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

A Secure Data Sharing and Communication with Multiple Cloud Environments with Java API

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.

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.

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 [4, 5].

A typical approach for data confidentiality protection is to encrypt a data with a (usually symmetric) key before storing it to cloud. However, encryption makes it difficult to flexibly sharing data between different users. On one side, sharing the data encryption key to all users easily enables a user to access all data that stored in cloud of a data owner, which violates the least privilege principle. On the other side, any change of access control policy either demands decryption and re-encryption of the data in cloud, which exposes clear data in cloud, or the owner has to re-encrypt the data and re upload to cloud, which brings computation and bandwidth cost to the owner. Furthermore, collusion between a legitimate user and the cloud easily allows unauthorized data sharing and distribution.

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 [6]. 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 [7].

The remaining of this paper is organized as follows. We discuss Cloud Computing in Section 2. In Section 3 we discuss about security model. In section 4 we discuss about Recent Scenario. In section 5 we discuss about the proposed approach. Conclusions are given in Section 6. Finally references are given.

2. Cloud Computing

Cloud computing services are provided either by subscription-based or pay-per-use, in real time over the Internet. But it is always in mind that how we can use it for our needs.

Imagine being an executive of a company, with the responsibilities to ensure all your employees have the right hardware and software for their jobs. Typical company setting would be one computer to one head count. However, software licenses may not, due to costing. Therefore, you may have to be selective in your purchase decision.

Instead of installing a suite of software on each computer, you can make use of Cloud Computing service. All you have to do, is to run a cloud computing systems interface software. This application allows employees to log into an Internet-based service which hosts all the programs a user need for his job. A simple cloud computing systems interface software is a Web browser.

Remote machines owned by the service provider would run everything from e-mail to word processing to complex data analysis programs.

Think Hotmail, Yahoo! Mail or Gmail; perhaps you may already have some hand-on experience with cloud computing after all. Instead of running an e-mail program on your computer, you are logging onto a Web email account “remotely”. The software and storage for your account doesn't exist on your computer - it's on the service provider's "computer cloud".

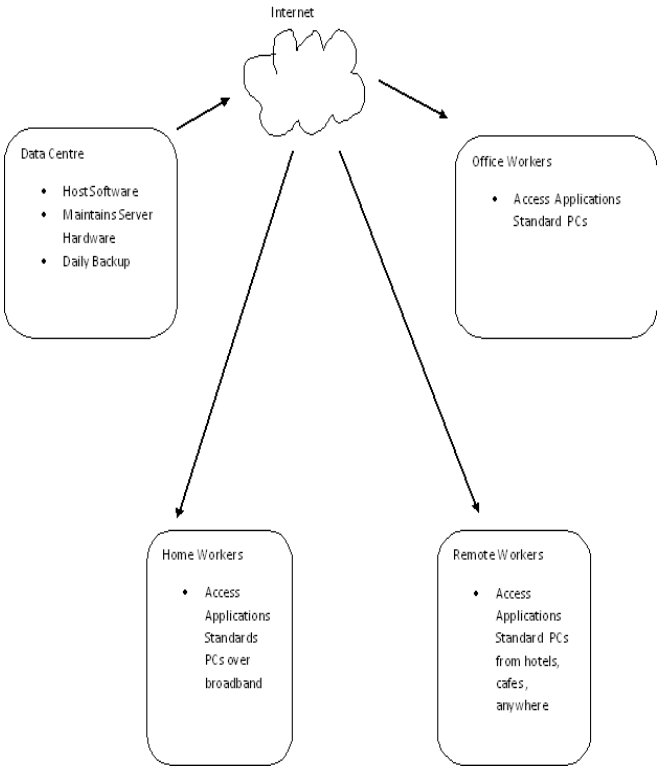

Figure 1: Computation

3. Security Model

The Jericho Forum proposed an interesting approach to cloud computing security. Starting with a description of Cloud Layers below, allows us to envision the problem:

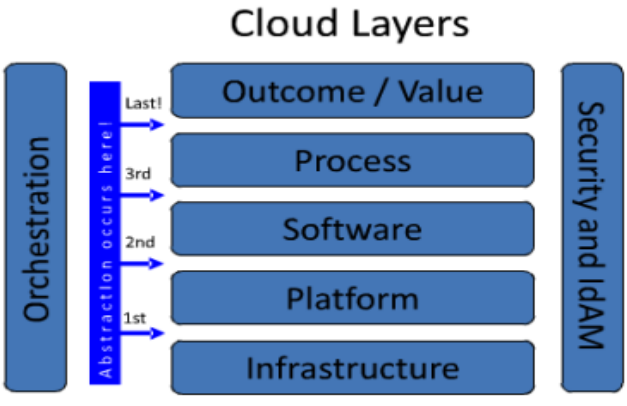

Figure 2: Cloud Security Layer (Source: Jericho Forum)

Here, the Forum proposed that Security (and Identity Management) are elements that cross all layers and in effect provide a design they call Collaboration Oriented Architecture (COA). Once this foundation has been laid, they defined Cloud Security as a proposed a cube-shaped model that highlights various possibilities of architecture, the one addressed here is, of course, the outsourced / external / De-Parameterized option.

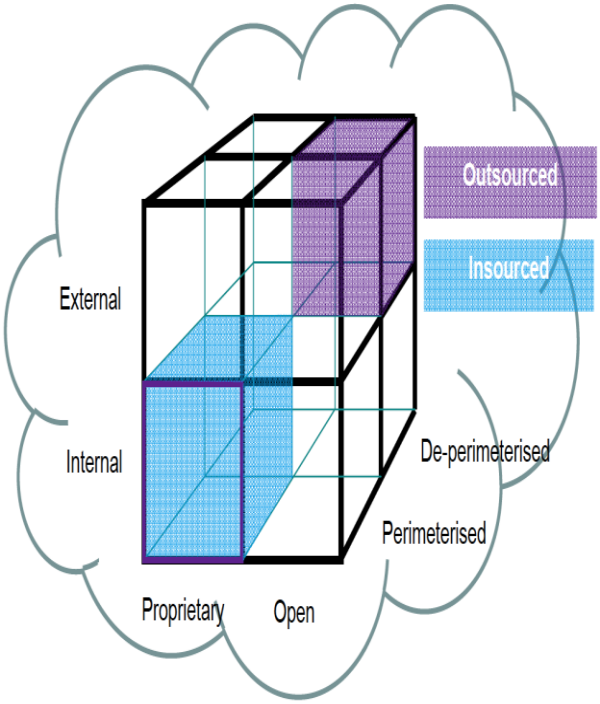

Figure 3: Cloud Security Model

Then we can break down Cloud computing into three delivery types:

1. Infrastructure as a Service (IaaS)
2. Platform as a Service (PaaS)
3. Software as a Service (SaaS)

And then proceeded to define the Cloud consumption models:

1. Private
2. Public
3. Managed
4. Hybrid

If we arrange those elements in a matrix, we will get a cube similar to figure 4.

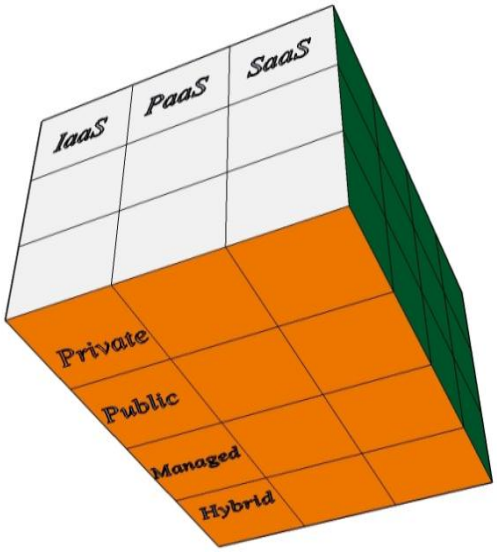

Figure 4: Cube Model

So, allow me here to define the problem statement a bit differently than above. Because these are the early days of any cloud discussion, and that translates usually into this time being the formative years, let's expand the basic three tenets of security, which are:

1. Confidentiality
2. Availability and
3. Integrity

And add additional controls. We will borrow from Donn Parker's Hexad, and add:

4. Possession (or Control)
5. Authenticity and
6. Utility

Clearly, in the case of Cloud computing, and especially in the Public/External case, we no longer have any control. Once the bits "leave our network," control passes elsewhere.

Losing one control typically mandates an increase in the other controls. Here, we have another set of problems. Let us explore the remaining controls:

Confidentiality: Confidentiality refers to limiting information access and disclosure to authorized users "the right people" and preventing access by or disclosure to unauthorized ones "the wrong people." It has to be sent, or assembled, in the Cloud, remain there in an encrypted form, and be transferred to us, for processing. Once the data is at our location, we have to decrypt it, perform the

operations needed, then re-encrypt and resend to the Cloud.

Availability: Availability refers, unsurprisingly, to the availability of information resources. An information system that is not available when you need it is almost as bad as none at all. It may be much worse, depending on how reliant the organization has become on a functioning computer and communications infrastructure. Availability, like other aspects of security, may be affected by purely technical issues (e.g., a malfunctioning part of a computer or communications device), natural phenomena (e.g., wind or water), or human causes (accidental or deliberate).

Integrity: Integrity refers to the trustworthiness of information resources. It includes the concept of "data integrity" -namely, that data have not been changed inappropriately, whether by accident or deliberately malign activity. It also includes "origin" or "source integrity" that is, that the data actually came from the person or entity you think it did, rather than an imposter.

Possession: Remote data possession checking is that focuses on how to frequently, efficiently and securely verify that a storage server can faithfully store its client's (potentially very large) original data without retrieving it.

Authenticity: The quality or condition of being **authentic**, trustworthy, or genuine.

Utility: Utility is where Cloud Computing excels. If we can figure out the other five elements, we can be the Bruce Willis of this story. Cloud Computing is clearly an idea whose time is near. Companies will put data in the Cloud; use the Cloud; and expand the Cloud in a tremendously accelerating rate, regardless of data security and privacy.

4. Recent Scenario

In 2010, Cong Wang et al. [8] 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. [9] 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. [10] 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 [11] 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. [12] 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. [13] 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. [14] 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. [15] 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 exemplar.

5. 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. Our approach is shown in Figure 5.

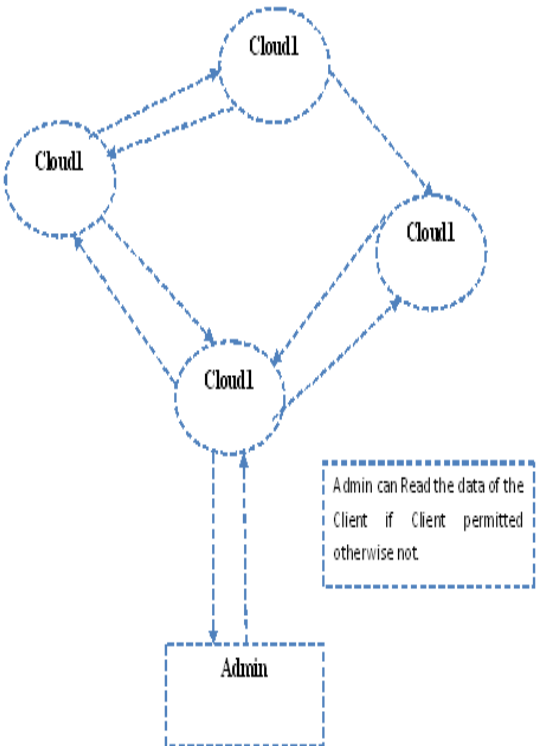

Figure 5: Our Approach

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.
Our algorithm for different part of security is given below:
1) Authentication

```
Login()
  {
  label1 = new JLabel();
  label1.setText("Username:");
  text1 = new JTextField(15);

  label2 = new JLabel();
  label2.setText("Password:");
  text2 = new JPasswordField(15);
```

```
}
2) Encryption and Decryption with Data Encryption Standard (DES) in Java
public class DesEncrypter {
   Cipher ecipher;
   Cipher dcipher;

   DesEncrypter(SecretKey key) {
      try {
         ecipher = Cipher.getInstance("DES");
         dcipher = Cipher.getInstance("DES");
         ecipher.init(Cipher.ENCRYPT_MODE, key);
         dcipher.init(Cipher.DECRYPT_MODE, key);

      } catch (javax.crypto.NoSuchPaddingException e) {
      } catch (java.security.NoSuchAlgorithmException e) {
      } catch (java.security.InvalidKeyException e) {
      }
   }

   public String encrypt(String str) {
      try {
         // Encode the string into bytes using utf-8
         byte[] utf8 = str.getBytes("UTF8");

         // Encrypt
         byte[] enc = ecipher.doFinal(utf8);

         // Encode bytes to base64 to get a string
         return new sun.misc.BASE64Encoder().encode(enc);
      } catch (javax.crypto.BadPaddingException e) {
      } catch (IllegalBlockSizeException e) {
      } catch (UnsupportedEncodingException e) {
      } catch (java.io.IOException e) {
      }
      return null;
   }

   public String decrypt(String str) {
      try {
         // Decode base64 to get bytes
         byte[] dec = new sun.misc.BASE64Decoder().decodeBuffer(str);

         // Decrypt
         byte[] utf8 = dcipher.doFinal(dec);

         // Decode using utf-8
         return new String(utf8, "UTF8");
      } catch (javax.crypto.BadPaddingException e) {
      } catch (IllegalBlockSizeException e) {
      } catch (UnsupportedEncodingException e) {
      } catch (java.io.IOException e) {
      }
      return null;
   }
}
```

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 S_i 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, $B_1B_2...B_8$, for example, denotes the block consisting of the bits of B_1 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.

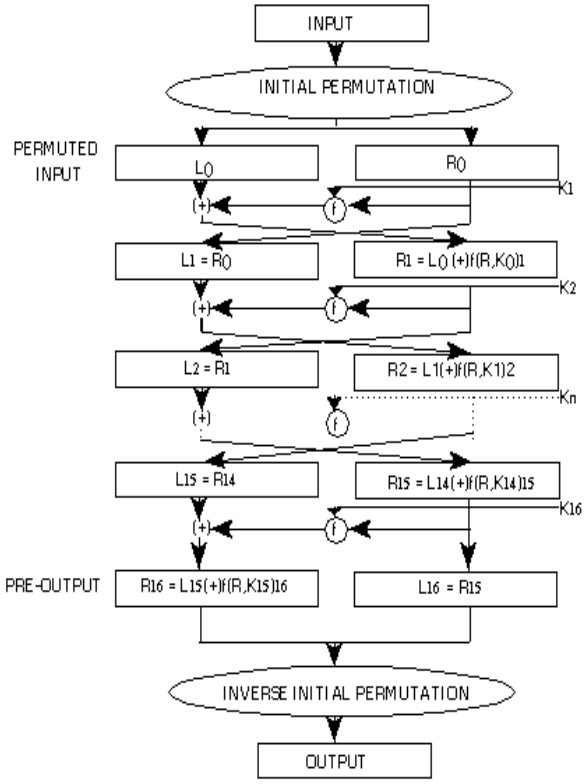

Figure 6: Enciphering computation

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 notations 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 K_n which is a permuted selection of bits from KEY. That is
(2) $K_n = KS(n,KEY)$
with K_n 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 K_n determined by (2).
As before, let the permuted input block be LR. Finally, let L_0 and R_0 be respectively L and R and let Ln and Rn be respectively L' and R' of (1) when L and R are respectively L_{n-1} and R_{n-1} and K is K_n; that is, when *n* is in the range from 1 to 16,
(3) $L_n = R_{n-1}$
$R_n n = L_{n-1}(+)f(R_{n-1},K_n)$
The preoutput block is then $R_{16}L_{16}$.
The key schedule KS of the algorithm is described in detail in the Appendix. The key schedule produces the 16 K_n 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) $R_{n-1} = L_n$
$L_{n-1} = R_n$ (+) $f(L_n,K_n)$
Where now $R_{16}L_{16}$ is the permuted input block for the deciphering calculation and L_0 and R_0 is the preoutput block. That is, for the decipherment calculation with $R_{16}L_{16}$ as the permuted input, K_{16} is used in the first iteration, K_{15} in the second, and so on, with K_1 used in the 16th iteration.
The Cipher Function *f*

A sketch of the calculation of *f*(R,K) is given in Figure 7.

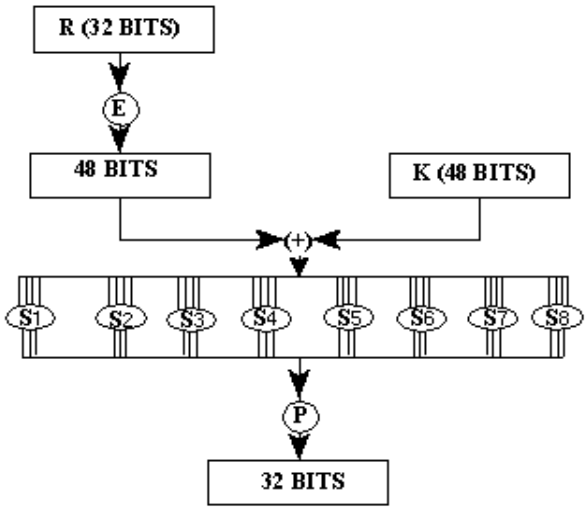

Figure 7: Calculation of *f*(R, K)

The above algorithm is used for data encryption and decryption.

6. Conclusion

We discuss several aspects of cloud computing including the advantages and disadvantages. We also analyze several asymptotic behavior of cloud computing according with several analyses. 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/cloudomputing/index.html.

[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] AWS Customer Agreement http://aws.amazon.com/agreement/, 2011.

[5] C. Security Alliance. Security Guidance for Critical Areas of Focus in Cloud Computing V2.1, 2009.

[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, 2010.

[16] Yunqi Ye, Liangliang Xiao, I-Ling Yen, Farokh Bastani, "Secure, ependable, 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.

Bachelor of Engineering from Acropolis Institute of Technology & Research, Indore (M.P) in Computer Science with first division Pursuing ME (computer science) from Shri Vaishnav institute of Technology and science Indore.