Cloud Security Enhancement by Client-side Data Partitioning and Encryption

Debanjan Deb
Btech, Department of CSE
SRMIST
Chennai,603203,India
debanjan.deb97@gmail.com

Aviral Verma
Btech, Department of CSE
SRMIST
Chennai,603203,India
aviralverma97@outlook.in

J.V. Vidhya
Asst. Professor, Department of CSE
SRMIST
Chennai,603203,India
vidhya.j@ktr.srmuniv.ac.in

Abstract - Cloud Computing is a technology that has seen a spurious growth recently and is being deployed for personal as well as business purposes. Cloud computing can be defined as the distribution of computing services – namely storage, databases, software, networking, intelligence and analytics across the Internet to provide greater pace to innovation and accelerate resources with increased flexibility.

With this feature becoming increasingly integral to various services provided across the inter-connected digital world, it is imperative that its susceptibility be assessed and security made impenetrable to protect the sensitive information Cloud servers store. Cloud security has been vulnerable to threats and in several cases has led to Data Loss, Information Hacking and Denial of Services. These incidents have given rise to widespread concern regarding the data security that these Cloud Services employ. However, security models and security tools are being continually enhanced. This project aims to implement a Security Enhancement mechanism that gives the client-side greater control over data security and access.

Index terms: - Cloud Security, Diffie-Hellman Key Exchange, AES Encryption Standard, Multiple Blocks Cloud Storage.

I. INTRODUCTION

Cloud Computing has been in a steep ascent in the recent years due to the flexibility the service provides to each of the Service Providers and the Customers. It provides a shared pool of resources that are accessible over the internet over multiple devices [1]. Data is stored on a server on the internet instead of storing locally. This enables easier access of data as well as frees up local storage space.

Cloud computing also enables a number of resources to be interconnected and work on a single resource. There are various benefits associated with Cloud Computing. These are be summarized as below,

i. Cost: Cloud Computing eliminates the expenses of software and hardware in bulk at datacentres as the service requires limited components. There is no need of setting up on-site data centres; instead, data is accessible over a shared platform on the internet.

ii. Speed: Cloud Computing increases the speed of service as the service is provided over the internet on a pool of resources. Thus, the service is not monopolised and dependent on a single system.

iii. Productivity: Cloud computing minimises on-site operation[6] such as “racking and stacking”—setting up hardware, patching software and other laborious IT management tasks.

iv. Performance: The greatest benefit of Cloud computing comes in the enhanced performance of the service. As these services are provided over a large network of datacentres, there is regular Updation of software and gives reliable performance.

Cloud Security refers to the mechanisms, technologies and policies deployed to protect data, communication and infrastructure of the Cloud computing service. Cloud Security is an integral part of the Cloud computing service. It is a sub-category of Information Security and Network Security.

1.1 Background

As per the current system, cloud security is still an area of concern. With the help of cloud computing, organizations can utilize services and information is stored at a physical location that outreaches their control. This utility gave rise to various security questions like privacy, confidentiality, integrity and demand a reliable computing atmosphere wherein data discretion be upheld. In order to sufficiently rely on the service of computing, there is an imminent call for a system that performs verification, authentication and encrypted transfer of data, thereby preserving data privacy. Here in our project we are trying to point out some measures, which will make it more safe and secure.

1.2 Innovation Idea

The idea that motivates this project is the collaboration of cryptography with cloud computing to ensure an efficient

security mechanism that successfully protects data storage as well as authenticates data access.

At its core, Cloud Services are nothing but widely inter-connected network of configurable system resources.

Cryptography is equipped with various protocols, in our focus namely the Diffie-Hellman Key Exchange and Advanced Encryption Standard (AES-256) that are utilised for secure data transfer over a network.

The idea is to implement this feature of cryptography on cloud servers to resolve the concerns over Cloud Security.

Moreover, the purpose is to build an implementation that is capable of partitioning the various data for storage over distributed server systems. [5] The partitioning of data along with encryption and access protocol will provide an efficient encryption scheme for cloud computing enhancing its security and providing more power to client.

II. EVOLUTION

The work done until date can be distributed into three periods:

i. Idea period
ii. Pre cloud period
iii. Current Cloud period

During1960s, John McCarthy, Douglas Parkhill, and others reconnoitred the idea of computing in the form of a public service. The idea was to use commodity hardware and software, computing resources to delivered an infinite elastic online public utility.[3]

In the late 1960s, J.C.R. Licklider inspired the ARPANET, and in the early 1970s global networking became a reality. In the early 1980s, the TCP/IP suite emerged as the protocol for ARPANET, and the Domain Name System (DNS) established naming designations for websites.

The now widely recognised cloud phase began the year 2007 when the cataloguing of IaaS, PaaS, and SaaS were officially finalized.[9] The cloud-computing chapter of history has seen some peculiar and intriguing breakthroughs initiated by the leading web officialdoms of the digital world.

III. LITERATURE SURVEY

The Literature Survey is enclosed in the table 3.1.

IV. CLIENT SIDE SECURITY ENHANCEMENT PHASES

The process involved in proposed system is comprised of the following modules,

i. Partitioning
ii. Encryption
iii. Key-Exchange
iv. Decryption
v. Merging

Each section varies greatly in the methodologies.

1. *Partitioning*

a) The first phase of the system proposes partitioning of the data that is to be stored over a cloud storage server into several chunks.

b) The said functionality is achieved by splitting files of all formats such as *.txt, *.jpg, *.mp4 etc. into smaller fixed size chunks using the partitioning module.[11]

c) The system reads the files selected by the user and requests the chunk size i.e. the maximum size of each of the chunks the file will be divided into.

d) The optimal chunk size is dependent on the cloud service the customer is utilising.[12] For example, Amazon S3 requires chunk size of 5 MB minimum whereas Azure demands they must be much less than 4 MB.

e) The program then reads the source file up to the specified chunk size and writes on to an output part file iteratively until the end of file is reached.

f) Thus, the client is provided with numerous divisions of a single file that can be further encrypted and uploaded on to a cloud storage.

2. *Encryption*

a) The Data Encryption Module holds the responsibility of encryption and decryption of a data at the client side. It is a simple module that serves the purpose of encrypting each file individually.[6] It is designed with cloud storage in mind.

b) The Data Encryption Module uses the AES-256 (32 Bit Key) for encrypting the data. Using the module, the client can specify a chosen directory that contains the various files required to be encrypted before upload on the cloud storage. The said directory can also be the primary cloud sync folder.

c) The module has been built using Python 2.7 and the PyCrypto module. At execution, it asks for the complete address of the directory containing sensitive files and demands a 32-bit key.[10].

TABLE 3.1 LITERATURE SURVEY

Author/Year	Title	Main Focus	Algorithm	Drawbacks
Rongxing et al **Publisher :** ASIACCS **Year :** 2010	Security Provenance: The Essential Bread & Butter of Data Forensics in Cloud Computing	New Security and Provenance Data Forensics Tool	Bilinear Pairing Method	Complex Mathematical Models
R. La'Quata Sumter **Publisher :** ACMSE **Year :** 2009	Cloud Computing: Security & Risk Classification	Store Information of every Process on Cloud Servers	Security Capture Device	Practical only for Small Cloud Environments
Mladen A. Vouch **Publisher :** IEEE **Year :** 2008	Cloud Computing Issues, Research & Implementation	Cloud Computing with Virtualization, Cyber Infrastructure, SOA and End Users	Issue Authentication	User Dissatisfaction
Wenchao **Publisher :** IEEE **Year :** 2010	Towards a Data Centric View of Cloud Security	Data Centric Perspective, Forensic, System Analysis and Data Management	Declarative Secure distributed Systems (DS2), Secure Network Data Log (SeNDlog)	Complex Implementation. Awaits Evaluation from Cloud Vendors
Soren Bleikertz **Publisher :** CCSW **Year :** 2010	Security Audits of Multi-Tier Infrastructures in Public Infrastructure Clouds	Implement Cloud Security Analysis tool and simulate it to real factors	Complex Query Language over Amazon's Elastic Compute Cloud (EC2) and Python	Software is linked to work with EC2 and not in general systems
Flavio Lombard, Roberto Di Pietro **Publisher :** IEEE **Year :**2010	Transparent Security for Cloud	Transparent Cloud Protection System (TCPS) for better Security Management	Secure Architecture system of TCPS	Failed to deploy realistic scenarios and validate their work
Wayne A. Jansen **Publisher :** IEEE **Year :** 2011	Cloud Hooks: Security and Privacy Issues in Cloud Computing	The essentiality of configuring security on critical systems	Revamped Security Policies with Strong Commands	Unknown outcome of tool or a solution on real infrastructure
Jinpeng et al **Publisher :** CCSW **Year :** 2009	Managing Security of Virtual Machine Images in a Cloud Environment	Cloud's Image Repository.	Use a Filter to capture malware and replace all sensitive passwords with stronger ones.	Captures weren't 100 % accurate and could lead to legitimate issues. Required regular Updation
Miranda & Siani **Publisher :** IEEE **Year :** 2009	A Client-Based Privacy Manager for Cloud Computing	A client-based privacy manager tool for information processing in the cloud	Client-side Key Authentication Mechanism	Lack of implementation in all scenarios
Dan Lin, Anna Squicciarini **Publisher :** SACMAT **Year :** 2010	Data Protection Models for Service Provisioning in the Cloud	Data Protection Framework for sensitive information	Policy Ranking, Integration and Enforcement	Not validated on real environments.

d) The key specified by the user is then used to create and encrypted index file and various files with randomly generated names.

e) The data in the files along with its metadata is safely encrypted.

3. *Key-Exchange*

a) The Diffie-Hellman Key Exchange module is responsible for generating a public-private key pair using the Diffie-Hellman algorithm for authentication of access.

b) The key pair thus generated is used to exchange the 256 bit AES key for decryption.[10]

4. Decryption

a) At the receiver's end, the encrypted files when downloaded are decrypted using the AES key and the index.

b) The index provides the information pertaining to various source file names and their corresponding randomly generated names.[7] It also gives information regarding original file formats.

c) The files are then decrypted in similar manner as the AES Decryption process.

d) The result is a series of decrypted files that require merging to access the original files.

5. Merging

a) The system merges the various part files by accessing the manifest to gain information regarding the original file names.

b) When the user provides the original source file name along with the command to merge it's part files, the system obtains all the part files downloaded.

c) The system uses regular expression to match all the part files in the directory with the source file name.

d) For each chunk matched, the system reads the data stored in them and writes them onto a new file that replicates the source fil, thereby merging all the chunks.[12]

e) The part files are then deleted from the directory.

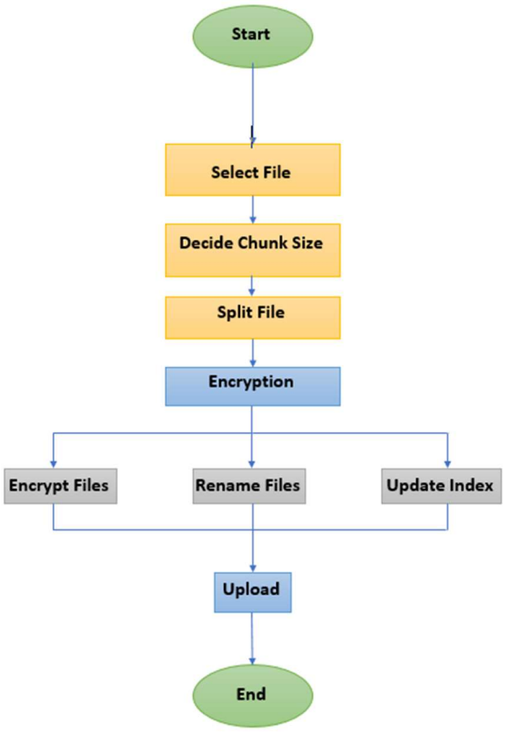

Fig. 1 Process Flow

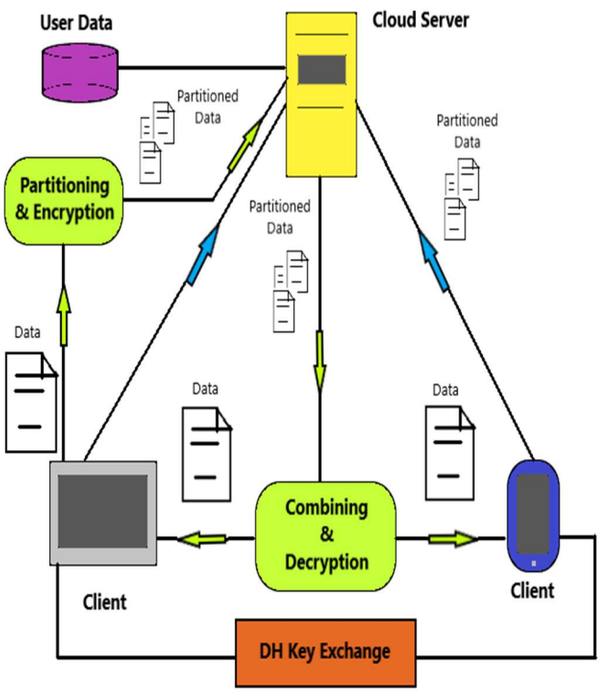

Fig. 2 System Architecture

V. RESULTS

In several test runs across multiple devices, we proceeded to select a directory containing various files of different sizes and types such as images, videos, documents, PDFs, audios etc.

The directory as stated by the user in the program is then used as the home directory of the File Splitting module. For each file present in the directory, the program demands of the user the optimal chunk size that will become the maximum size of each chunk file the original file is split into.

1. Server Side

In our test run, we selected three files namely, 696.pdf (PDF Document), Batman.mkv (Video) and Sunrise.mp3 (Audio). With an optimal chunk size provided by the user of 500 KB, the said files were split into 4, 3 and 2 parts respectively.

Each chunk file limited at a maximum of 500 KB while few with the residual size observed as the last part files. These part files are named in the format file_name-(part number).file_type.

Thus once all the files in the directory are partitioned, the program moves onto the second module i.e. the Directory Encryption Module.

The user is then prompted by the program to provide with a key of amenable size.

In this next stage, the directory as provided by the user is encrypted using the AES-256 algorithm with a 32 bit key generated by the password provided by the user and an Initialisation Vector produced using SHA algorithm.

The key value is then used to encrypt all the files in the directory also renaming each file by a random 16 bit Hexagonal number. This function adds to the security feature while maintaining the metadata associated with each file. Eg, 26fci97hjres45asutahijgiod887.enc

The program also creates an index of all the files containing a dictionary with a key-value pair that imitates the original filename – new filename storage. This index is stored in the directory into an index.json file wich is also further encrypted.

The index file is deemed necessary for decoding original file names at the receiving end.

The user is finally prompted to communicate the password through DH Key Exchange Protocol as stated previously.

The whole directory is then ready to be uploaded onto any cloud server.

2. *Client Side*

The client then downloads the said files onto their local machine from the cloud server. The process of decryption starts with the Directory Decryption Module.

In this module, the user now is asked of their keys regarding the DH Key Exchange Process to retrieve the password required to decrypt all the files. With the correct password in place the user enters this password prompted by the program to initiate the decryption process.

Firstly, the index file is read to rename all the files in the directory from random number to their original post encryption file names. The file is read onto the program and the dictionary is retrieved to splat out the required information.

The Decryption process then begins using the same 32 bit key to decrypt all the files at each thread into original data. As the process continues each file is restored to its original form i.e. their original names and types.

Thus once the decryption is complete we are left with the various part files such as 696.pdf-1, Batman.mkv-3 etc. The index file is the only exception of the process. It is left as is.

The part files are then merged in the File Merging Module wherein the user is prompted to provide the filename they wish to be merged. Using regular expressions, all the part files of the file demanded are matched and are rewritten as a single file.

Observedly, Batman.mkv-1, Batman.mkv-2 and Batman.mkv-3 are merged into singular file Batman.mkv.

Similarly all the files are merged in the directory and the original data is restored.

3. *Performance Analysis*

Implementing Threading into the Encryption Process enhances the performance of the program as depicted by the graph below.

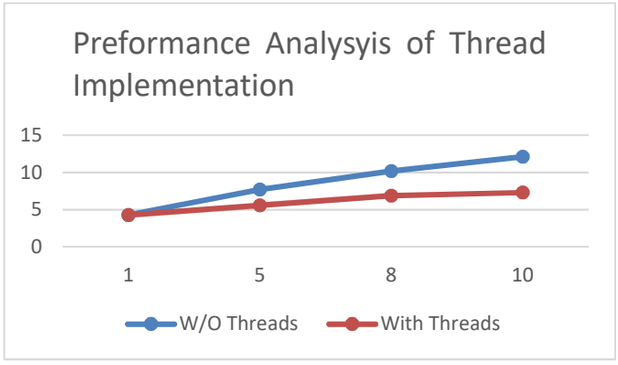

Fig. 3 Performance Analysis

VI. CONCLUSION

In conclusion, we have represented the security means that possess the ability to secure the data of users stored on cloud servers against malicious entities. Insecure breach into the server is completely avoided by Diffie-Hellman key exchange algorithm. Implementation of AES cryptographic algorithms over a cloud computing environment is also ventured. The data partitioning scheme provides a dynamic perspective towards protection & security over cloud systems.

Providing security to the customer's data on the cloud will empower the standing of Cloud Computing & Storage.

An efficient program is presented that promises to give much required power and control to the most at risk party of Cloud Computing, i.e. the customer.

Providing security to the customer's data on the cloud will empower the standing of Cloud Computing & Storage.

ACKNOWLEDGMENT

We would like to express our humble gratitude to Dr. Sandeep Sancheti, Vice Chancellor of SRMIST, for the facilities extended towards project work and continued support. We extend our sincere thanks to Dr. Muthamizhchelvan, Director, Faculty of Engineering and Technology, SRMIST for his invaluable guidance. We wish to thank Dr. B. Amutha, Head of Department, and Dr. K. Annapoorni, Academic Advisor, Computer Science and Engineering, SRMIST for their leadership and motivation. Our inexpressible respect and gratitude to Ms. J.V. Vidhya, Assistant Professer, SRMIST for providing us with an opportunity to pursue our project under her mentorship. She has provided us the freedom and support to explore the research topics of our interest. We sincerely thank staff and students of the Computer Science and Engineering Department of SRMIST for their help during our research.

REFERENCES

[1]. V. Fusenig, A. Sharma 2012. Security architecture for cloud networking, in: 2012 International Conference on Computing, Networking and Communications (ICNC). Presented at the 2012 ICNC Conference.

[2]. B.R. Kandukuri, V.R. Paturi, A. Rakshit, 2009. Cloud Security Issues, in: IEEE International Conference on Services Computing, SCC 2009.

[3]. S. Ramgovind, M.M. Eloff, E. Smith, 2010. The management of security in Cloud computing, in: Information Security for South Africa (ISSA), 2010.

[4]. Karun Handa, Uma Singh, 2015. Data Security in Cloud Computing using Encryption and Steganography. International Journal of Computer Science and Mobile Computing. IJCSMC, 2015.

[5]. H. Tianfield, 2011. Cloud computing architectures, in: 2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC). Presented at the 2011 IEEE SMC Conference.

[6]. Op – ed : Encryption, not restriction, is the key to safe cloud computing. https: / / www. nextgov.com/cloud-computing/2012/10/oped-encryp-tion-not-restriction-key-safe-cloudcomputing/58608

[7]. " Cloud Security and Privacy " , Tim Mather, Shahed Latif and Subra Kumaraswamy. – O'Reilly Book.

[8]. David Talbot. " How Secure Is Cloud Computing?" Technology Review [Online].Available: http://www.technologyreview.com/computing/23951/ 2009

[9]. Tania Gaur, Divya Sharma, 2016. A Secure and Efficient Client-Side Encryption Scheme in Cloud Computing. I.J. Wireless and Microwave Technologies. IJWMT 2016.

[10]. Elminaam, DiaaSalama Abdul et al. "Performance Evaluation of Symmetric Encryption Algorithms." IJCSNS International Journal of Computer Science and Network Security 8.12. 2008

[11]. "Heuristics in Physical Design Partitioning ", Bhargab Sinha, Naushad Manzoor Laskar, Rahul Sen.

[12]. "Security Aware Partitioning for efficient file system search", Aleatha Parker Wood, Christina Strong, Ethan L. Miller.