Review on Data Storage Security in Cloud Computing

Prof. Swarnalata Bollavarapu
Asst Prof, Dept of Computer Engineering
MPSTME, NMIMS
Mumbai, India

Nikhil Kamath
Student, Dept of Computer Engineering
MPSTME, NMIMS
Mumbai, India

Abstract

Cloud computing is a computing technique, where a large group of systems are connected to private or public networks, where data owner can store his data on remote systems and frees himself from storage burden and uses the data on-demand, anytime, everywhere. As a Cloud data user does not possess direct control of his data, security is one of the few challenging issues which need to be addressed. Security in Cloud computing can be addressed in many directions viz. authentication, integrity, confidentiality and many more. Data integrity or correctness is an issue where there may be some unauthorized alteration in the data without consent of the data owner. Hence, data storage security in cloud computing is of the utmost importance nowadays. In this paper, we summarize some encryption-decryption algorithms which is used to secure the user's data in a cloud environment using a cloud simulation toolkit.

1. Introduction

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction[1]. This cloud model is comprises of five essential characteristics**,** three service models, and four deployment models.

The 'Cloud' itself is a virtualization of resources – networks, servers, applications, data storage and services – which the end user has on-demand access to. These resources can require minimal management or service provider interaction. Cloud Computing brings with it many benefits to the end user. These include[2]:

- Access to a huge range of applications without having to download or install anything.
- Applications can be accessed from any computer and from anywhere.
- Users can avoid expenditure on hardware and software; only using what they need.
- Companies can share resources in one place.
- Consumption is billed as a utility with minimal upfront costs.
- Scalability via on-demand resources.

1.1. Essential Characteristics of Cloud Computing

The essential characteristics of Cloud Computing are as follows[1]:

1) On-demand self-service: A consumer can simultaneously provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

2) Broad network access: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, tablets and workstations).

3) Resource pooling: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different virtual and physical resources dynamically assigned and reassigned according to the demands of the consumer. Examples of resources include storage, memory, processing and network bandwidth.

4) Rapid elasticity: Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand.

5) Measured service: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction

appropriate to the type of service (e.g., storage, bandwidth, processing and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

1.2. Cloud Service Models

The Cloud Service Models are as follows:

1) Software as a Service (SaaS): The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface.

2) Platform as a Service (PaaS): The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider.

3) Infrastructure as a Service (IaaS): The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications.

1.3. Cloud Deployment Models

The Cloud Deployment Models are as follows[1]:

1) Public cloud: The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.

2) Private cloud: The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.

3) Hybrid cloud: The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).

4) Community cloud: The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.

1.4. Security Concerns in Cloud Computing

The security concerns in cloud computing is as follows[3]:

Security concern #1: With the cloud model control physical security is lost.

Security concern #2*:* Company has violated the law (risk of data capture by (foreign) government).

Security concern #3: Service incompatibility

Securityconcern#4*:* Who controls the encryption/ decryption keys? Logically it should be the customer.

Security concern #5: Ensuring the integrity of the data (transfer, storage, and retrieval) really means that it changes only in response to authorized transactions. A common standard to ensure data integrity does not yet exist.

Security concern #6: Users must keep up to date with application improvements to be sure they are protected.

Security concern #7: Some government regulations have strict limits on what data about its citizens can be stored and for how long, and some banking regulators require that customer's financial data remain in their home country.

Security concern #8: The dynamic and fluid nature of virtual machines will make it difficult to maintain the consistency of security and ensure the auditability of records.

Security concern #9: Customers may be able to sue cloud service providers if their privacy rights are violated. Concerns arise when it is not clear to individuals why their personal information is requested or how it will be used or passed on to other parties.

1.5. System Model

1) Cloud User: the user, who can be an individual or an organization originally storing their data in cloud and accessing the data.

2) Cloud Service Provider (CSP): the CSP, who manages cloud servers (CSs) and provides a paid storage space on its infrastructure to users as a service.

3) Third Party Auditor (TPA) or Verifier: the TPA or Verifier, who has expertise and capabilities that users may not have and verifies the integrity of outsourced data in cloud on behalf of users. Based on the audit result, the TPA could release an audit report to user.

The cloud storage model consists of three main components as illustrated in figure 1[4].

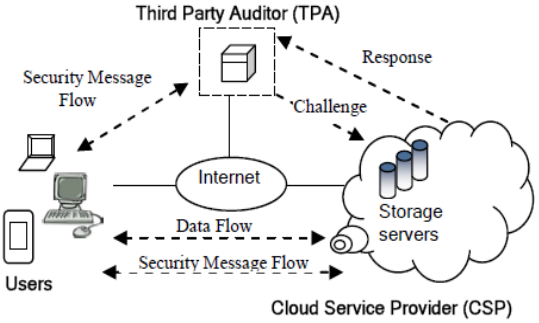

Figure 1. Cloud Data Storage Model [4]

2. Literature review

2.1. Data Storage in Cloud Computing

Cloud computing is an approach where software, hardware and/or other resources are provisioned "as a Service". Cloud brings some benefits to its users such as relief from the burden of storage management, universal access to data, ubiquitous, lower capital expenditure etc. Various issues related to Cloud computing include the following:

- Security of data from theft
- Data Integrity on Cloud
- Secure transmission of data to and from Cloud sever
- Verifying files without much overhead/Computation
- Rights management
- Maintain security during sharing and many more.

Data storage correctness or some time more generally referred as data integrity verification is one of the major Cloud security problems. Data can be altered by unauthorized entity without intimating to data owner. How would the data owner make sure that his data has not been modified by other intruders (or may be by the Cloud provider itself, accidently or intentionally). So detecting such kind of unlawful activities on data is an utmost priority issue. Data storage correctness schemes can be divided into two categories. The two categories are as follows:

1) Without Trusted Third Party (TTP) and
2) With TTP, based on who makes the verification.

As in traditional network security, we try to protect the confidentiality of data in its two stages of data life cycle and they are as follows:

1) Data at rest: For data at rest, symmetric key encryption techniques (E.g. AES, TDES, DES etc.) are recommended, which are secure but more time consuming approaches.
2) Data in transition: For data in transition, SSL kind of already available secure protocols are recommended. For integrity verification, hash functions such as SHA-1, MD-5 are relied upon.

The confidentiality and integrity of the outsourced data in clouds are of paramount importance for their functionality. The reasons are listed as follows [5]:

1) The CSP, whose purpose is mainly to make a profit and maintains a reputation, has intentionally hide data loss an incident which is rarely accessed by the users.
2) The malicious CSP might delete some of data or is able to easily obtain all the information and sell it to the biggest rival of Company.
3) An attacker who intercepts and captures the communications is able to know the user's sensitive information as well as some important business secrets.
4) Cloud infrastructures are subject to wide range of internal and external threats.

3. Algorithms and examples

3.1. Reverse Caesar Cipher

3.1.1. Introduction

One of the simplest examples of a substitution cipher is the Caesar cipher. It is said to have been used by Julius Caesar to communicate with his army. Caesar is considered to be one of the first persons to have ever employed encryption for the sake of securing messages. Further enhancement to original three places shifting of character in Caesar cipher uses modulo twenty six arithmetic encryption key that is greater than twenty six.

$$En(x) = (x+n) \bmod 26$$

The most pressing weakness of this cipher is simplicity of its encryption and decryption algorithms; the system can be deciphered without knowing the encryption key. It is easily broken by reversing encryption process with simple shift of alphabet ordering.

$$Dn(x) = (x-n) \bmod 26$$

The earliest caesar cipher method include the main drawbacks is plaintext and key is used only 26 alphabets.

This paper[6] overcome the above problem to plaintext is used case sensitive, numbers and special characters in order of ASCII full characters (256 char). This proposed method providing the inverse of Caesar cipher that supports more security for the data compared with the earliest Caesar cipher. And also it

can be used simply encode the message for preserving privacy. It is complicated to understand the cipher text compared with the other methods.

3.1.2. Algorithm

The figure 2[6] below shows the encryption/decryption process.

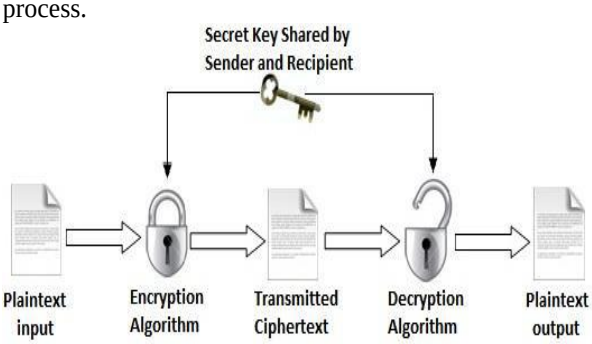

Figure 2. Encryption/Decryption Process [6]

Encryption Algorithm:

1. Split the letter of the plaintext.
2. Assign the position (i) of the letter.
3. Generate the ASCII value of the plaintext letter.
4. Assigned same Key value is considered as a key.
5. To apply the below given formula:

   $E = (p + k + i) \% 256$

   where, p – Plaintext, k – key, i – Position.
6. Generate the ASCII character of the corresponding decimal value in the result from the above given formula. This would be the cipher text.

Decryption Algorithm:

1. Generate the ASCII value of the cipher text character.
2. Here the same encryption key used.
3. Assigned the position (i) of the cipher text.
4. To apply the below given formula:

   $D = ((c – k – i) + 256) \% 256$

   where, c – Cipher text, k – key, i – Position.
5. Generate the ASCII character of the corresponding decimal value. This would be the original plaintext.

3.1.3. Example

Encryption:

Let, the character is “c”. Now according to the steps we will get the following:

1. ASCII of “c” is 99 in decimal.
2. Assign a fixed key value is 10.
3. Assign the position (i) is 0.
4. Apply the following formula ,

   E = (p + k + i) % 256

   = (99 + 10 + 0) % 256

   = 109
5. As per the algorithm the cipher text would be “m”.

Decryption:

After encrypting “c” we have got “m” as the cipher text. Now according to decryption algorithm let’s try to get back the original text i.e. “c”.

1. 109 is the ASCII value of the cipher text character “m”.
2. Here, Same key “10” is used.
3. Here, position (i) “0” is used.
4. The formula is applied to the ASCII value 109 of the cipher text character and key 10.

   D = ((c – k – i) + 256) % 256

   = ((109-10-0) + 256) % 256

   = 99
5. “c” is the ASCII character of the decimal 99. Character “c” would be the original plaintext.

3.1.4. Advantages

- The algorithm is very simple in nature.
- The user can easily encrypt and decrypt the combination of alphabets, numbers and special characters efficiently.
- It is case sensitive.

3.2. Advanced Encryption Standard (AES)

3.2.1. Introduction

- It’s 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.
- AES algorithm ensures that the hash code is encrypted in a highly secure manner.
- AES has a fixed block size of 128 bits and uses a key size of 128 in this paper[5].

3.2.2. Algorithm

1. Key Expansion
2. Initial Round
3. Add Round Key
4. Rounds
5. Sub Bytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.
6. Shift Rows—a transposition step where each row of the state is shifted cyclically a certain number of steps.
7. Mix Columns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.

8. Add Round Key—each byte of the state is combined with the round key; each round key is derived from the cipher key using a key schedule.
9. Final Round (no Mix Columns)
10. Sub Bytes
11. Shift Rows
12. Add Round Key

3.3. Data Encryption Standard (DES)

3.3.1. Introduction

- It's a symmetric-key encryption standard.
- Its block size is 64-bit and a 56 bit key is used during execution.
- It is a symmetric cryptosystem, specifically a 16-round Feistel Cipher.

3.3.2. Algorithm

1. Get the Plaintext
2. Get the Password
3. Convert the Characters into binary form
4. Derive the Leaders (L1 to L16) from the Password
5. Apply the Formula to get the encrypted and decrypted message

3.4. Rivest Shamir Adleman (RSA)

3.4.1. Introduction

RSA is widely used Public-Key algorithm. RSA stands for Ron Rivest, Adi Shamir and Len Adleman, who first described it in 1977. By securing the data, unauthorized access isn't allowed.

User data is encrypted first and then it is stored in the Cloud. When required, user places a request for the data for the Cloud provider; Cloud provider authenticates the user and delivers the data.

RSA is a block cipher, in which every message is mapped to an integer. RSA consists of Public-Key and Private-Key. In the Cloud environment, Pubic-Key is known to all, whereas Private-Key is known only to the user who originally owns the data. Thus, encryption is done by the Cloud service provider and decryption is done by the Cloud user or consumer. Once the data is encrypted with the Public-Key, it can be decrypted with the corresponding Private-Key only.

RSA algorithm involves three steps:

1. Key Generation
2. Encryption
3. Decryption

3.4.2. Algorithm

Key Generation- Before the data is encrypted, Key generation should be done. This process is done between the Cloud service provider and the user.

1. Choose two distinct prime numbers a and b. For security purposes, the integers a and b should be chosen at random and should be of similar bit length.
2. Compute n = a * b.
3. Compute Euler's totient function, Ø(n) = (a-1) * (b-1).
4. Chose an integer e, such that $1 < e < Ø(n)$ and greatest common divisor of e, Ø(n) is 1. Now e is released as Public-Key exponent.
5. Now determine d as follows: $d = e^{-1} (\text{mod } Ø(n))$ i.e., d is multiplicate inverse of e mod Ø(n).
6. d is kept as Private-Key component, so that d * e = 1 mod Ø(n).
7. The Public-Key consists of modulus n and the public exponent e i.e, (e, n).
8. The Private-Key consists of modulus n and the private exponent d, which must be kept secret i.e, (d, n).

Encryption- Encryption is the process of converting original plain text (data) into cipher text (data).

1. Cloud service provider should give or transmit the Public-Key (n, e) to the user who wants to store the data with him or her.
2. User data is now mapped to an integer by using an agreed upon reversible protocol, known as padding scheme.
3. Data is encrypted and the resultant cipher text(data) C is calculated using,
   $C = m^e \text{ (mod n)}$.
4. This cipher text or encrypted data is now stored with the Cloud service provider.

Decryption- Decryption is the process of converting the cipher text (data) to the original plain text(data).

1. The cloud user requests the Cloud service provider for the data.
2. Cloud service provider verifies the authenticity of the user and gives the encrypted data i.e, C.
3. The Cloud user then decrypts the data by computing,
   $m = C^d \text{ (mod n)}$.
4. Once m is obtained, the user can get back the original data by reversing the padding scheme.

3.4.3. Example

Key Generation:

1. We have chosen two distinct prime numbers a=61 and b=53.

2. Compute n=a*b, thus n=61*53 = 3233.
3. Compute Euler's totient function, Ø(n)=(a-1)*(b-1),
   Thus Ø(n)=(61-1)*(53-1) = 60*52 = 3120.
4. Chose any integer e, such that 1 < e < 3120 that is coprime to 3120. Here, we chose e=17.
5. Compute d , $d = e^{-1} (\text{mod } Ø(n))$,
   thus $d=17^{-1} (\text{mod } 3120) = 2753$.
6. Thus, the Public-Key is (e, n) = (17, 3233) and,
   the Private-Key is (d, n) = (2753, 3233).
   This Private-Key is kept secret and it is known only to the user.

Encryption:

1. The Public-Key (17, 3233) is given by the Cloud service provider to the user who wishes to store the data.
2. Let us consider that the user mapped the data to an integer m=65.
3. Data is encrypted now by the Cloud service provider by using the corresponding Public-Key which is shared by both the Cloud service provider and the user.
   $C = 65^{17} (\text{mod } 3233) = 2790$.
4. This encrypted data i.e, cipher text is now stored by the Cloud service provider.

Decryption:

1. When the user requests for the data, Cloud service provider will authenticate the user and delivers the encrypted data (If the user is valid).
2. The cloud user then decrypts the data by computing,
   $m = C^{d} (\text{mod } n) = 2790^{2753} (\text{mod } 3233) = 65$.
3. Once the m value is obtained, user will get back the original data.

3.4.4. Advantages

Only the authorized user can access the data. Even if some intruder (unauthorized user) gets the data accidentally or intentionally if he captures the data also, he can't decrypt it and get back the original data from it. Hence forth, data security is provided by implementing RSA algorithm.

4. Cloud simulation tools

4.1 CloudSim

- support for simulation and modeling of large scale Cloud computing data centers
- virtualized server hosts, with customizable policies for provisioning host resources to virtual machines
- energy-aware computational resources
- data center network topologies and message-passing applications
- support for dynamic insertion of simulation elements, stop and resume of simulation
- support for user-defined policies for allocation of hosts to virtual machines and policies for allocation of host resources to virtual machines

4.2 CloudAnalyst

- Easy to use Graphical User Interface (GUI)
- Ability to define a simulation with a high degree of configurability and flexibility
- Repeatability of experiments
- Graphical output
- Use of consolidated technology and ease of Extension (Java Swing)

4.3 GreenCloud

- Simulation environment for energy-aware cloud computing data centers.
- GreenCloud is an extension of the well-known NS2 network simulator.
- Focused primarily on the communications within a cloud, i.e., all of the communication processes are simulated on packet level.

4.4 iCanCloud

- Both existing and non-existing cloud computing architectures can be modeled and simulated
- A flexible cloud hypervisor module
- Customizable VMs can be used to quickly simulate uni-core/multi-core systems.
- provides a user-friendly GUI to ease the generation and customization of large distributed models.
- provides a POSIX-based API and an adapted MPI library for modeling and simulating applications.
- New components can be added to the repository of iCanCloud to increase the functionality

5. Results and discussion

Table 1[6] states the comparison between Reverse Caesar Cipher, AES, DES and RSA algorithms. These algorithms have some disadvantages. The disadvantage of AES algorithm is that it requires more rounds of communication as compared to DES. The DES algorithm has the following disadvantages:

- Its key size (56 bits) is too short for proper security. Also, DES uses 64-bit blocks, which

raises some potential issues when encrypting several gigabytes of data with the same key.
- It is vulnerable to brute force attacks.
- Only one private key is used for encryption as well as for decryption because it is symmetric encryption technique so if we lose that key to decrypt the data then we cannot get the readable data at the receiving end.

The RSA algorithm has the following disadvantages:
- A disadvantage of using public-key cryptography for encryption is speed. Since RSA has two keys, it requires significant amount of calculations, it will take a lot of time to encrypt and decrypt large files. Hence, it works well for small files.
- Performance is often a disadvantage. Cipher texts are much larger than the plaintexts, so communication requirements typically go up. The computations on these large cipher texts are typically slower than if you just performed the computation on the plaintext itself.

Table 1. Comparison between RCC, AES, DES and RSA [6]

Factors	RCC	AES	DES	RSA
Input Size	20	56	29	
Key Size	256 bits	128 bits	64 bits	
Cipher Text	Substitu-tion Cipher	Symmet-ric Block	Symmet-ric Block	Reversible Protocol
Security	Security Conside-red	Proven Inadequ-ate	Proven Inadequ-ate	Security Considered
Memory	1.56	66.23	54.68	
Possible Keys	2^{256}	2^{128}	2^{56}	
CPU Usage	5	45	35	

6. Conclusion and future scope

Security of the end-user data stored on the cloud is a major concern now-a-days. Hence, to secure the data stored on the cloud various algorithms have been summarized in this paper along with the cloud simulation tools. Through this paper, we summarize Reverse Caesar Cipher, RSA, AES and DES algorithms which are used for securing the data stored on the cloud. The algorithms can be implemented in java using the CloudSim Toolkit, Eclipse and Ant packages. By using these packages a virtual cloud-like environment can be created which can be used to study and analyze the data stored on the cloud by the end-user. These algorithms might have some disadvantages/drawbacks. In due course we will come up with some improvements for these algorithms so that the disadvantages/drawbacks are eliminated which in turn can be helpful to increase the security for the end user's data stored on the cloud.

7. References

[1] Peter Mell, Timothy Grance, "The NIST Definition of Cloud Computing", *National Institute of Standards and Technology*, September 2011, Special Publication 800-145.

[2] Mythry Vuyyuru, Pulipati Annapurna, K.GanapathiBabu, A.S.K Ratnam, "An Overview of Cloud Computing Technology", July 2012, *International Journal of Soft Computing and Engineering (IJSCE)*, ISSN: 2231-2307, Volume-2, Issue-3, pp.244-246.

[3] Kalyani D. Kadam, Sonia K. Gajre, R. L. Paikrao, "Security issues in Cloud Computing", 2012, *National Conference on Innovative Paradigms in Engineering & Technology (NCIPET-2012)*, pp.22-26.

[4] Syam Kumar P, Subramanian R, "An Efficient and Secure Protocol for Ensuring Data Storage Security in Cloud Computing", November 2011, *IJCSI International Journal of Computer Science Issues*, Vol. 8, Issue 6, No 1, pp.261-274.

[5] Vasu Raju, Raj Kumar, and Anand Raj, " Techniques for Efficiently Ensuring Data Storage Security in Cloud Computing", October 2011, *IJCTA*, ISSN:2229-6093, Vol 2(5), pp. 1717-1721.

[6] P. Subhasri, Dr. A Padampriya, "Implementation of Reverse Caesar Cipher Algorithm for Cloud Computing", July 2013, *International Journal for Advanced Research in Engineering and Technology (IJARET)*, ISSN 2320-6802, Vol. 1, Issue VI, pp.43-46.

[7] Dr. A Padampriya, P. Subhasri, "Cloud Computing: Reverse Caesar Cipher Algorithm to Increase Data Security", April 2013, *International Journal of Engineering Trends and Technology (IJETT)*, ISSN 2231-5381, Volume 4, Issue 4, pp.1067-1071.

[8] K. S. Suresh, Prof K.V. Prasad, "Secutiy Issues and Security Algorithms in Cloud Computing", October 2012, *International Journal of Advanced Research in Computer Science and Software Engineering*, ISSN: 2277 128X, Volume 2, Issue 10, pp.110-114.

[9] Neha Jain, Gurpreet Kaur, "Implementing DES Algorithm in Cloud for Data Security", 2012,

VSRD International Journal of Computer Science and Information Technology (VSRD-IJCSIT), Vol. 2(4), pp.316-321.

[10] Maha TEBAA, Said EL HAJJI, Abdellatif EL GHAZI, "Homomorphic Encryption Applied to the Cloud Computing Security", July 4-6, 2012, *Proceedings of the World Congress on Engineering*, ISSN: 2078-0958 Vol I.

[11] Parsi Kalpana, Sudha Sigaraju, "Data Security in Cloud Computing using RSA Algorithm", September 2012, *International Journal of Research in Computer and Communication Technology (IJRCCT)*, ISSN: 2278-5841, Vol 1, Issue 4, pp.143-146.

[12] K. Sunitha, S. K. Prashanth, "Enhancing Privacy in Cloud Service Provider Using Cryptographic Algorithm", July-August, 2013, *IOSR Journal of Computer Engineering (IOSR-JCE)*, ISSN: 2278-8727, Volume 12, Issue 5, pp.62-64.

[13] Dr. A Padampriya, P. Subhasri, "Cloud Computing: Security Challenges and Encryption Practices", March 2013, *International Journal of Advanced Research in Computer Science and Software Engineering*, ISSN: 2277 128X, Volume 3, Issue 3, pp.255-259.

[14] N. Saravanan, A. Mahendiran, N. Venkata Subramanian and N. Sairam, "An Implementation of RSA Algorithm in Google Cloud using Cloud SQL", 2012, *Research Journal of Applied Sciences*, Engineering and Technology 4(19): 3574-3579, ISSN: 2040-7467.

[15] Mandeep Kaur, Manish Mahajan, "Using Encryption Algorithms to Enhance the Data Security in Cloud Computing", January 2013, *International Journal of Communication and Computer Technologies*, ISSN: 2278-9723, Volume 01-No. 12, Issue: 03, pp.56-59.