Privacy Preserving Distributed Cloud Storage

Praveenkumar Khethavath[1]*, Doyel Pal[2]

[1] Department of Mathematics, Engineering and Computer Science, LaGuardia Community College, Long Island City, NY 11101.
[2] Computer Science Department, Oklahoma State University, Stillwater, OK 74078.

* Corresponding author. Tel.: 718-482-5725; email: pkhethavath@lagcc.cuny.edu
Manuscript submitted February 15, 2015; accepted May 22, 2015.
doi: 10.17706/jcp.10.5.329-335

Abstract: Cloud computing models stores data and computing resources in virtual servers located in large data centres' and are managed using a centralized architecture. Distributed cloud uses resources provided by users who are geographically distributed over a large area. Distributed cloud is completely decentralised and resources are provided in a P2P fashion. This model of distributed cloud is based on mutual benefit for users and is free of cost. Distributed cloud provides all the services provided by regular cloud computing models. Since the resources are provided by regular users, security and privacy for data, storage and computing resources are main concerns. In this paper we propose a privacy preserving storage and file sharing mechanism among different users in a distributed cloud. This mechanism involves processing, storing and retrieval of encrypted data in a secure and privacy preserving manner.

Key words: Distributed cloud, homomorphic encryption, privacy, secure storage.

1. Introduction

Cloud computing [1]-[3] is an innovative model which has attracted plentiful attention from both industry as well as academia. Cloud computing allows users to manage resources in flexible and scalable manner. There is a substantial increase in number of IT firms, schools and regular users using the cloud resources. Cloud resources have been replacing the desktops, laptops, large storage devices, servers, etc. In order to effectively use these available resources we had introduced distributed cloud computing model [4]-[6]. Distributed cloud is formed using the existing resources and it provides storage and computation resources similar to existing cloud to users in a distributed peer-to-peer fashion. The distributed cloud is more than an existing cloud with multiple data centres distributed in different locations. Distributed cloud is dynamic in nature as users come in and out of the system.

Virtualization [7] is key concept for making cloud computing possible. Xen is one of the open source standard for hardware virtualization used by many cloud providers. Hyper-V, KVM, and Sun xVM are some of the other key virtualization management tools used. Cloud providers provide resources in form of various instances based on the user needs. For example Amazon provides for different sizes of virtual machine instances — Small, Medium, Large and Extra Large. These resources are scalable and can be increased or decreased according to the needs. Important information to be noted is we need not have huge data centres to run virtualization. These virtualization tools can be run on small machines like simple desktops, laptops.

Distributed cloud uses the virtualization techniques to effectively use these resources which are not

being used effectively. Cloud computing uses centralized architecture. With increase in number of users the network traffic has increased tremendously at data centre locations. Moreover it requires huge amounts of electricity to maintain these data centres. Distributed cloud can be used to mitigate these problems of cloud and use the unused resources effectively and thereby providing a green computing model.

Distributed cloud will provide all the services that a cloud can provide. Since these resources are provided by individual users located in a wide geographical area the network traffic is distributed across widely. In addition to that there won't be any single point of failure as it uses a completely decentralized architecture. Security and privacy will still be a concern in both distributed cloud as well as existing cloud models. Since users store and do computation on resources provided by other users we need a secure and privacy preserving mechanism to store and retrieve data. Confidentiality, integrity and availability are three major security concerns' along with privacy when resources are provided by others. Data confidentiality and integrity can be taken care using the encryption mechanisms. Availability depends on the resource management model. We proposed a novel resource management mechanism for distributed cloud using Multi-valued hash table mechanism and game theoretic model [6], [8]. In this paper we propose a secure and privacy preserving mechanism to share, store and retrieve data in a distributed cloud model.

This paper is organized as follows. Problem Description and Related Work is explained in Section 2 and 3 respectively, Security and privacy preserving mechanism along with the algorithms used are described in Section 4, Analysis of algorithm, performance is explained in Section 5 and conclusion in Section 6.

2. Problem Description

Resource provider is a user who provides the services in a distributed cloud. Resources can be any kind of service offered by resource provider. Software as a Service, Platform as a Service, Infrastructure as a Service and Storage as a Service can be provided by the resource provider.

Distributed cloud provides storage and computation resources in a peer-to-peer fashion over an open network. This leads to security and privacy issues that need to be resolved. Transferring the data and processing information to another user providing the service also comes with the responsibility of security and privacy. If we send the data and information in an encrypted format, the resource needs to have the key in order to update it. Providing security key to the resource provider leads to security and privacy issues. Moreover when we need to search the data using keywords, all the information regarding the data provider, data content, and the details of queried data can be monitored.

The security model considered is to encrypt the data and its names using homomorphic encryption scheme. To retrieve the data we use the encrypted file name to avoid advertising the real keywords we are searching for over the network. In order to preserve privacy of contents of file, i.e., actual data we split them and perform homomorphic encryption on the data and store it on the resource providers. We use homomorphic properties to query and validate the resultant data.

3. Related Work

3.1. Homomorphic Encryption

In this paper we use the homomorphic encryption technique to store or share the files in a privacy preserving manner among different resource providers. Homomorphic encryption technique allows specific types of arithmetic computations on ciphertext and when decrypted the plaintext matches with the result of arithmetic operations on plaintexts without seemingly inherent loss of the encryption. In ring theory, homomorphism is a mapping $\varphi : R \rightarrow S$ that respects both addition and multiplication. Therefore,

$$\forall x, y \in R, \qquad \varphi(x + y) = \varphi(x) + \varphi(y) \; and \; \varphi(xy) = \varphi(x)\,\varphi(y)$$

The homomorphic encryption which preserves a single operation such as addition or multiplication is

known as partially homomorphic encryption and those which preserves both of the addition and multiplication is known as fully homomorphic encryption. In this paper we use the multiplicative homomorphic encryption scheme. RSA, ElGamal [9], [10] encryption schemes support multiplicative homomorphic property. According to the multiplicative homomorphic encryption scheme if $E(x)$denotes the cipher text of plain text x then it satisfies

$$E(x_1) \cdot E(x_2) = E(x_1 \cdot x_2) \quad (1)$$

3.2. Distributed Cloud Model

In distributed cloud model, resource providers (RP) are distinctive i.e. these distributed cloud servers are individuals with resources to offer. Users of distributed cloud need to discover these resource providers and request them for using resources. Nodes in the distributed cloud can be both users and resource providers (RP). The distributed cloud uses a completely decentralized mechanism to discover and allocate resources. Users in the distributed cloud share recourses in a P2P fashion.

Distributed cloud has all the characteristics of existing cloud architecture including proper management of resources, scalability and data security. Network constraints will become less using the distributed cloud which is internet based, because data and resources distributed would be closer and is therefore more likely to be accessed from closer location. Moreover latency would be reduced because resources would be chosen closer to the users. We intend to use multi-valued hash table mechanism for resource discovery and use a game theoretic mechanism for allocating the resources available in a distributed cloud.

Privacy preserving is one of the main issues in both distributed and cloud computing systems. In [11], authors used public key homomorphic authenticator mechanism to achieve privacy preserving public cloud data auditing systems. In cloud since all the resources are in one location, auditing information should be preserved. In [12], author uses public key encryption based keyword search in cloud so that we need not decrypt entire message. But in distributed cloud we only search for exact keywords to find the file name and we don't want to decrypt the data to find out the value of keyword. This method of searching encrypted data over an encrypted pool of messages is computationally more expensive. In [13], [14], authors proposed homomorphic encryption to protect sensitive data stored in cloud data center and shared by multiple clients. Homomorphic encryption mechanism is used to preserve security and privacy of data by using additive or multiplicative properties. In order to check the similarity between two encrypted files, Euclidean distance can be used to measure the distance between them [15].

4. Security and Privacy Preserving Mechanism

In our proposed solution we are considering security and preserving privacy of data to be stored on resource providers. Since we are using distributed cloud we want to do computation, store data and retrieve them back from the resource providers. In order to retrieve the data we use file names or key words to search in the distributed cloud. The key idea is to split data/ file into multiple partitions and encrypt those partitions using a multiplicative homomorphic encryption scheme. For the file name of a data or keywords we do not use the split mechanism.

Preserving security and privacy of the data is a twofold process. First step is to encrypt the name of the data and in second step we encrypt the actual data. To store and retrieve data in distributed cloud we propose two algorithms. Algorithm 1 and Algorithm 2 describe the procedure to store data on resource providers and retrieve data from resource providers respectively. To store data on resource providers we encrypt the data name/ file name (Algorithm 1 Step 1) and distribute the data content among different resource providers (Algorithm 1 Step 2). Encrypting the data name / file name is a simple one step process. We split the original data content into multiple random partitions and encrypt each partitions using

multiplicative homomorphic encryption scheme. We calculate the Euclidean distance between different random encrypted data partitions before distribute them among different resource providers. Retrieving the data from different resource providers includes validating the data name/ file name (Algorithm 2 Step 1) and retrieving the data partitions (Algorithm 2 Step 2) from different resource providers. To validate the data name / file name we calculate the distance between encrypted data/ file name. If the distance is zero then we proceed to retrieve the data partitions of that particular data/ file. After all the partitions are retrieved, user measures the distances between the encrypted random partitions and compare them with distances stored. The same distance indicates that the file is not tampered.

Algorithm: 1. Storing data on resource providers

Input: File name f, Data m, Public key — private key pair (x, y).

Step 1: Encrypt the data name/ file name

1. Encrypt the file name f using a multiplicative homomorphic encryption scheme.

Step 2: Distribute the data among different Resource Providers

1. Split the data m into n number of partitions, such as, $m_1, m_2, \ldots, m_n$.
2. $for\ each$ data partitions $m_i\ \forall i \in \{1, n\}$

a. Encrypt the data partitions using a multiplicative homomorphic encryption scheme, $E(m_1)$, $E(m_2), \ldots, E(m_n)$.

3. $end\ for$
4. Choose a random number $r \mid 1 < r < n$.
5. Calculate the Euclidean distance d between r random pair of encrypted data partitions.
6. $for\ each\ r$ random pair of encrypted data

a. Store the calculated Euclidean distances $d_1, d_{2,\ldots}, \ldots, d_r$ in $< distance\ (d_i), E(m_j), E(m_k) >$ format. Here d_i is the distance between $E(m_j)\ and\ E(m_k)$.

7. $end\ for$
8. Distribute the encrypted data partitions among resource providers.

Algorithm 2: Retrieve the data from resource providers

Step 1: Validate the name

1. Search using the encrypted file name $E(f)$.
2. $for\ all$ possible resource providers R_i

a. $for\ all$ available encrypted file name $E(f_{rpi})$ in the resource provider R_i

i. Calculate the Euclidean distance $D(C)$ between $E(f)$ and $E(f_{rpi})$

ii. $If\ D(C) = 0$

3. Retrieve the data partitions using Step 2.

i. $end\ if$

b. $end\ for$

4. $end\ for$

Step 2: Retrieving the data

1. $for\ each\ r$ random pairs of encrypted data partitions $E(m_i')$ retrieved

c. Calculate the distances, $d_1', d_2', \ldots, d_r'$, between encrypted pair of random data partitions.

d. Compare $if\ d_i = d_i'$

i. Data partition is not tampered.

ii. Decrypt the Data partition with private key y.

e. $end\ if$

2. *end for*

5. Analysis

In this paper we analyze our proposed solution in terms of correctness analysis, privacy analysis and performance analysis.

5.1. Correctness Analysis

In our solution we calculate the Euclidean distance between two encrypted file name E(f) and $E(f_{rpi})$ (line no. 2.a.i of Algorithm 2 Step 1). Since we encrypt the data using multiplicative homomorphic encryption scheme (Eqn.1) we can perform arithmetic operations on those encrypted data. To prove the correctness, the decrypted result of Euclidean distance D(C) (line no. 2.a.ii of Algorithm 2 Step 1) between two encrypted data yields the Euclidean distance between two original data.

$$D(C) = D(E(f) * E(f) + E(f_{rpi}) * E(f_{rpi}) - 2 * E(f) * E(f_{rpi})) = D(E(f - f_{rpi})^2) = (f - f_{rpi})^2$$

The zero Euclidean distance between f and f_{rpi} data denotes that they are same. Similarly to retrieve the data partitions we calculate the Euclidean distances between them.

5.2. Privacy Analysis

In this subsection we explain how we preserve the privacy of data. We distribute the data partitions in encrypted format among different resource providers (Algorithm 1 Step 2 line no. 2.a.i.) and the key pair $(x, y)$is only known to the user. Therefore no other party can learn nothing but $E(m_1), E(m_2), \ldots, E(m_n)$. We calculate the distance between encrypted file names and retrieve the data partitions only when the distance between the encrypted file names is zero. We also calculate the distance between the random pairs of encrypted data partitions to verify the integrity of data. Our proposed scheme preserves privacy and is secure since no third party or adversary can come to know about the original data as we distribute the data partitions in encrypted format, compute operations on encrypted data and key pair is only known to the user.

5.3. Performance Analysis

We implement our algorithm for different sizes of file storage in a simulated distributed cloud environment. To evaluate the performance of our algorithm we vary the size of files and perform privacy preserving homomorphic encryption scheme on the data partitions and file names. To query encrypted file names our algorithm takes 0.2 milliseconds. To store different sizes of files in distributed cloud environment and to retrieve them back using our proposed solution takes 0.2 second for 10KB. Fig. 1 depicts the evaluation times of our algorithm for varying file sizes. From the performance evaluation we can see that the elapsed time increases linearly with increasing file sizes. In order to meet time constraints the partition size can be varied accordingly.

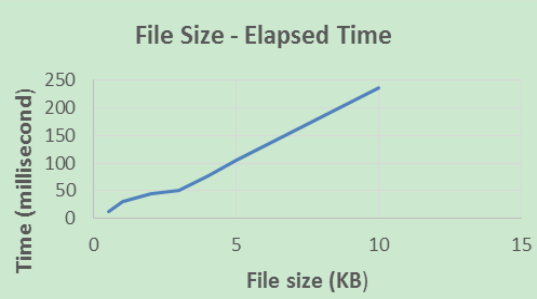

Fig. 1. File size-elapsed time.

6. Conclusion

In this paper we use homomorphic encryption mechanism and its multiplicative property to preserve security and privacy of data in distributed cloud. Our method works on encrypted data without revealing the actual data. We also show that our algorithm is correct and preserves the privacy of data. The performance analysis of our algorithm shows its efficiency and feasibility. This paper not only shows that we can preserve privacy and security of data in distributed cloud, but also distributed cloud is feasible and implementable. Distributed cloud can be used for many applications such as VANETS, distributed medical storage. In future we intend to work on preserving privacy of database records in distributed cloud.

References

[1] Michael, A., Armando, F., Rean, G., Anthony, J., Randy, K., *et al*. (2009). *Above the Clouds: A Berkeley View of Cloud Computing* (Technical Report EECS-2009-28). UC Berkeley.

[2] Peter, M., & Timothy, G. (January 2011). *NIST Definition of Cloud Computing*. National Institute of Standards and Technology.

[3] Buyya, R. & Chee, S. Y., & Venugopal, S. (2008). Market-oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities. *Proceedings of High 10th IEEE International Conference on Performance Computing and Communications*.

[4] Endo, P. T., de Almeida P. A. V., Pereira, N. N., Goncalves, G. E., Sadok, D., Kelner, J., Melander, B., & Mangs, J. *Resource Allocation for Distributed Cloud: Concepts and Research Challenges*.

[5] Babaoglu, O., Moreno, M., & Michele, T. (2012). Design and implementation of a P2P Cloud system. *Proceedings of the 27th Annual ACM Symposium on Applied Computing*.

[6] Khethavath, P., Johnson, T., & Eric, C.-T. (2013). Introducing a distributed cloud architecture with efficient resource discovery and optimal resource allocation. *Proceedings of IEEE Ninth World Congress on Services*.

[7] Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., & Warfield, A. (October 2003). Xen and the art of virtualization, *ACM SIGOPS Operating Systems Review*, *37(5),* 164-177.

[8] Praveen, K., Johnson, T., & Hong, L. (2014). Game theoretic approach to resource provisioning in a distributed cloud. *Proceedings of International Conference on Data Science & Engineering*.

[9] Rivest, R. L., Adi, S., & Len, A. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM, 21(2),* 120-126.

[10] Elgamal, T. (Jul. 1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inform. Theory, 31(4),* 469-472.

[11] Cong ,W., Qian, W., Kui, R., & Lou, W.-J. (March 14-19, 2010). Privacy-preserving public auditing for data storage security in cloud computing. *Proceedings of INFOCOM* (pp. 1, 9).

[12] Liu, Q., Wang, G.-J., & Wu, J. (2009). An efficient privacy preserving keyword search scheme in cloud computing. *Proceedings of International Conference on Computational Science and Engineering*: Vol. 2 (pp. 715, 720).

[13] Maha, T., & Said, E. H. (2013). Secure cloud computing through homomorphic encryption. *International Journal of Advancements in Computing Technology*, *5(16).*

[14] Li, J., Chen, S.-C., Song, D.-J. (2012). Security structure of cloud storage based on homomorphic encryption scheme. *Proceedings of IEEE 2nd International Conference on Cloud Computing and Intelligent*: Vol. 1 (pp. 224, 227).

[15] Pal, D., *et al.* (2014). Designing an algorithm to preserve privacy for medical record linkage with error-prone data. *JMIR Medical Informatics*, *2(1).*

Praveenkumar Khethavath was born in Hyderabad, India. He received his Ph.D. degree from the Department of computer Science at Oklahoma State University in 2014. He is currently an assistant professor in the Mathematics, Engineering and Computer Science Department at LaGuardia Community College. His current research interests include cloud computing, security and privacy, wireless sensor networks and cyber physical systems.

Doyel Pal was born in Kolkata, India. She received her master's degree from University of Calcutta, India. She is currently a Ph.D. student at the Department of Computer Science at Oklahoma State University. Her current research interests include cyber-security and privacy and cloud computing.