrypted data sets, denoted as Dune. Then, the equations Denc U Dune = D and Denc ∩ Dune = ε hold. The pair {Denc, Dune} is defined as a global privacy-protection solution. The privacy protection cost incurred by a solution {Denc, Dune} is denoted as Cpp({Denc, Dune}). With the notations framed above, the cost Cpp({Denc, Dune}) in a given period [T0, T], can be deduced by the following formula:

$$Cpp(\{Denc, Dune\}) = \varepsilon \left(\Sigma di \epsilon Denc\ Si \cdot PR \cdot fi \cdot t\right) \cdot dt.$$

The privacy-preserving cost rate for Cpp({Denc, Dune}), denoted as CRpp, is defined as follows:

$$CRpp = \Sigma\ di\ \varepsilon\ Denc\ Si \cdot PR \cdot fi.$$

In the real world, Si and fi possibly vary over time, but here they are static so that we can concisely present the core ideas of our approach. The dynamic case will be explored in our future work. With this assumption, CRpp determines Cpp({Denc, Dune}) in a given period. The problem of how to make privacy-protection cost as low as possible given an SIT can be modeled as an optimization problem on CRpp:

$$\text{Minimize } CRpp = \Sigma\ di\ \varepsilon\ Denc\ Si \cdot PR \cdot fi,$$
$$Denc$$

The privacy leakage caused by unencrypted data sets in Dune must be under a given threshold.

Definition 3 (Privacy leakage constraint)

Let ε be the privacy leakage threshold allowed by a data holder, then a privacy requirement can be represented as PLm(Dune) ≤ ε, Dune. This privacy requirement is defined as a Privacy Leakage Constraint, denoted as PLC.

With a PLC, the problem becomes a constrained optimization problem. So, we can save privacy-protection cost by minimizing it. As it is challenging to obtain the exact value of PLm(Dune), this approach is to address the problem via substituting the PLC with one of its sufficient conditions.

ISSN 2348 - 7968

4 Privacy Protection and Privacy Leakage of Novel Approach

It is fundamental to measure privacy leakage of anonymized data sets to quantitatively describe how much privacy is disclosed.

4.1 Single Intermediate Data Set Privacy Representation

The privacy-sensitive information is essentially regarded as the association between sensitive data and individuals. An original sensitive data set is denoted as do; an anonymized intermediate data set is denoted as d*; the set of sensitive data as SD; and the set of quasi-identifiers as QI. Let S denote a random variable ranging in SD, and Q be a random variable ranging within QI. Suppose s ε SD and q ϵ QI. The joint possibility of an association {s, q}, denoted as p(S = s, Q = q) (abbr. p(s, q)), is the information that adversaries intend to recover [13]. When an adversary has observed d* and a quasi-identifier q, the conditional possibility p(S = s|Q = q) representing intrinsic privacy-sensitive information of an individual can be inferred. If p(S = s|Q = q) is deduced as a high value or even 1.0, the privacy of the individual with q will be awfully breached.

The approach proposed in is employed to compute the probability distribution P*(S,Q) of {s, q} in do after observing d*. Formally, PLs(d*) is defined as

$$PLs(d^*) = H(S,Q) - H^*(S,Q)$$

H(S,Q) is the entropy of random variable {S,Q} before d* is observed, while H*(S,Q) is that after observation. P(Q, S) is estimated as a uniform distribution according to the maximum entropy principle [25]. Based on this, H(S,Q) can be computed by H(S,Q) = log(|QI| · |SD|). H·(S,Q) is calculated from distribution P*(S,Q) by

$$H^*(S,Q) = - \Sigma_{q \varepsilon QI, s \varepsilon SA} p(s, q) \cdot \log(p(s, q))$$

4.2 Privacy Leakage of Multiple Intermediate Data Sets

The value of the joint privacy leakage incurred by multiple data sets in D = {d1, d2, . . . , dn}, n ε N, is defined by

$$PLm(D) = H(S,Q) - HD(S,Q)$$

H(S,Q) and HD(S,Q) are the entropy of {S,Q} before and after data sets in D are observed, respectively. H(S,Q) = log(|QI| · |SD|). HD(S,Q) can be calculated once P(S,Q) is estimated after data sets in D are observed. Given the relationship between ϵ and PLm(Dune) in PLC, ε ranges in the interval [max1≤i≤n{PLs(di)}, log(|QI| · |SA|)].

Zhu et al. [13] proposed an approach to indirectly estimate P(S,Q) for multiple data sets with the maximum entropy principle. But this approach becomes inefficient when many data sets are involved because the number of variables and constraints possibly increase sharply when the number of data sets grows. According to the experiments in [13], it takes more than 200 minutes to quantify the privacy of two data sets with 6,000 records. Further, since Dune is uncertain before a solution is found, we need to try different Dune, where Dune ε 2D. So, the inefficiency will become unacceptable in many applications where a large number of intermediate data sets are involved.

Fortunately, the PLC can be achieved without exactly acquiring PLm(Dune) because our goal is to control the privacy disclosure caused by multiple data sets. A promising approach is to substitute the PLC with its sufficient conditions. Specifically, our approach is to replace the exact value of PLm(Dune) with one of its upper bounds which can be calculated efficiently.

4.3 Privacy-Protection Cost Reducing Algorithm

In this section, a algorithm is designed to reduce privacy-protection cost. In the state search space for an SIT, a state node SNi in the layer Li herein refers to a vector of partial local solutions, i.e., SNi corresponds to { ε1j1, . . . , _ εiji }. Note that the state-search tree generated according to an SIT is different from the SIT itself, but the height is the same. Appropriate heuristic information is quite vital to guide the search path to the goal state. The goal state in our algorithm is to find a near-optimal solution in a limited search space.

Heuristic values are obtained via heuristic functions. A heuristic function, denoted as f(SNi), is defined to compute the heuristic value of SNi. Generally, f(SNi) consists of two parts of heuristic information, i.e., f(SNi) = g(SNi) + h(SNi), where the information g(SNi) is gained from the start state to the current state node SNi and the information h(SNi) is estimated from the current state node to the goal state, respectively.

Intuitively, the heuristic function is expected to guide the algorithm to select the data sets with small cost but high privacy leakage to encrypt. Based on this, g(SNi) is defined as g(SNi) = Ccur/(ε - i+1), where Ccur is the privacy-protection cost that has been incurred so far, ε is the initial privacy leakage threshold, and ε i+1 is the privacy leakage threshold for the layers after Li. Specifically, Ccur is calculated by Ccur = Σ dj ε U k=1 EDk (Sj · PR · fj). The smaller Ccur is, the smaller total privacy-protection cost will be. Larger (ε - ε i+1) means more data sets before Li+1 remain unencrypted in terms of the RPC property, i.e., more privacy protection expense can saved.

The value of h(SNi) is defined as h(SNi) = (ε i+1· Cdes · BFAVG)/PLAVG. Similar to the meaning of (ε - ε i+1) in g(SNi), smaller ε i+1 in h(SNi) implies more data sets before Li+1 are kept unencrypted. If a data set with smaller depth in an SIT is encrypted, more data sets are possibly unencrypted than that with larger depth, because the former possibly has more descendant data sets. For a state node SNi, the data sets in its corresponding EDk are the roots of a variety of subtrees of the SIT. These trees constitute a forest, denoted as F ε i. In h(SNi), Cdes represents the total cost of the data sets in F ε i. Potentially, the less Cdes is, the fewer data sets in following layers will be encrypted. BFAVG is the average branch factor of the forest Fεi , and can be computed by BFAVG = NE/NI , where NE is the number of edges and NI is the number of internal data sets in Fεi . Smaller BFAVG means the search space for sequent layers will be smaller, so that we can find a nearoptimal solution faster. The value of PLAVG indicates the average privacy leakage of data sets in Fεi. Heuristically, the algorithm prefers to encrypt the data sets which incur less cost but disclose more privacy-sensitive information. Thus, higher PLAVG means more data sets in Fπi should be encrypted to preserve privacy from a global perspective.

Algorithm 1: Privacy Protection Cost Reducing Heuristic

Description: - Iteratively identifies the datasets that need to be encrypted; achieving a low level privacy-protection cost under the constraint PCL1.

Input: - A SIT with root do; all attribute values of each datasets are given, i.e., size, frequency, privacy leakage; privacy requirement threshold ε.

Output: - A vector of local solutions { ε 1, ….. , ε h} that comprise a near-optimal global privacy-protection solutions; and the global privacy-protection cost Cglobal.

Step 1- Initialize the following variables

1.1 Define a priority queue: PQueue.

1.2 Construct the initial search node with the root of the SIT:

SN0 = {{ ε 0} ← {{do}, ε,f(SN) ←0,ED0 ← {do}, Ccur ← 0, ε 1 ← ε }, i.e., the five parameters are the current solutions, the current heuristic value, the current ED, the current cost and the privacy leakage requirement for the sequent layer.

1.3 SN0.

Step 2- Iteratively retrieve the search nodes from PQueue, and in turn add their child search node to PQueue.

2.1 Retrieve the search node with the highest heuristics from PQueue: SN1 ← PQueue.

2.2 Check whether ED1 = ε. If yes, a solution is found and the algorithm will go to step 3.

2.3 Label the datasets in CDEi as encrypted if their privacy leakage is larger than ε i. Sort the unlabeled datasets un CDEi acsendingly according to Ck/PLs (dk), dk ε CDEi: SORT(CDEi). If the number of unlabeled datasets are larger than M, only the first M datasets are considerd to generate candidate nodes.

2.4 Generate all the possible local solutions in Ai.

2.5 Select a solution from Ai: ε ← SELECT(Ai).:

1) Calculate the privacy leakage upper bound of this solution and the encryption cost: PLlocal ← ΣdεUDε PLs(d), Clocal ← ΣdεUDε (Sk · CR · fk), where ε ={EDε, UDε}.

2) Calculate the remaining privacy leakage εi+1 ← εi.- PLlocal.

2.6 Compute the heuristic value.

2.7 Construct new search node from the obtained values, add it to PQueue. Then go to Step 2.1.

Step 3- Obtain the global encryption cost Cglobal : Cglobal ← Ccur, and the corresponding solution {ε1, ….., εh}.

Algorithm 1 specifies the details of the heuristic algorithm. A priority queue is exploited to keep state nodes. Only the qualified state nodes that are added to the priority queue, i.e., the corresponding partial global solutions are feasible. To avoid the size of the priority queue increase dramatically, the algorithm only retains the state nodes with top K highest heuristic values. When determining to add child search nodes in layer Li+1

into the priority queue, the algorithm generates a local encryption solution from CDEi at first. The algorithm probably suffers from poor efficiency because it has to check all combinations of data sets in CDEi. To circumvent this situation, the algorithm sorts the data sets in CDEi according to the value Ck/PLs(dk), where dk ε CDEi and Ck = Sk · PR · fk. If |CDEi| is larger than a threshold M, only the first M data sets in the sorted CDEi will be examined while the remaining is set to be encrypted. Intuitively, data sets with higher privacy protection cost and lower privacy leakage are expected to remain unencrypted. The value Ck/PLs(dk) can help to guide the algorithm to find these data sets with a higher possibility. Hence, the algorithm is guided to approach the goal state in the state space as close as possible. Above all, in the light of heuristic information, the proposed algorithm can achieve a near optimal solution practically.

5 Conclusion and Future work

In this paper, novel approach is proposed that identifies which part of data sets needs to be encrypted while the rest does not based on their size, frequency, and privacy leakage requirement in order to save the privacy-protection cost. A tree structure has been modeled from the generation relationships of data sets to analyze privacy propagation among data sets. The problem of saving privacy preserving cost as a constrained optimization problem which is addressed by decomposing the privacy leakage constraints is modeled. A practical heuristic algorithm has been designed accordingly.

In accordance with various data and computation intensive applications on cloud, data set management is becoming an important research area. Privacy protection for data sets is one of important yet challenging research issues, and needs intensive investigation. With the contributions of this paper, planning to further investigate privacy aware efficient scheduling of intermediate data sets in cloud by taking privacy protection as a metric together with other metrics such as storage and computation. Optimized balanced scheduling strategies are expected to be developed toward overall highly efficient privacy aware data set scheduling.

References

[1] L. Wang, J. Zhan, W. Shi, and Y. Liang, "In Cloud, Can Scientific Communities Benefit from the Economies of Scale?," IEEE Trans. Parallel and Distributed Systems, vol. 23, no. 2, pp. 296-303, Feb. 2012.

[2] H. Takabi, J.B.D. Joshi, and G. Ahn, "Security and Privacy Challenges in Cloud Computing Environments," IEEE Security & Privacy, vol. 8, no. 6, pp. 24-31, Nov./Dec. 2010.

[3] H. Lin and W. Tzeng, "A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding," IEEE Trans. Parallel and Distributed Systems, vol. 23, no. 6, pp. 995-1003, June 2012.

[4] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data," Proc. IEEE INFOCOM' 11, pp. 829-837, 2011.

[5] M. Li, S. Yu, N. Cao, and W. Lou, "Authorized Private Keyword Search over Encrypted Data in Cloud Computing," Proc. 31st Int'l Conf. Distributed Computing Systems (ICDCS '11), pp. 383-392, 2011.

[6] B.C.M. Fung, K. Wang, and P.S. Yu, "Anonymizing Classification Data for Privacy Preservation," IEEE Trans. Knowledge and Data Eng., vol. 19, no. 5, pp. 711-725, May 2007.

[7] X. Zhang, C. Liu, J. Chen, and W. Dou, "An Upper-Bound Control Approach for Cost Effective Privacy Protection of Intermediate Data Set Storage in Cloud," Proc. Ninth IEEE Int'l Conf. Dependable, Autonomic and Secure Computing (DASC '11), pp. 518-525, 2011.

[8] I. Roy, S.T.V. Setty, A. Kilzer, V. Shmatikov, and E. Witchel, "Airavat: Security and Privacy for Mapreduce," Proc. Seventh USENIX Conf. Networked Systems Design and Implementation (NSDI '10), p. 20, 2010.

[9] K.P.N. Puttaswamy, C. Kruegel, and B.Y. Zhao, "Silverline: Toward Data Confidentiality in Storage-Intensive Cloud Applications," Proc. Second ACM Symp. Cloud Computing (SoCC '11), 2011.

[10] K. Zhang, X. Zhou, Y. Chen, X. Wang, and Y. Ruan, "Sedic: Privacy-Aware Data Intensive Computing on Hybrid Clouds," Proc. 18th ACM Conf. Computer and Comm. Security (CCS '11), pp. 515-526, 2011.

[11] S.B. Davidson, S. Khanna, T. Milo, D. Panigrahi, and S. Roy, "Provenance Views for Module Privacy," Proc. 30th ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems (PODS '11), pp. 175-186, 2011.

[12] P. Samarati, "Protecting Respondents' Identities in Microdata Release," IEEE Trans. Knowledge and Data Eng., vol. 13, no. 6, pp. 1010-1027, Nov. 2001.

[13] G. Wang, Z. Zutao, D. Wenliang, and T. Zhouxuan, "Inference Analysis in Privacy Preserving Data Re-Publishing," Proc. Eighth IEEE Int'l Conf. Data Mining (ICDM '08), pp. 1079-1084, 2008.

[14] J.A. Kelner and A. Madry, "Faster Generation of Random Spanning Trees," Proc. 50th Ann. IEEE Symp. Foundations of Computer Science (FOCS '09), pp. 13-21, 2009.

[15] Xuyun Zhang, Chang Liu, Surya Nepal, Suraj Pandey, and Jinjun Chen, Member, IEEE "A Privacy Leakage Upper Bound Constraint-Based Approach for Cost-Effective Privacy Preserving of Intermediate Data Sets in Cloud" IEEE transactions on parallel and distributed system, vol. 24, no. 6, june 2013