International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-2 Number-4 Issue-6 December-2012

Study and Analysis of Data Sharing and Communication with Multiple Cloud Environments

Shikha Joshi[1], Pallavi Jain[2]
M.Tech Scholar, Computer Science, Shri Vaishnav Institute of Technology & Science, Indore[1]
Assistant Professor, Computer Science, Shri Vaishnav Institute of Technology & Science, Indore[2]

Abstract

Cloud storage enables users to access their data anywhere and at any time. It achieves the dream of getting computing, storage and communication resources as easy as to get water and electricity. All resources can be gotten in a plug-and-play way. It has the advantages of high scalability, ease-of-use, cost-effectiveness and simplifying infrastructure planning etc. However, the emerging use of cloud storage has led to the problem of verifying that storage server indeed store the data. When users store their data in cloud storage, they mostly concern about whether the data is intact. In this paper we propose an efficient approach for making cloud data management and sharing in a secure manner. For this we mainly concentrate on six different security majors which are Confidentiality, Availability, Integrity, Possession, Authenticity and Utility. We implement these phenomena in java. We also present a graph comparison to show the betterment of our result.

Keywords

Cloud Computing, Security, Java API, High Scalability

1. Introduction

Cloud computing not to be confused with grid computing, Cloud Computing enables cloud customers to remotely store their data into the cloud so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources [1]. The benefits brought by this new computing model include but are not limited to: relief of the burden for storage management, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc [2].

In recent years, cloud computing is gaining much momentum in the IT industry. Especially, we have seen the dramatic growth of public clouds, in which the computing resources can be accessed by the general public. One of the biggest advantages of a public cloud is its virtually unlimited data storage capabilities and elastic resource provisioning [3]. Many IT enterprises and individuals are outsourcing their databases to the cloud servers, in order to enjoy the much lower data management cost than maintaining their own data centers. It has never been easier than now that a variety of users/clients could access or share information stored in the cloud, independent of their locations.

Because the cloud computing is composed of different local systems and includes the members from multiple environments, therefore the security in cloud is complicate. In one side, the security mechanism should provide guarantees secure enough to the user, on the other side, the security mechanism should not be too complex to put the users into an inconvenient situation. The openness and flexibility of the computer and popular commercial operating systems have been important factors supporting their widespread adoption. However, that very same openness and flexibility have been proved to be a double edged sword, because it brings complexity, reduces trust degree and threat against security. So there should be a balance between the security and the convenience [4]. The dependable and secure computing includes not only security and confidentiality, but also reliability, availability, safety and integrity [5].

Security has been considered as one of the critical concerns that hinder public cloud to be widely used. With the separation of data ownership and storage, a data owner has strong motivation to preserve its control of access and usage of shared data, while leverage storage, computation, and distribution functions provided by cloud, and desire that a public cloud should not learn any clear data. It has been widely recognized that data security should be mainly relied on cloud customers instead of cloud service providers [6,7].

Cloud computing also faces the data security challenges as that of any other communication models. As data owners store their data on external servers, there have been increasing demands and concerns for data confidentiality, authentication and access control [8]. Besides confidentiality and privacy breaks, the external servers could also use part of the data or whole for their financial gain and hence tarnishing the data owners market or even bringing economic losses to the data owner. These concerns originate from the fact that cloud servers are usually operated by commercial providers which are very likely to be outside of the trusted domain of users [9].

The remaining of this paper is organized as follows. In Section 2 we discuss about recent scenario. In section 3 we discuss about proposed work. In section 4 we discuss about the proposed approach. Conclusions are given in Section 5. Finally references are given.

2. Recent Scenario

In 2010, Cong Wang et al. [10] define and solve the problem of effective yet secure ranked keyword search over encrypted cloud data. Ranked search greatly enhances system usability by returning the matching files in a ranked order regarding to certain relevance criteria (e.g., keyword frequency), thus making one step closer towards practical deployment of privacy-preserving data hosting services in Cloud Computing. They give a straightforward yet ideal construction of ranked keyword search under the state-of-the-art searchable symmetric encryption (SSE) security definition, and demonstrate its inefficiency. To achieve more practical performance, we then propose a definition for ranked searchable symmetric encryption, and give an efficient design by properly utilizing the existing cryptographic primitive, order-preserving symmetric encryption (OPSE).

In 2010, Thuy D. Nguyen et al. [11] proposed a Monterey Security Architecture addresses the need to share high-value data across multiple domains of different classification levels while enforcing information flow policies. The architecture allows users with different security authorizations to securely collaborate and exchange information using commodity computers and familiar commercial client software that generally lack the prerequisite assurance and functional security protections. MYSEA seeks to meet two compelling requirements, often assumed to be at odds: enforcing critical, mandatory security policies, and allowing access and collaboration in a familiar work environment. Recent additions to the MYSEA design expand the architecture to support a cloud of cross-domain services, hosted within a federation of multilevel secure (MLS) MYSEA servers. The MYSEA cloud supports single-sign on, service replication, and network-layer quality of security service. This new cross domain, distributed architecture follows the consumption and delivery model for cloud services, while maintaining the federated control model necessary to support and protect cross domain collaboration within the enterprise.

In 2010, Chia-Feng,Lin et al. [12] proposed about Web Services Distributed Management (WSDM) which is one of the industry standards. However, to implement the WSDM interfaces needs to understand server Web service standards. It increases the complexity and difficulty to build the management system. They simplified the Web service management effort between services using hook technology. Our management systems provide message flow oriented management atomically without modifying service code. Enterprise can control all flows and review them at any time.

In 2010, Hong Zhou and Hongji Yang [13] proposed a novel approach to reengineering enterprise software for cloud computing by building ontology for enterprise software and then partitioning the enterprise software ontology to decompose enterprise software into potential service candidates. Ontology development process includes three steps, namely, building ontologies for source code, data, and application framework respectively, integrating captured ontologies and deploying the final produced ontology.

In 2010, G. Hughes et al. [14] proposed about continues, to describe the structure and operation of an object mapping declarative language and the object oriented system which employs it. Both are currently under development to support the management of these numerous Cloud Computing components. The ultimate aim is to develop a system that combines the rich capability of an imperative assembly with the concise simplicity of a declarative language.

In 2010, Xing Chen et al. [15] describe and construct the Internetware Cloud which focus on middleware management and investigate the reusability of the basic management operations and management processes in the MaaS solution.

In 2010, Chia-Feng,Lin et al. [16] analyze the requirements of access protocols for storage systems based on data partitioning schemes in widely distributed cloud environments. They consider the regular semantics instead of atomic semantics to improve access efficiency. Then, we develop an access protocol following the requirements to achieve correct and efficient data accesses. Various protocols are compared experimentally and the results show that our protocol yields much better performance than the existing ones.

In 2011, Mohemed Almorsy et al. [17] introduces a new cloud security management framework based on aligning the FISMA standard to fit with the cloud computing model, enabling cloud providers and consumers to be security certified. Their framework is based on improving collaboration between cloud providers, service providers and service consumers in managing the security of the cloud platform and the hosted services. It is built on top of a number of security standards that assist in automating the security management process. They have developed a proof of concept of our framework using .NET and deployed it on a testbed cloud platform. They evaluated the framework by managing the security of a multitenant SaaS application example.

3. Problem Domain

In this paper we propose a secure way of data sharing and data In this dissertation we mainly concentrate on USER-CLOUD Security. We are mainly concentrating on four problem domain:

1) API and Interfaces are insecure

Cloud providers provide a set of software interfaces or API that customers use to manage and interact with cloud services. Provisioning, management, orchestration, and monitoring are all performed using these interfaces. The security of the normal cloud is all depend on the API used. From authentication and access control to encryption and activity monitoring, these interfaces must be designed to protect against both accidental and malicious attempts to circumvent policy. But in the case of third party control or from different interface cloud the API are the same. This introduces the complexity of the new layered API; it also increases risk, as organizations may be required to relinquish their credentials to third parties in order to enable their agency services.

Examples are Anonymous access and/or reusable tokens or passwords, clear-text authentication or transmission of content, inflexible access controls or improper authorizations, limited monitoring and logging capabilities and unknown service or API dependencies.

2) Malicious Insiders

The threat of a malicious insider is well-known to most organizations. This threat is amplified for consumers of cloud services by the convergence of IT services and customers under a single management domain, combined with a general lack of transparency into provider process and procedure. For example, a provider may not reveal how it grants employees access to physical and virtual assets, how it monitors these employees, or how it analyzes and reports on policy compliance. We consider one more thing in this category, for example we upload the data in the cloud environment and the cloud providers can misuse our data.

3) Data Sharing

IaaS vendors deliver their services in a scalable way by sharing infrastructure. Often, the underlying components that make up this infrastructure, for example CPU caches and GPUs were not designed to offer strong isolation properties for a multi-tenant architecture. To address this gap, a virtualization hypervisor mediates access between guest operating systems and the physical compute resources. Still, even hypervisors have exhibited flaws that have enabled guest operating systems to gain inappropriate levels of control or influence on the underlying platform. A defense in depth strategy is recommended, and should include compute, storage, and network security enforcement and monitoring.

4) Data Loss

There are many ways to compromise data. Deletion or alteration of records without a backup of the original content is an obvious example. Unlinking a record from a larger context may render it unrecoverable, as can storage on unreliable media. Loss of an encoding key may result in effective destruction. Finally, unauthorized parties must be prevented from gaining access to sensitive data. The threat of data compromise increases in the cloud, due to the number of and interactions between risks and challenges which are either unique to cloud, or more dangerous because of the architectural or operational characteristics of the cloud environment.

4. Proposed Approach

In this paper we propose a secure way of data sharing and data integrity in both for the cloud provider and the client. For this we concern on six different security issues which are Confidentiality, Availability, Integrity, Possession, Authenticity and Utility. We consider on those security issues in our approach.

In this approach we can share and inter communicate data in the cloud environment. The data is visualized in encrypted form only. We can see the actual data if you have any decryption key present, otherwise you not see the actual data. We can share the data after a proper sharing key. Admin can read the data of the cloud if the client provides the permission by an authentication read key otherwise admin cannot visualize the actual data of the client.

We apply encryption and Decryption algorithm on those data which is uploaded on the cloud.
Up until recently, the main standard for encrypting data was a symmetric algorithm known as the Data Encryption Standard (DES). However, this has now been replaced by a new standard known as the Advanced Encryption Standard (AES) which we will look at later. DES is a 64 bit block cipher which

means that it encrypts data 64 bits at a time. This is contrasted to a stream cipher in which only one bit at a time (or sometimes small groups of bits such as a byte) is encrypted.

The algorithm is designed to encipher and decipher blocks of data consisting of 64 bits under control of a 64-bit key. Deciphering must be accomplished by using the same key as for enciphering, but with the schedule of addressing the key bits altered so that the deciphering process is the reverse of the enciphering process. A block to be enciphered is subjected to an initial permutation IP, then to a complex key-dependent computation and finally to a permutation which is the inverse of the initial permutation IP-1.
The key-dependent computation can be simply defined in terms of a function f, called the cipher function, and a function KS, called the key schedule. A description of the computation is given first, along with details as to how the algorithm is used for encipherment. Next, the use of the algorithm for decipherment is described. Finally, a definition of the cipher function f is given in terms of primitive functions which are called the selection functions Si and the permutation function P. The following notation is convenient: Given two blocks L and R of bits, LR denotes the block consisting of the bits of L followed by the bits of R. Since concatenation is associative, B1B2...B8, for example, denotes the block consisting of the bits of B1 followed by the bits of B2...followed by the bits of B8.

Blocks are composed of bits numbered from left to right, i.e., the left most bit of a block is bit one. The computation which uses the permuted input block as its input to produce the pre output block consists, but for a final interchange of blocks, of 16 iterations of a calculation that is described below in terms of the cipher function f which operates on two blocks, one of 32 bits and one of 48 bits, and produces a block of 32 bits.

Let the 64 bits of the input block to an iteration consist of a 32 bit block L followed by a 32 bit block R. Using the notation defined in the introduction, the input block is then LR.

Let K be a block of 48 bits chosen from the 64-bit key. Then the output L'R' of iteration with input LR is defined by:
(1) L' = R
R' = L(+)f(R,K)
where (+) denotes bit-by-bit addition modulo 2.

As remarked before, the input of the first iteration of the calculation is the permuted input block. If L'R' is the output of the 16th iteration then R'L' is the preoutput block. At each iteration a different block K of key bits is chosen from the 64-bit key designated by KEY.

With more notation we can describe the iterations of the computation in more detail. Let KS be a function which takes an integer n in the range from 1 to 16 and a 64-bit block KEY as input and yields as output a 48-bit block Kn which is a permuted selection of bits from KEY. That is

(2) Kn = KS(n,KEY)
with Kn determined by the bits in 48 distinct bit positions of KEY. KS is called the key schedule because the block K used in the n'th iteration of (1) is the block Kn determined by (2).

As before, let the permuted input block be LR. Finally, let L() and R() be respectively L and R and let Ln and Rn be respectively L' and R' of (1) when L and R are respectively Ln-1 and Rn-1 and K is Kn; that is, when n is in the range from 1 to 16,
(3) Ln = Rn-1
Rnn = Ln-1(+)f(Rn-1,Kn)
The preoutput block is then R16L16.
The key schedule KS of the algorithm is described in detail in the Appendix. The key schedule produces the 16 Kn which are required for the algorithm.
Deciphering
The permutation IP-1 applied to the preoutput block is the inverse of the initial permutation IP applied to the input. Further, from (1) it follows that:

(4) R = L'
L = R' (+) f(L',K)

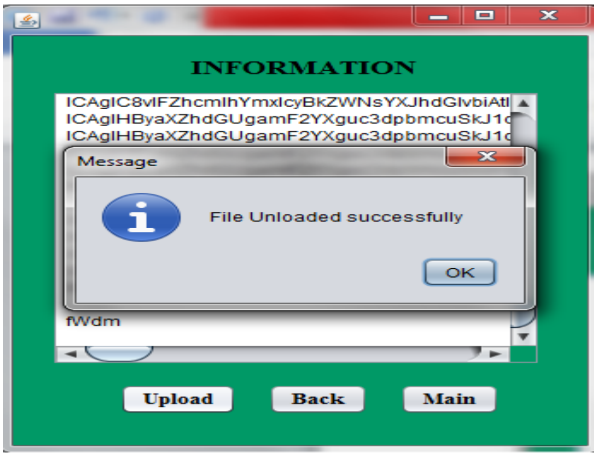
Consequently, to decipher it is only necessary to apply the very same algorithm to an enciphered message block, taking care that at each iteration of the computation the same block of key bits K is used during decipherment as was used during the encipherment of the block. Using the notation of the previous section, this can be expressed by the equations:

(5) Rn-1 = Ln
Ln-1 = Rn (+) f(Ln,Kn)

where now R16L16 is the permuted input block for the deciphering calculation and L() and R() is the preoutput block. That is, for the decipherment calculation with R16L16 as the permuted input, K16 is used in the first iteration, K15 in the second, and so on, with K1 used in the 16th iteration.

When our data, business process, applications are deployed to Cloud, how secure are our data, business process & applications going to be. This is one of the top most questions by the customers. What are

security solutions that are provided by cloud service provider, what are the security solutions that can be built into products, Business applications, and Enterprise applications by different IT vendors?
In our approach we are somehow able to answer the questions. The results rae shown in Figure 1, Figure2, Figure 3 and Figure 4.



```
ICAgIC8vIFZhcmlhYmxlcyBkZWNsYXJhdGlvbiAtI
ICAgIHByaXZhdGUgamF2YXguc3dpbmcuSkJ1c
ICAgIHByaXZhdGUgamF2YXguc3dpbmcuSkJ1c
fWdm
```

File Unloaded successfully

Figure 1: File uploads with Encryption

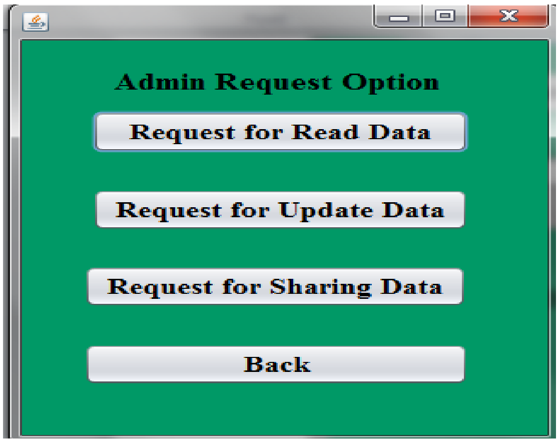
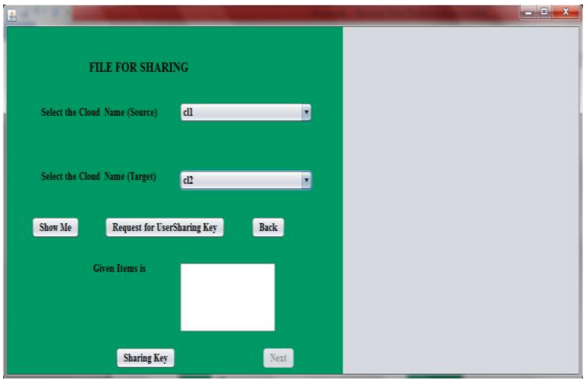


Figure 2: Admin Request Option

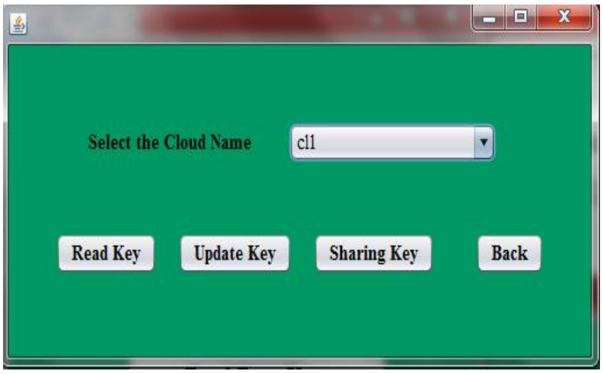


Select the Cloud Name (Source): cl1

Select the Cloud Name (Target): cl2

Given Items is

Figure 3: Request For Sharing



Select the Cloud Name: cl1

Figure 4: Key Request

5. Conclusion

The increased degree of connectivity and the increasing amount of data has led many providers and in particular data centers to employ larger infrastructures with dynamic load and access balancing. This lead to the demand of cloud computing. But there are some security concerns when we handle and share data in the cloud computing environment. In this dissertation we propose a new cloud computing environment where we approach a trusted cloud environment which is controlled by both the client and the cloud environment. Our approach is mainly divided into two parts. First part is controlled by the normal user which gets permission by the cloud environment for performing operation and for loading data. Second part shows a secure trusted computing for the cloud, if the admin of the cloud want to see the data then it take permission from the client environment. This provides a way to hide the data and normal user can protect their data from the cloud provider. This provides a two way security which helps both the cloud and the normal user. For the above concept we propose a java based algorithm. Cloud providers have begun offering users at-cost access to on demand computing infrastructures. We also discuss about a Cloud-based cooperative cache system for reducing execution times of data-intensive processes. The benefits of deploying applications using cloud computing include reducing run time and response time, minimizing the risk of deploying physical infrastructure, lowering the cost of entry, and increasing the pace of innovation. In this paper we discuss few aspects of cloud computing and also there area. We also propose a novel approach which is cloud computing mapping and management through class and object hierarchy. In this approach we first design a cloud environment where we can analyze several object oriented aspects based on some assumptions. Then we deduce message passing behavior through a backup files based on the properties of object orient like class and object. We also provide better security approaches in terms of the previous methodology.

References

[1] P. Mell and T. Grance, “Draft nist working definition of cloud computing,” Referenced on Jan. 23rd, 2010 Online at http://csrc.nist.gov/groups/SNS/cloud-computing/index.html, 2010.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “Above the clouds: A berkeley view of cloud computing,” University of California, Berkeley, Tech. Rep. UCB-EECS-2009-28, Feb 2009.

[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. Above the clouds: A berkeley view of cloud computing, Feb 2009.

[4] Jason Reid Juan M. González Nieto Ed Dawson, "Privacy and Trusted Computing", Proceedings of the 14th International Workshop on Database and Expert Systems Applications, IEEE, 2003.

[5] Algirds Avizienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr, “Basic Concepts and Taxonomy of Dependable and Secure Computing”, IEEE transactions on dependable and secure computing, vol.1, No.1, January-March, 2004.

[6] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing", Journal of Network and Computer Applications, 34(1), 2011, pp 1-11.

[7] S. Kamara and K. Lauter, "Cryptographic cloud storage", FC'10: Proc. 14thIntl.Conf. on Financial cryptography and data security,2010, pp. 136-149.

[8] S. D. C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, A Data Outsourcing Architecture Combining Cryptography and Access Control, Proc. ACM Workshop on Computer Security Architecture (CSAW’07), Nov 2007, USA.

[9] S. Yu, C. Wang, K. Ren, and W. Lou, Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing, Proc. IEEE INFOCOM 2010, San Diego, CA, pp. 1-9.

[10] Cong Wang,, Ning Cao,, Jin Li,, Kui Ren, and Wenjing Lou , “Secure Ranked Keyword Search over Encrypted Cloud Data” ICDCS 2010, IEEE.

[11] Thuy D. Nguyen, Mark A. Gondree, David J. Shifflett, Jean Khosalim, Timothy E. Levin, Cynthia E. Irvine , “A Cloud-Oriented Cross-Domain Security Architecture”, The 2010 Military Communications Conference - Unclassified Program - Cyber Security and Network Management.

[12] Chia-Feng,Lin, Ruey-Shyang Wu, Shyan-Ming Yuan , Ching-Tsorng Tsai,” A Web Services Status Monitoring Technology for Distributed System Management in the Cloud”, 2010 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery.

[13] Hong Zhou, Andrew Hugill and Hongji Yang, “An Ontology-Based Approach to Reengineering Enterprise Software for Cloud Computing”, 2010 IEEE 34th Annual Computer Software and Applications Conference.

[14] G. Hughes, D. Al-Jumeily & A. Hussain,” Supporting Cloud Computing Management through an Object Mapping Declarative Language” , 2010 Developments in E-systems Engineering.

[15] Xing Chen, Xuanzhe Liu, Fuzhi Fang, Xiaodong Zhang, Gang Huang,” Management as a Service: An Empirical Case Study in the Internetware Cloud”, IEEE International Conference on E-Business Engineering.

[16] Yunqi Ye, Liangliang Xiao, I-Ling Yen, Farokh Bastani, “Secure, Dependable, and High Performance Cloud Storage”, 2010 29th IEEE International Symposium on Reliable Distributed Systems.

[17] Mohemed Almorsy, John Grundy and Amani S. Ibrahim, “Collaboration-Based Cloud Computing Security Management Framework”, 2011 IEEE 4th International Conference on Cloud Computing.