International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Cost-Effective Privacy Preserving of Intermediate Data Sets in Cloud using Privacy Leakage Upper Bound Constraint-Based Approach

Ravindra Suresh Kamble[1], Sheikh Gouse[2]

[1]Department of Computer Science & Engineering, MLRIT Institutions of Technology, Hyderabad, Telangana, India

Abstract: ***Cloud computing is an evolving paradigm with tremendous momentum, but its unique aspects exacerbating security and privacy challenges. Cloud computing provides massive computation power and storage capacity which enable users to deploy computation and data-intensive applications without infrastructure investment. Along the processing of such applications, a large volume of intermediate data sets will be generated, and often stored to save the cost of recomputing them. However, preserving the privacy of intermediate data sets becomes a challenging problem because adversaries may recover privacy-sensitive information by analyzing multiple intermediate data sets. Encrypting ALL data sets in cloud is widely adopted in existing approaches to address this challenge. But we argue that encrypting all intermediate data sets are neither efficient nor cost-effective because it is very time consuming and costly for data-intensive applications to end decrypt data sets frequently while performing any operation on them. In this paper, we propose a novel upper bound privacy leakage constraint-based approach to identify which intermediate data sets need to be encrypted and which do not, so that privacy-preserving cost can be saved while the privacy requirements of data holders can still be satisfied. Evaluation results demonstrate that the privacy-preserving cost of intermediate data sets can be significantly reduced with our approach over existing once where all data sets are encrypted.***

Keywords: Cloud computing, data sets, privacy preserving, Data privacy management, privacy upper bound.

1. Introduction

1.1 What is cloud computing?

Cloud computing [1] is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the common use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams [1]

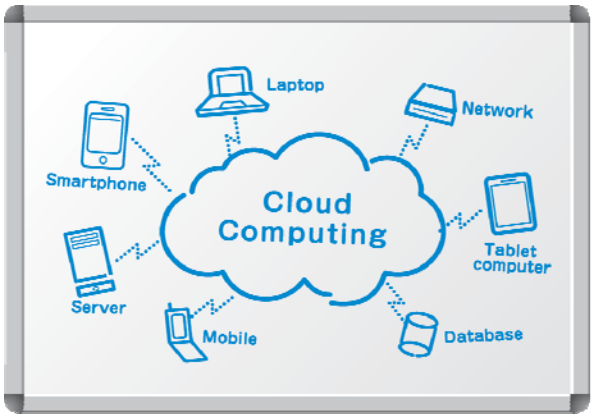

Figure1: Structure of cloud computing

Cloud computing entrusts remote services with a user's data, software and computation. Cloud computing consists of hardware and software resources made available on the Internet as managed third-party services. These services typically provide access to advanced software applications and high-end networks of server computers [2].

1.2 How Cloud Computing Works?

The goal of cloud computing is to apply traditional supercomputing, or high-performance computing power, normally used by military and research facilities, to perform tens of trillions of computations per second, in consumer-oriented applications such as financial portfolios, to deliver personalized information, to provide data storage or to power large, immersive computer games [3]

The cloud computing uses networks of large groups of servers typically running low-cost consumer PC technology with specialized connections to spread data-processing chores across them. This shared IT infrastructure contains large pools of systems that are linked together. Often, virtualization techniques are used to maximize the power of cloud computing.

1.3 Characteristics and Services Models

The salient characteristics of cloud computing based on the definitions provided by the National Institute of Standards and Terminology (NIST) are outlined below:

- **On-demand self-service**: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service's provider [1].
- **Broad network access**: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs) [4].
- **Resource pooling**: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources

dynamically assigned and reassigned according to consumer demand [7]. There is a sense of location-independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or data center). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines.

- **Rapid elasticity**: Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time [2].
- **Measured service**: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be managed, controlled, and reported providing transparency for both the provider and consumer of the utilized service [5].

1.4 Services Models

Cloud Computing comprises three different service models, namely [1] [6]

- Infrastructure-as-a-Service (IaaS),
- Platform-as-a-Service (PaaS),
- Software-as-a-Service (SaaS).

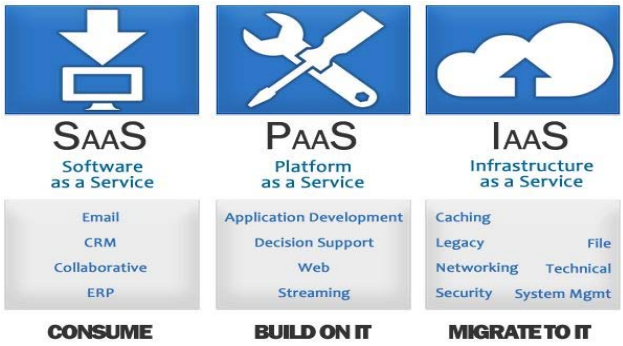

Figure 2: Cloud Service Models

The figure 2 shows the three service models along with example. The three service models or layer are completed by an end user layer that encapsulates the end user perspective on cloud services. The model is shown in figure below. If a cloud user accesses services on the infrastructure layer, for instance, she can run her own applications on the resources of a cloud infrastructure and remain responsible for the support, maintenance, and security of these applications herself. If she accesses a service on the application layer, these tasks are normally taken care of by the cloud service provider.

1.5 Privacy of data sets

Parallel dataflow programs generate enormous amounts of distributed data that are short-lived, yet are critical for completion of the job and for good run-time performance. We call this class of data as intermediate data. [2] This paper is the first to address intermediate data as a first-class citizen, specifically targeting and minimizing the effect of run-time server failures on the availability of intermediate data, and thus on performance metrics such as job completion time [9]. We propose new design techniques for a new storage system called ISS (Intermediate Storage System), implement these techniques within Hadoop, and experimentally evaluate the resulting system.

The privacy concerns caused by retaining intermediate data sets in cloud are important but they are paid little attention. Storage and computation services in cloud are equivalent from an economical perspective because they are charged in proportion to their usage [10]. Thus, cloud users can store valuable intermediate 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 recomputation to obtain these data sets. Such scenarios are quite common because data users often reanalyze results, conduct new analysis on intermediate data sets, or share some intermediate results with others for collaboration. Without loss of generality, the notion of intermediate data set herein refers to intermediate and resultant data sets [9] .

However, the storage of intermediate data enlarges attack surfaces so that privacy requirements of data holders are at risk of being violated. Usually, intermediate data sets 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. But, little attention has been paid to such a cloud-specific privacy issue.

2. Data Privacy Management Platform

- Comprehensive solution including Privacy Assessments and Certifications, Monitoring Tools and Compliance Controls.
- Manage privacy globally across all your online channels - web, cloud, apps, and ads.
- Powered by a robust cloud-based technology infrastructure.
- Enables businesses to protect their brand, build trust, and maintain compliance.

The following figure shows Data privacy Management platform in cloud.

Figure 3: Data privacy Management platform

3. Data Privacy and Security

Information is increasingly pervasive within the business enterprise. Management of the information flow within and beyond the organization requires special attention to information that is sensitive, such as personally identifiable information (PII) or protected health information (PHI). Businesses face potential litigation, operational and compliance issues, and damage to their reputations if they fail to properly protect critical information. Costs and losses may be significant but damage to reputation may be unrecoverable.

Protiviti's Data Security [3] and Privacy Management professionals provide a full spectrum of assessment, transformation, and management services to help organizations identify and address privacy exposures before they become problems. We help companies identify the information they need to treat as private. We create the processes and metrics needed to manage the information to meet both business and regulatory requirements. We can also ensure there is operational alignment with existing records management policies and programs. If necessary, our e-Discovery and Forensics team can support you in any litigation activities you may pursue. Our services include:

1. Data Governance
2. Data Classification
3. Data Leakage
4. Encryption & Storage Strategy & Implementation
5. Privacy Management & Implementation
6. PCI, HIPAA, HITRUST and Other Security Compliance Readiness & Assessment

4. Data Encryption and Decryption

Encryption [4] is the process of translating plain text data *(*plaintext*)* into something that appears to be random and meaningless (ciphertext*)*. Decryption [5] is the process of converting ciphertext back to plaintext.

To encrypt more than a small amount of data, symmetric encryption is used. A symmetric key is used during both the encryption and decryption processes. To decrypt a particular piece of ciphertext, the key that was used to encrypt the data must be used.

The goal of every encryption algorithm is to make it as difficult as possible to decrypt the generated ciphertext without using the key. If a really good encryption algorithm is used, there is no technique significantly better than methodically trying every possible key. For such an algorithm, the longer the key, the more difficult it is to decrypt a piece of ciphertext without possessing the key.

It is difficult to determine the quality of an encryption algorithm. Algorithms that look promising sometimes turn out to be very easy to break, given the proper attack. When selecting an encryption algorithm, it is a good idea to choose one that has been in use for several years and has successfully resisted all attacks.

4.1 Kinds of Encryption

4.1.1 Symmetric key encryption

In symmetric-key schemes, the encryption and decryption keys are the same. Thus communicating parties must have the same key before they can achieve secret communication.

4.1.2 Public key encryption

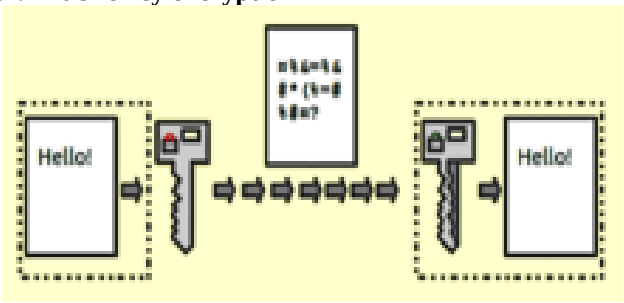

Figure 4: Illustration of how a file or document is sent using Public key encryption.

In public-key encryption schemes, the encryption key is published for anyone to use and encrypt messages. However, only the receiving party has access to the decryption key that enables messages to be read.[6] Public-key encryption was first described in a secret document in 1973;[6] before then all encryption schemes were symmetric-key (also called private-key).

5. Related Work

Existing technical approaches for preserving the privacy of data sets stored in cloud mainly include encryption and anonymization. On one hand, encrypting all data sets, a straightforward and effective approach, is widely adopted in current research.

However, processing on encrypted data sets efficiently is quite a challenging task, because most existing applications only run on unencrypted data sets. Although recent progress has been made in homomorphic encryption [7] 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 like generalization can withstand most privacy attacks on one single data set, while preserving privacy for multiple data sets is still a challenging problem. The following figure 5 shows a scenario showing privacy threats due to intermediate data sets.

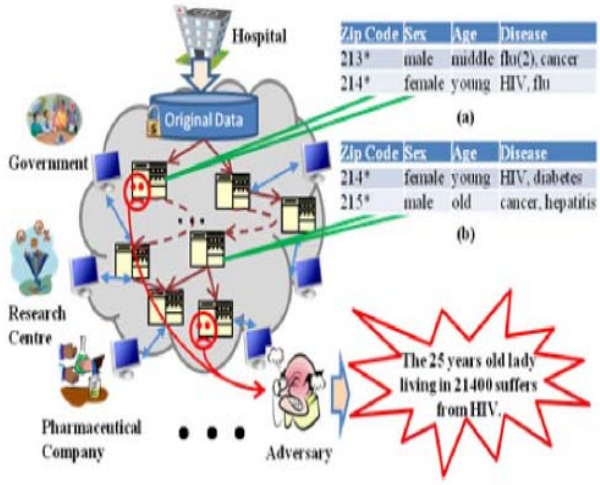

Figure 5: A scenario showing privacy threats due to intermediate data sets.

Figure 5 shows an online health service provider, e.g., Microsoft Health Vault [8], 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. Intermediate data sets generated during data access or process are retained for data reuse and cost saving. Two independently generated intermediate data sets (Fig. 5a) and (Fig. 5b) in Fig. 5 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 [9]. Knowing that a lady aged 25 living in 21,400 (corresponding quasi identifier is h214_; female; youngi) is in both data sets, an adversary can infer that this individual suffers from HIV with high confidence if Fig. 5a and Fig. 5b are collected together. Hiding Fig. 5a or Fig. 5b by encryption is a promising way to prevent such a privacy breach. Assume Fig. 5a and Fig. 5b are of the same size, the frequency of accessing Fig. 5a is 10 and that of Fig. 5b is 100. We hide Fig. 5a to preserve privacy because this can incur less expense than hiding Fig. 5b.

6. Proposed Work

Encrypting all intermediate data sets will lead to high overhead and low efficiency when they are frequently accessed or processed. As such, we propose to encrypt part of intermediate data sets rather than all for reducing privacy-preserving cost. In this paper, we propose a novel approach 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 intermediate data sets to analyze privacy propagation of data sets.

As quantifying joint privacy leakage of multiple data sets efficiently is challenging, we exploit an upper bound constraint to confine privacy disclosure. Based on such a constraint, we model the problem of saving privacy-preserving cost as a constrained optimization problem. This problem is then divided into a series of sub-problems by decomposing privacy leakage constraints. Finally, we design a practical heuristic algorithm accordingly to identify the data sets that need to be encrypted.

Block Diagram for Proposed Work is shown below which contains datasets along with various generalizations.

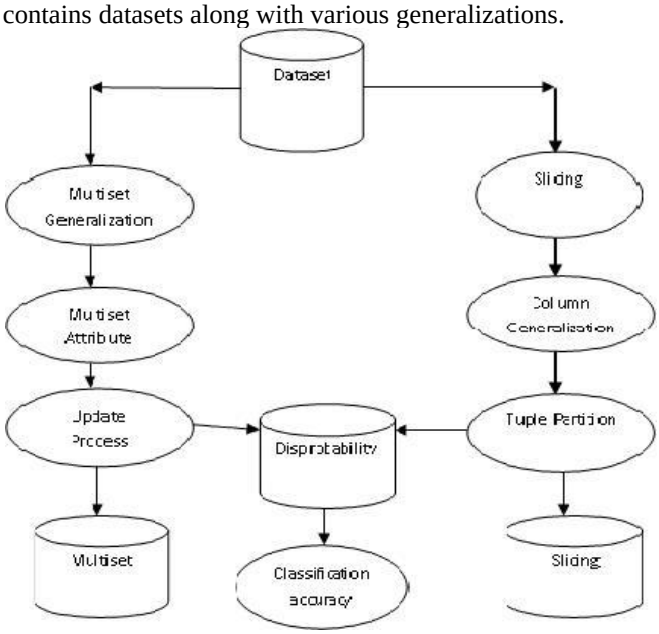

Figure 6: Block diagram

The major contributions of our research are threefold.

- First, we formally demonstrate the possibility of ensuring privacy leakage requirements without encrypting all intermediate data sets when encryption is incorporated with anonymization to preserve privacy.
- Second, we design a practical heuristic algorithm to identify which data sets need to be encrypted for preserving privacy while the rest of them do not.
- Third, experiment results demonstrate that our approach can significantly reduce privacy-preserving cost over existing approaches, which is quite beneficial for the cloud users who utilize cloud services in a pay-as-you-go [1] fashion.

7. Modules

For the Cost-Effective Privacy Preserving of Intermediate Data Sets in Cloud [10] using Privacy Leakage Upper Bound Constraint-Based Approach, following modules are used for Privacy-preserving:

1. Data Storage Privacy Module.
2. Privacy Preserving Module.
3. Intermediate Dataset Module.

4. Privacy Upper Bound Module.

7.1 Data Storage Privacy Module

The privacy concerns caused by retaining intermediate datasets in cloud are important but they are paid little attention. A motivating scenario is illustrated where an on-line health service provider, e.g., Microsoft Health Vault has moved data storage into cloud for economical benefits. Original datasets are encrypted for confidentiality. Data users like governments or research centre's access or process part of original datasets after anonymization. Intermediate datasets generated during data access or process are retained for data reuse and cost saving. We proposed an approach that combines encryption and data fragmentation to achieve privacy protection for distributed data storage with encrypting only part of datasets.

7.2 Privacy Preserving Module

Privacy-preserving techniques like generalization can withstand most privacy attacks on one single dataset, while preserving privacy for multiple datasets is still a challenging problem. Thus, for preserving privacy of multiple datasets, it is promising to Anonymized all datasets first and then encrypts them before storing or sharing them in cloud. Privacy-preserving cost of intermediate datasets stems from frequent en/decryption with charged cloud services.

7.3 Intermediate Dataset Module

An intermediate dataset [11] is assumed to have been anonymized to satisfy certain privacy requirements. However, putting multiple datasets together may still invoke a high risk of revealing privacy-sensitive information, resulting in violating the privacy requirements. Data provenance is employed to manage intermediate datasets in our 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 was generated. Re-producibility of data provenance can help to regenerate a dataset from its nearest existing predecessor datasets rather than from scratch.

7.4 Privacy Upper Bound Module

Privacy quantification of a single data-set is stated. We point out the challenge of privacy quantification [12] of multiple datasets and then derive a privacy leakage upper-bound constraint correspondingly. We propose an upper-bound constraint based approach to select the necessary subset of intermediate datasets that needs to be encrypted for minimizing privacy-preserving cost. The privacy leakage upper-bound constraint is decomposed layer by layer.

8. Privacy-Preserving Cost Reducing Heuristic Algorithm

Heuristic Approach Heuristic approaches can be further categorized into distortion based schemes and blocking based schemes. To hide sensitive item sets, distortion based scheme changes certain items in selected transactions from 1's to 0's and vice versa. Blocking based scheme replaces certain items in selected transactions with unknowns. These approaches have been getting focus of attention for majority of the researchers due to their efficiency, scalability and quick responses.

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 h_1j1; . . . ; _iji i, where _kjk 2 _k, 1 _ k _ i. 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 fðSNiÞ, is defined to compute the heuristic value of SNi. Generally, fðSNiÞ consists of two parts of heuristic information, i.e., fðSNiÞ ¼ gðSNiÞ þ hðSNiÞ, where the information gðSNiÞ is gained from the start state to the current state node SNi and the information hðSNiÞ 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, gðSNiÞ is defined as gðSNiÞ ¼_ Ccur=ð" _ "iþ1Þ, where Ccur is the privacy preserving cost that has been incurred so far, " is the initial privacy leakage threshold, and "iþ1 is the privacy leakage threshold for the layers after Li. Specifically, Ccur is calculated by Ccur ¼ Pdj2[ik¼1EDkðSj _ PR _ fjÞ.

The smaller Ccur is, the smaller total privacy-preserving cost will be. Larger ð" _ "iþ1Þ means more data sets before Liþ1 remain unencrypted in terms of the RPC property, i.e., more privacy preserving expense can saved. The value of hðSNiÞ is defined as hðSNiÞ ¼ ð"iþ1_ Cdes _ BFAVGÞ=PLAV G. Similar to the meaning of ð" _ "iþ1Þ in gðSNiÞ, smaller "iþ1 in hðSNiÞ implies more data sets before Liþ1 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_i. In hðSNiÞ, Cdes represents the total cost of the data sets in F_i , and is computed via Cdes ¼ P dl2EDk P dj2PDðdlÞðSj _ CR _ fjÞ.

Potentially, the less Cdes is, the fewer data sets in following layers will be encrypted. BFAVG is the average branch factor of the forest F_i , and can be computed by BFAVG ¼ NE=NI , where NE is the number of edges and NI is the number of internal data sets in F_i . Smaller BFAVG means the search space for sequent layers will be smaller, so that we can find a near optimal solution faster. The value of PLAVG indicates the average privacy leakage of data sets in F_i , calculated by PLAVG ¼Pdl2EDkP dj2PDðdlÞ PLsðdjÞ=NI . 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 F_i should be encrypted to preserve privacy from a global perspective. Based on the above analysis, the heuristic value of the search node SNi can be computed by the formula: f SNi ð Þ¼Ccur=ð" _ "iþ1Þ þ ð"iþ1 _ Cdes _ BFAVGÞ=PLAVG

9. Results

The comparison of privacy preserving cost for encrypting all the intermediate datasets [11] in existing system and encrypting only part of intermediate datasets in our approach shows that we are reducing the privacy p reserving cost by using our approach as shown in the figure7

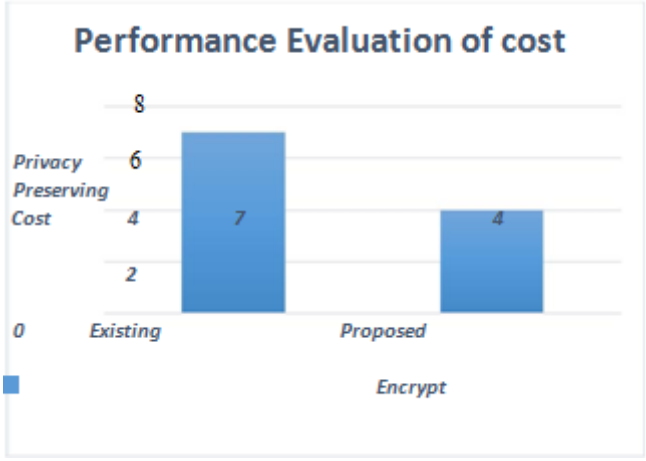

Figure 7: Reducing the privacy preserving cost by our approach

In the figure7 the vertical axis represents the cost required to encrypt the datasets and the horizontal axis shows the two categories existing and proposed, the shaded bars in the graph shows the encryption cost required. By observing the existing and proposed encryption costs we can evaluate the performance of our approach. By using our approach we can also prove that the time consuming is very less for encrypting only part of intermediate datasets compared with the existing approaches can be shown in the figure 8

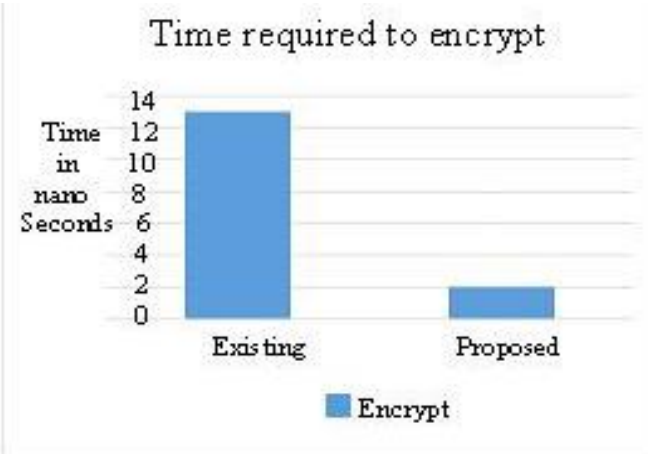

Figure 8: Result of time comparison for existing and our approach

In the figure8 the vertical axis represents the time required to encrypt the dataset in nanoseconds and the horizontal axis has two categories existing and proposed and t he shaded bars shows the encrypt time. We can easily analyze the time required to encrypt the datasets in the existing and our approach.

By comparing the cost for encrypting all the intermediate datasets and only part of intermediate datasets in the cloud we are saving the privacy preserving cost it can be shown in the following equation.

CSAV=CALL-CHEU

Here CSAV is the privacy preserving cost saved, CALL is the privacy preserving cost for encrypting all the intermediate datasets and CHEU is the privacy preserving cost for encrypting only part of intermediate datasets in the cloud. The resultant of our approach shows that the saving cost should be increases going on increasing the threshold value.

10. Conclusion

In accordance with various data and computation intensive applications on cloud, intermediate data set management is becoming an important research area. Privacy preserving for intermediate data sets is one of important yet challenging research issues, and needs intensive investigation. With the contributions of this paper, we are planning to further investigate privacy aware efficient scheduling of intermediate data sets in cloud by taking privacy preserving 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. We have proposed an approach that identifies which part of intermediate data sets needs to be encrypted while the rest does not, in order to save the privacy preserving cost. A tree structure has been modeled from the generation relationships of intermediate data sets to analyze privacy propagation among data sets. We have modeled the problem of saving privacy-preserving cost as a constrained optimization problem which is addressed by decomposing the privacy leakage constraints.

References

[1] M.Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A.Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M.Zaharia, “A View of Cloud Computing,” Comm. ACM, vol. 53,no. 4, pp. 50-58, 2010

[2] D. Yuan, Y. Yang, X. Liu, and J. Chen, “On-Demand Minimum Cost Benchmarking for Intermediate Data Set Storage in Scientific Cloud Workflow Systems,” J. Parallel Distributed Computing,vol. 71, no. 2, pp. 316-332, 2011

[3] D. Zissis and D. Lekkas, “Addressing Cloud Computing Security Issues,” Future Generation Computer Systems, vol. 28, no. 3, pp. 583-592, 2011.

[4] "Encryption Basics | EFF Surveillance Self-Defense Project." Encryption Basics | EFF Surveillance Self-Defense Project. Surveillance Self-Defense Project, n.d. Web. 06 Nov. 2013. https://ssd.eff.org/tech/encryption.

[5] Goldreich, Oded. Foundations of Cryptography: Volume 2, Basic Applications. Vol. 2. Cambridge university press, 2004
[6] Bellare, Mihir. "Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements." Springer Berlin Heidelberg, 2000. Page 1.
[7] Damien Stehle; Ron Steinfeld (2010-05-19). "Faster Fully Homomorphic Encryption" (PDF). International Association for Cryptologic Research. Retrieved 2010-09-15.
[8] Microsoft HealthVault, http://www.microsoft.com/health/ww/products/Pages/healthvault.aspx, July 2012.
[9] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam “L-Diversity: Privacy Beyond K-Anonymity,” ACM Trans. Knowledge Discovery from Data, vol. 1, no. 1, article 3, 2007.
[10] S.Hemalatha, S.Alaudeen Basha “Enabling for Cost-Effective Privacy Preserving of Intermediate Data Sets in Cloud” International Journal of Scientific and Research Publications, Volume 3, Issue 10, October 2013
[11] C. Lakshmi, “ An Approach for Privacy Preserving Cost of Intermediate Data Set in Cloud”, International Journal of Advanced Research in Computer Science and Software Engineering 4(5), May-2014,pp107-111
[12] W. Du, Z. Teng, and Z. Zhu, “Privacy-Maxent: Integrating Background Knowledge in Privacy Quantification,” Proc. ACMSIGMOD Int’l Conf. Management of Data (SIGMOD ’08), pp. 459-472, 2008.

Author Profile

Mr. Ravindra Suresh Kamble student of MLR Institutions of Technology, Hyderabad pursuing M. Tech degree in Computer Science and Engineering.

Sheikh Gouse B.Tech, M.Tech CSE. He is currently working in the Department of Computer Science and Engineering, MLRIT, Telangana, India. He is having 8 years of teaching experience**.** He is Certified in Oracle 9i: SQL & Java SE 6. His research interesting areas Programming (C&DS, C++, and JAVA), Data Mining, Software Engineering, Network Security & Computer Networks.