A Comprehensive Approach to Ensure Secure Data Communication in Cloud Environment

M. Sudha
Assistant Professor (Senior)
School of Information Technology & Engineering
VIT University, India

Dr.Bandaru Rama Krishna Rao
Senior Professor
School of Information Technology & Engineering
VIT University, India

M. Monica
Assistant Professor
School of Computing Sciences & Engineering
VIT University, India

ABSTRACT
Cloud computing is an Internet based development, in concept, it is a paradigm shift whereby details are abstracted from the users who no longer need knowledge of, expertise in, or control over the technology infrastructure that supports them. According to a 2008 IEEE paper, "Cloud Computing is a paradigm in which information is permanently stored in servers on the internet and cached temporarily on clients that include desktops, entertainment centers, table computers and notebooks etc." Some examples of emerging Cloud computing infrastructures are Microsoft Azure, Amazon EC2, Google App Engine, and Aneka. Cloud service providers enable users to access and use the necessary resources via the internet. To provide these resources, providers often fall back upon other providers in the cloud, hence this raises security issues in Cloud Environment as Clouds have no borders and the data can be physically located anywhere in the world. So this phenomenon raises serious issues regarding user authentication and data confidentiality. Hence it is proposed to implement a simple Data Protection framework which performs authentication, verification and encrypted data transfer, thus maintaining data confidentiality. Programming is performed using JAVA platform, Cloud environment is created using wired and wireless LAN networks. And Advanced Encryption Standard security algorithm is implemented for ensuring security framework.

Keywords: Cloud computing, Data security, Internet, Advanced Encryption standard.

1. INTRODUCTION
Cloud computing is a computing paradigm in which tasks are assigned to a combination of connections, software and services accessed over a network. This network of servers and connections is collectively known as "the cloud." Computing at the scale of the cloud allows users to access supercomputer-level power. Using a thin client or other access point, like an iPhone, BlackBerry or laptop, users can reach in to the cloud for resources as they need them. For this reason, cloud computing has also been described as "on-demand computing." This vast power is made possible though distributed, large-scale cluster computing, often in concert with server virtualization software, like Xen, and parallel processing. Cloud computing can be contrasted with the traditional desktop computing model, where the resources of a single desktop computer are used to complete tasks, and an expansion of the client/server model. To paraphrase Sun Microsystems' famous adage, in cloud computing the network becomes the supercomputer.

The coming shift to cloud computing is a major change in the industry. One of the most important parts of that shift is the advent of cloud platforms. As its name suggests, this kind of platform lets developers write applications that run in the cloud, or use services provided from the cloud, or both. Different names are used for this kind of platform today, including on-demand platform and platform as a service (PaaS). Whatever it's called, this new way of supporting applications has great potential. Problem statement is clearly defined in Section II followed by related works in section III. Section IV describes the systems requirements and the modules and the framework implementation is explained in Section V.

2. PROBLEM STATEMENT
Data protection is a critical issue in cloud computing environments. Clouds have no borders and the data can be physically located anywhere in the world. So this phenomenon raises serious issues regarding user authentication and data confidentiality. Hence it is proposed to implement a simple Data Protection Package which performs authentication, verification and encrypted data transfer, thus maintaining data confidentiality.

3. RELATED WORKS
Cong Wang et al. [1] stated that data security is a problem in cloud data storage, which is essentially a distributed storage system. And explained their proposed scheme to ensure the correctness of user's data in cloud data storage, an effective and flexible distributed scheme with explicit dynamic data support, including block update, delete, and append relying on erasure-correcting code in the file distribution preparation to provide redundancy parity vectors and guarantee the data dependability. Their scheme could achieve the integration of storage correctness insurance and data error localization, i.e., whenever data corruption has been detected during the storage correctness verification across the distributed servers, Could almost guarantee the simultaneous identification of the misbehaving server(s) through detailed security and performance analysis.

John Harauz et al. [3], described the Security Content automation protocol (SCAP) and benefits it can provide with latest cloud computing paradigm with reference to the latest report released by NIST, giving insight as to what SCAP is trying to do, It states that many tools for system security, such as patch management and vulnerability management software, use proprietary formats,

nomenclatures, measurements, terminology, and content. Their example states that, when vulnerability scanners do not use standardized names for vulnerabilities, it might not be clear to security staff whether multiple scanners are referencing the same vulnerabilities in their reports.

Siani Pearson, et al. [4], described the overview of privacy issues within cloud computing and a detailed analysis on privacy threat based on different type of cloud scenario was explained, the level of threat seem to vary according to the application area. Their work has stated the basic guidelines for software engineers when designing cloud services in particular to ensure that privacy are not mitigated.

Meiko Jensen et al. [5], described a selection of issues of Cloud Computing security and the Web Services security frameworks (attacking the Cloud Computing system itself), stating the importance and capabilities of browser security in the Cloud computing context, and sketched the threat of flooding attacks on Cloud systems. Showed, the threats to Cloud Computing security are numerous, and each of them requires an in-depth analysis on their potential impact and relevance to real-world Cloud Computing scenarios. It is well understood that from their investigation, a first good starting point for improving Cloud Computing security consists in strengthening the security capabilities of both Web browsers and Web Service frameworks.

Balachandra Reddy Kandukuri et.al [6], described some of the security issues that have to be included in Service Level Agreement (SLA), SLA is a document which defines the relationship between service provider and the recipient, typical Service level agreement contents includes Definition of services, Performance management, Problem Management, Security, Disaster recovery, proper termination of transaction also they have stated a methodology to standardize SLA's.

4. REQUIREMENTS SPECIFICATION

4.1 Hardware Requirements

The system running the application should have following minimum requirements:

1. Pentium Core
2. RAM Size 128mb
3. Processor 1.2GHz

4.2 Software Requirements

The system running the application must have the following:

1. Supporting OS: Windows XP, VISTA,LINUX: Red Hat, Ubuntu, Fedora
2. Java Development Kit - jdk1.6.0_02.
3. Java Runtime Environment - jre1.6.0_06.
4. Web Browser with Java Plug-in installed
5. Wireless connectivity driver.

4.3 Network Support

This project can be used to run on both Wireless and LAN networks. It supports following network architectures:

1. Local Area Network (using network cables)
2. Wireless Network (Wi-Fi)
3. Ad-Hoc Network
4. Dial-up or VPN network to a workplace.

4.4 Technology Specific Tools used

In this work we use following tools:

1. Java Development Kit - jdk1.6.0_02.
2. Java Runtime Environment - jre1.6.0_06.
3. Java.awt package for layout of the applet.
4. Java.net package for connection settings and message passing.
5. Socket Options interface of methods to get/set socket options.

5. IMPLEMENTATION

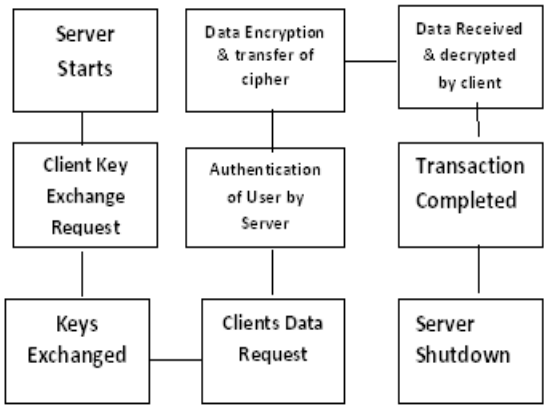


Figure.1 Design Framework – The High Level Design

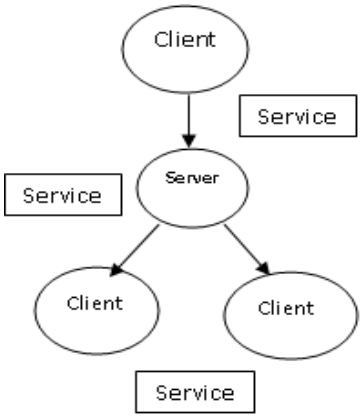


Figure 2 Context Diagram – The Low level Design

Client requests service to server and Server grants the request through a response. Client has to first register himself in the server to begin with. Server stores the password and client is registered. Then at the time of login, Username and password are verified, welcome message is prompted. Client sends a request to server; server creates a key and forwards it to the corresponding client for its use in encryption and or decryption. Client request for data which is encrypted by the server ant then sent to it. Client later decrypts it to obtain the plaintext.

5.1 Module 1: Study of the System

This proposed system can be mainly divided into two parts:

- Server
- Client

This project is mainly depended on client/server model. The client requests the server and server responses by granting the clients request. The proposed system should provide both of the above features along with the followed ones:

a) Server - The server should be able to perform the following features:

- The first and foremost problem is to find the server. We should identify the program in the server which processes the client's request.
- authentication of users
- generation of keys
- encryption of data files

b) Client: The client should be able to perform the following features:

- authenticate itself from server
- request for keys
- decrypt data files

5.2 Module 2: Authentication Of Client

This module involves testing the users for their authenticity by carrying out username and password verifications. There maybe two type of clients logging on to the server:

1. New Users
2. Existing Users

New Users shall give a required username and password which will be added to the database on the server side. Existing users shall verify their identity by providing their unique username and password. Once authenticated they can run the next module which provides the keys to clients.

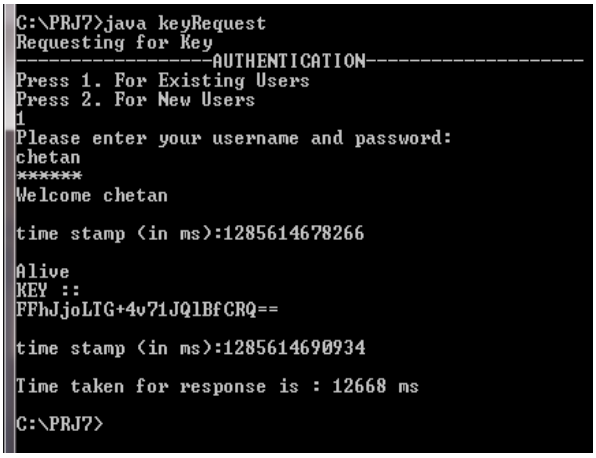


Figure 3.Authentication at key request

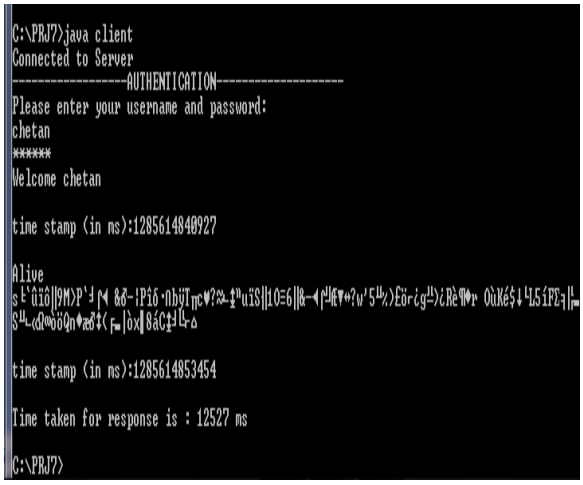


Figure 4. Authentication at data request

5.3 Module 3: key generation

This module handles key generation by the server side. The server generates unique keys for users once they authenticate themselves with the server. The key is generated using instances of AES key generator class. This key is then transferred to the client via the LAN connection which receives and stores a copy for it for decrypting purpose. The key is a 16 byte or a 128 bit key. An Example of a key generated here is:
8xRER4LyFiU3Hs9a40xExQ==

5.4 Module 4: Encryption of Data Files By Server

Once the keys are exchanged, the client requests for a data file to be transferred to itself. The server then encrypts the data file with AES algorithm explained below and sends the cipher text to the client. In cryptography, the Advanced Encryption Standard (AES) is a symmetric-key encryption standard. Each of these ciphers has a 128-bit block size, with key sizes of 128, 192 and 256 bits, respectively. The AES ciphers have been analyzed extensively and are now used worldwide, as was the case with its predecessor, the Data Encryption Standard (DES). The AES cipher is specified as a number of repetitions of transformation rounds that convert the input plaintext into the final output of cipher text. Each round consists of several processing steps, including one that depends on the encryption key. A set of reverse rounds are applied to transform cipher text back into the original plaintext using the same encryption key.

5.4.1 High-Level Description of the Algorithm

a) Key Expansion—round keys are derived from the cipher key using Rijndael's key schedule
Initial Round
AddRoundKey—each byte of the state is combined with the round key using bitwise xor
Rounds
SubBytes - a non-linear substitution step where each byte is replaced with another according to a lookup table.
ShiftRows - a transposition step where each row of the state is shifted cyclically a certain number of steps.

MixColumns - a mixing operation which operates on the columns of the state, combining the four bytes in each column.

1. AddRoundKey

Final Round (no MixColumns)

1. SubBytes
2. ShiftRows
3. AddRoundKey

In the SubBytes step, each byte in the array is updated using an 8-bit substitution box, the Rijndael S-box. This operation provides the non-linearity in the cipher.

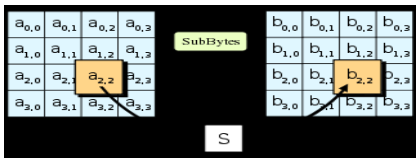


Figure 5. Sub Bytes

The Shift Rows step operates on the rows of the state; it cyclically shifts the bytes in each row by a certain offset. For AES, the first row is left unchanged. Each byte of the second row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of two and three respectively

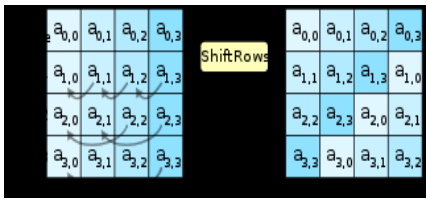
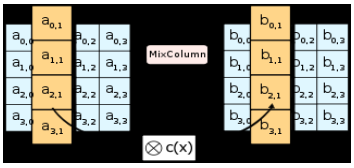


Figure 6. Shift Rows

In the Mix Columns step, the four bytes of each column of the state are combined using an invertible linear transformation. The Mix Columns function takes four bytes as input and outputs four bytes, where each input byte affects all four output bytes. Together with ShiftRows, MixColumns provides diffusion in the cipher. During this operation, each column is multiplied by the known matrix that for the 128 bit key is



$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}.$$

Figure 7. Mix Columns

In the AddRoundKey step, the subkey is combined with the state. For each round, a subkey is derived from the main key using Rijndael's key schedule each subkey is the same size as the state. The subkey is added by combining each byte of the state with the corresponding byte of the subkey using bitwise XOR. AES (Rijndael) uses a key schedule to expand a short key into a number of separate round keys. This is known as the Rijndael key schedule**.**

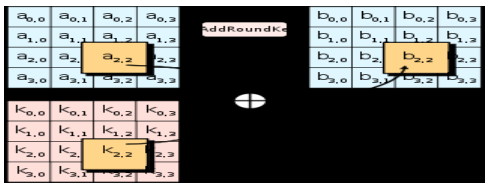


Figure 8. Round Key

5.5 Module 5: Decryption of Cipher Text By Client

After the keys are received and the cipher text is sent to the client by the user, the client uses the reverse process of the AES encryption .AES decryption to obtain the original plaintext that was transferred by the server.Hence the client receives the intended file in a secure manner over the LAN.

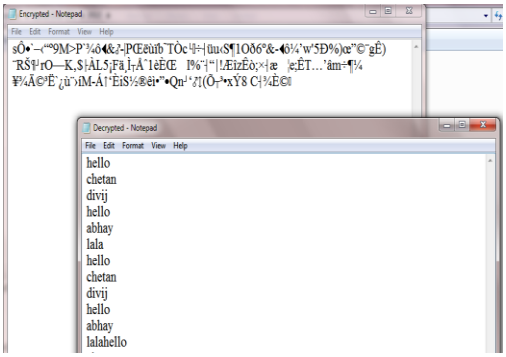
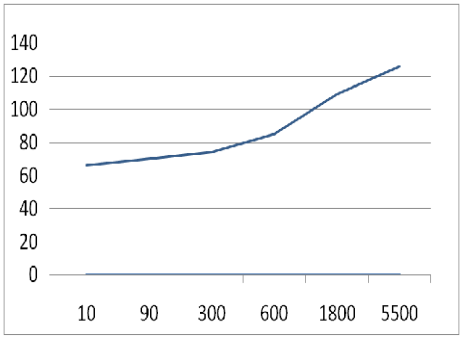


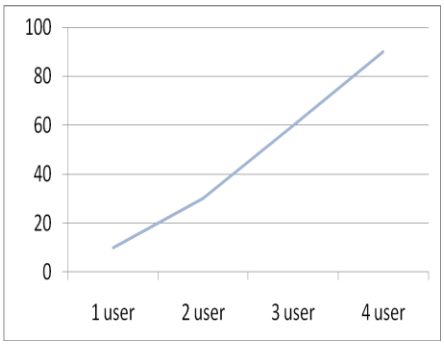
Figure 9.Snapshot of the encrypted and decrypted files

5.6 Module 6: Analysis of Client Server Interaction



X axis : bytes of data
Y axis : Processing time

Figure 10. processing time v/s the size of data exchanged



X axis : no . of users
Y axis : Response time
Figure 11. Plot of Response time v/s No. of users

To perform the ***Uptime Monitoring*** using a secure user program. This module involves the basic interaction between a Cloud Client and a Client Server.Uptime Monitoring hence performed measures the time taken for the processing of a request from a client by the server.We then plot the size of data transferred against time taken for processing and other plots related to uptime monitoring.

6. FUTURE WORK

1. Ahead of this we plan to add one more dimension to the security framework of this security suite i.e. to verify the signatures of the keys and files that are being transferred to ensure complete security from alteration.

2. Time monitoring of the whole process to ensure it's feasible in real-time environment of a network.

7. REFERENCES

[1] cong Wang, Qian Wang and Kui Ren. "Ensuring Data Storage Security in Cloud computing" 978-1-4244-3876-1/2009 IEEE.

[2] Lijun Mei, W.K.Chan and T.H.T se, "A Tale of Clouds: Paradigm comparisons and some thoughts on research issues", 2008 IEEE Asia-Pacific Services Computing Conference.

[3] John Harauz, Lori M. Kaufman and Bruce Potter, "Data security in the world of cloud computing ",2009 IEEE CO Published by the IEEE Computer and Reliability Societies.

[4] Siani Pearson, "Taking account of Privacy when Designing Cloud computing Services" *CLOUD'09*, May 23, 2009, Vancouver, Canada, 2009 IEEE.

[5] Meiko Jensen, Jorg Schwenk, Nils Gruschka and Luigi Lo Iacono, "On technical security issues in cloud computing "978-0-7695- 3840-2/ 092009,IEEE Computer Society.

[6] Balachandra Reddy Kandukuri, Ramakrishna Paturi V and Dr. Atanu Rakshit, "Cloud securityIssues" 978-0-7695-3811-2/09 2009, IEEE computer society.

[7] Guy Bunker, Farnam Jahanian, Aad van Moorsel and Joseph Weinman," Dependability in the cloud: Challenges and opportunities", IEEE 2009.

[8] Lizhe Wang, Jie Tao, Marcel Kunze , Alvaro Canales Castellanos, David Kramer and Wolfgang Karl, "scientific Cloud computing: early Definition and Experience", 2008 IEEE.

[9] Rajkumar Buyya1, Rajiv Ranjan2 and Rodrigo N. Calheiros, "Modeling and simulation of scalable cloud computing Environments and the CloudSim Toolkit: challenges and opportunities" 978-1-4244-4907-1/09, 2009 IEEE

[10] Ivona Brandic,"Towards Self manageable Cloud services" 0730-3157/09 , 2009 IEEE.

[11] http://www.cloudsecurity.org, accessed on April 10, 2009.

[12] D.J. Solove, "A Taxonomy of Privacy", University of Pennyslavania Law Review, vol 154, no 3, January 2006, p.477. http://papers.ssrn.com/.

[13] Cloud Security Alliance. http://www.cloudsecurityalliance.org.

[14] T. R. Peltier, J. Peltier, and J. Blackley, "*Information Security Fundamentals"*. Auerbach Publications, Boston, MA, USA, 2003.