An Application to Secure Multimedia Data in Cloud Environment using Paillier Cryptography

N. Srinivasa Rao [1], S.Rama Sree [2]

[1] Pursuing M.Tech, Department of CSE, [2] Professor, Head of the Department CSE
[1,2] Aditya Engineering College, Surampalem, Kakinada, E.G D.t, A.P, India

***Abstract*— The Cloud is an internet-based computing where shared resources, software and information are provided to the computer. Now a days it is one of the prominent technologies to provide wide variety of services. In this condition data can be categorized into sensitive and insensitive. Among these two types sensitive data requires security. The existing system does not provide any security concerns for storing sensitive data in the cloud environment. So, it is necessary to develop an application to provide better security to the sensitive data in the cloud environment. This paper presented one of the best cryptographic algorithm that is homomorphic encryption which is used to provide better security to the multimedia data in the cloud environment when streaming or after streaming is over. Through this algorithm data can be encrypt and upload into cloud environment as well as download and decrypt into original form. By using this application we can provide better security to the cloud environment when compare to existing one.**

***Keywords*— Homomorphism, Cryptography, Public Key, Private Key, Euler's totient function, Carmichael's function, Frame Buffer.**

I. INTRODUCTION

The main intention of the proposed method is, to develop an application which is used to provide a better security to the cloud environment by using homomorphic encryption algorithm. The basic idea was to encrypt the data before sending them to the Cloud provider. If the client want to perform any operation on encrypted data, it is necessary to decrypt that data first. For that client will need to contribute the private key to the server to decrypt the data before perform the operations required, which might affect the confidentiality of data stored in the Cloud [1].

The Homomorphic Encryption is one of the best algorithm which is used to perform operations of encrypted data without decrypting them. This work mainly focuses on developing the application of Homomorphic Encryption method which executes the calculations of encrypted confidential data without decrypting them.

II. RELATED WORK

A. Cloud Computing

Cloud computing is one of the important aspect in the information technology which contains all the IT components (hardware, software, networking, and services) that are essential to development and delivery of cloud services through the Internet or a private network.

The above definition does not mention any security concept for storing data in the Cloud. So we can easily understand that the Cloud Computing is having deficiency about security, confidentiality and visibility. To Provide Infrastructure as a Service (IaaS), Platform as a Service (PaaS) or Software as a Service (SaaS) is not enough if the Cloud provider doesn't provide a better security and confidentiality for customer's data.

The proposal is to encrypt the multimedia data before sending it to the cloud environment, but to perform the operations the data should be decrypted every time we need to strive on it. Up to now it was unfeasible to encrypt data and relay on the third party to keep them secure and able to perform distant operations on them. So to permit the Cloud provider to execute the operations on encrypted data without decrypting them requires using the cryptosystems based on Homomorphic Encryption.

B. Mathematical functions and notations

To understand how Paillier Cryptosystem works, we need to know the following basic mathematical concepts.

1) General common divisor (gcd): The GCD of two or more non zero integers is the largest positive integer that divides the numbers without a remainder. The greatest common divisor of a and b is written as: gcd (a, b).

For example $g$cd (4, 6) = 2, gcd (4, 14) = 2.

Two numbers are called *coprime* or *relatively prime* if their greatest common divisor equals to 1. For example, 9 and 28 are relatively prime.

2) Least common multiple (lcm): The lcm of two or more non zero integers is the smallest integer that is divisible by every member of a set of numbers without a remainder. The least common multiple of a and b is written as: lcm (a, b).

For example, $l$cm (4, 6) = 12, $l$cm (4, 14) = 28.

It's a remarkable fact that ab= $l$cm (a, b) $*$gcd(a,b) thus $l$cm (a,b) = ab/gcd(a,b)

This fact can be easily seen that gcd (a,b) is the product of the common prime factors of ab, and the remaining factors would result $l$cm(a,b).

3) Euler's totient function (phi (φ) function) the totient of a positive integer ***n*** is defined to be the number of positive integers less than or equal to ***n*** that are coprime to ***n***.

For example φ (9) = 6, since the six numbers 1, 2, 4, 5, 7 and 8 are coprime to 9.

If ***n*** can be factorized to distinct prime numbers ***p*** and ***q***, then $\varphi(n) = (p-1)(q-1)$.

For example: ϕ (15) = ϕ (3∗5) = (3−1) (5−1) = 8

4) Carmichael's function (λ function) is given by the least common multiple (lcm) of all the factors of the totient function ϕ (n). If n can be factorized to prime number p and q. Then $\lambda(n) = \text{lcm}(p-1, q-1)$.

5) Modular multiplicative inverse of an integer *a* modulo *m* is an integer *x* such that $a^{-1} \equiv x \bmod m$ this is equivalent to $ax \equiv 1 \bmod m$.

The multiplicative inverse of *a* modulo *m* exists if and only if *a* and *m* are coprime (i.e., if gcd (*a*, *m*) = 1).

6) Converting a decimal Number to any base number: The remainders that we get when we sequentially divide the decimal number by the base end up being the digits of the result, which are read from bottom to top.

Example: convert 190_{10} to base 3.
$190 = 2*3^4 + 1*3^3 + 0*3^2 + 0*3^1 + 1*3^0$

The following notations are used frequently in Paillier Cryptosystem explanation:

$\mathbb{Z}n$ - set of integers n
$\mathbb{Z}n^*$ - set of integers coprime to n - this set consists of ϕ (*n*) number of integers.
$\mathbb{Z}_{n2}{}^*$ - set of integers coprime to n^2 - this set consists of *n* ϕ (*n*) number of integers.

III. Proposed Architecture Model

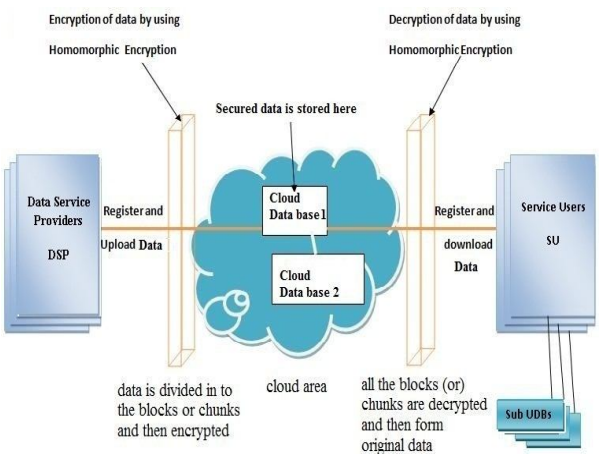

Fig 1: cloud framework with secure multimedia data transferring

Proposed architecture in fig1 is having three parties.

A) Cloud Service Provider:

It has a responsibility to maintain data or videos uploaded by data service providers. It has to maintain cache memory like mechanism called as sub video cloud databases for fast access. Its responsibility is to maintain recently accessed data.

B) Data Service Provider:

It is the owner of data file. It will decide whether the data to be uploaded is sensitive or insensitive. If it is insensitive no encryption is required and directly uploaded to the cloud. If it sensitive then he will apply proper authorization constraints to access that video. Before uploading this video he will encrypt that data. Later on if any user wants to see that data he should be authorized and provided with temporary access to decrypt that data [2].

C) Data Service User:

It is the end user or viewer of that data. It can see the insensitive data directly. But to see sensitive data it has to submit its authorization details with proper id. It should be registered at cloud server. Authentication information is accessed by the service provider to grant/revoke the services to that user.

Proposed model is twofold. First, it has to provide data security from the cloud service providers. Second objective ensures without compromising transmission time and quality constraints under encryption domain.

First objective can be achieved through homomorphic encryption proposed in [3].It is described as follows.

IV. Homomorphic Encryption

Homomorphic encryption is one for the form of encryption which allows particular types of operations to be carried out on cipher text and produce an encrypted result which, when decrypted, companion the result of operations performed on the plaintext.

Homomorpic encryption is one of the enviable features in modern communication system architectures. Homomorphic encryption would allow the chaining together of different services without displaying the data to each of those services, for example a chain of different services from different companies could 1) compute the tax 2) the currency exchange rate 3) shipping, on a transaction without exhibiting the unencrypted data to each of those services [4].

For homomorphic encryption we are using pailliers algorithm [5]. Now our proposal is to encrypt the multimedia data (i.e. Video, Audio, Text, Image, Graphics) before sending it to the cloud, but to perform the operations the data should be decrypted every time and we need to work on it. Till now it was impossible to encrypt data and to trust a third party to keep them safe and able to perform different

operations on them. So to allow the cloud user to perform the operations on encrypted data without decrypting them requires using the cryptosystems based on Homomorphic Encryption [6].

And this algorithm used to perform operations on encrypted data without knowing the private key (without decryption).

A) Key Generation Algorithm:

Choose any two large prime numbers p and q randomly and independently of each other such that the equation is:
gcd (pq, (p–1) (q–1)) =1.

1. This property is assured if both primes are of equal length.[3]
2. Compute n = pq and λ= lcm (p–1, q–1).
3. Select random integer g where $g \in \mathbb{Z}_{n^2}^*$
4. Ensure n divides the order of g by checking the existence of the following modular Multiplicative inverse:
   μ= (L (g λ mod n2)) $^{-1}$ mod n
   Where function L is defined as:
   L (u) = (u-1)/n.

Note that the notation a/b does not denote the modular multiplication of a times the modular multiplicative inverse of b but rather the quotient of a divided by b, i.e., the largest integer value $v \geq 0$ to satisfy the relation $a \geq vb$.

The public (encryption) key is (n, g).
The private (decryption) key is (λ, μ).

If using p, q of equivalent length, and a simpler variant of the above key generation steps would be to

set g = n+1, λ= φ (n),
and μ = φ (n)$^{-1}$ mod n,
where φ (n) = (p-1) (q-1).

Encryption:

1. Let m be a message to be encrypted where m Є Z_n
2. Select random r where $r \in \mathbb{Z}_n^*$
3. Compute cipher text as: c = g^m . r^n mod n^2. [10]

Decryption

1. Let c be the cipher text to decrypt, where $c \in \mathbb{Z}_{n^2}^*$
2. Compute the plaintext message as:
   m = L (c^λ mod n^2). μ mod n

As the original paper points out, decryption is "essentially one exponentiation modulo n^2."

B) Homomorphic Properties

One of the best features of the Paillier cryptosystem is its homomorphic properties [6]. As the encryption function is additively homomorphic, the following identities can be described:

1) Homomorphic Addition of Plaintexts: The product of two cipher texts will decrypt to the sum of their corresponding plaintexts,

D (E (m_1, r_1).E (m_2, r_2) mod n^2) = m_1+m_2 mod n

The product of a cipher text with a plaintext raising g will decrypt to the sum of the corresponding plaintexts,

D (E (m_1, r_1). g^{m2} mod n^2) = m_1+m_2 mod n.

2) Homomorphic Multiplication of Plaintexts: An encrypted plaintext raised to the power of another plaintext will decrypt to the product of the two plaintexts,

3)

D (E (m_1, r_1) ^{m2} mod n^2) = m_1m_2 mod n,
D (E (m_2, r_2) ^{m1} mod n^2) = m_1m_2 mod n

More generally, an encrypted plaintext raised to a constant k will decrypt to the product of the plaintext and the constant,

D (E (m_1, r_1) ^{k} mod n^2) = km1 mod n.

However, given the Paillier encryptions of two messages there is no known way to compute an encryption of the product of these messages without knowing the private key[9].

Second objective can be achieved through division of video into multiple adaptive frame buffers based on the bandwidth available by considering the snapshot of the network. Every buffer is partially encrypted [11]. This will reduce the encryption and decryption time as well as security cannot be violated. It is also used to recover original video back with maximum video quality [7].

To increase the security level, frame buffers are dispatched in random order that order is available to Video Service Provider. Internally at cloud server this random order is digitally signed and sends to the service user. For cross checking that user has to send the signature to video service provider. Now video service provider verifies both of signatures. If both signatures are matched then he will replied with correct order of video frames to the service user system to recover the video [8]. This process is only applied to highly sensitive videos. Remaining all videos can be either send without encryption or with less encryption.

V. RESULTS

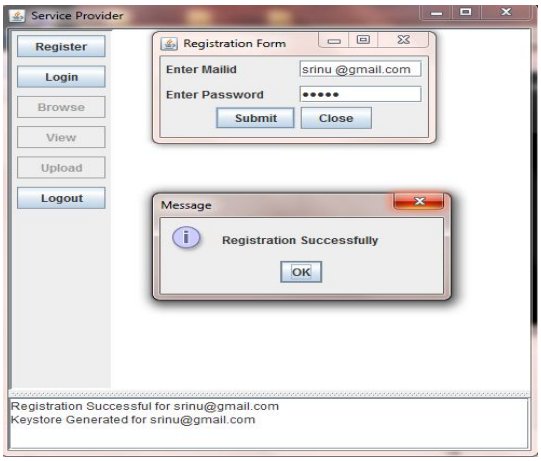

Fig 2: Service Provider Registration

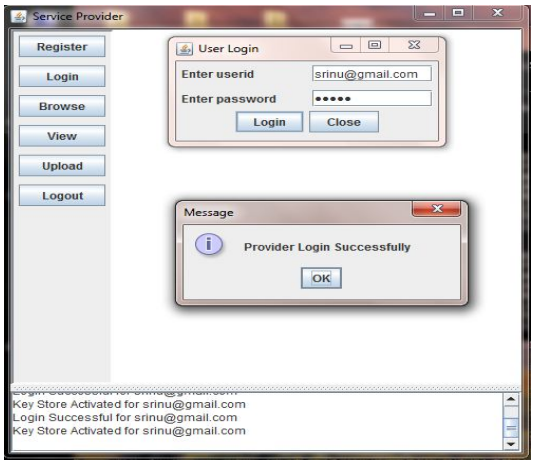

Fig 3: Service Provider Login

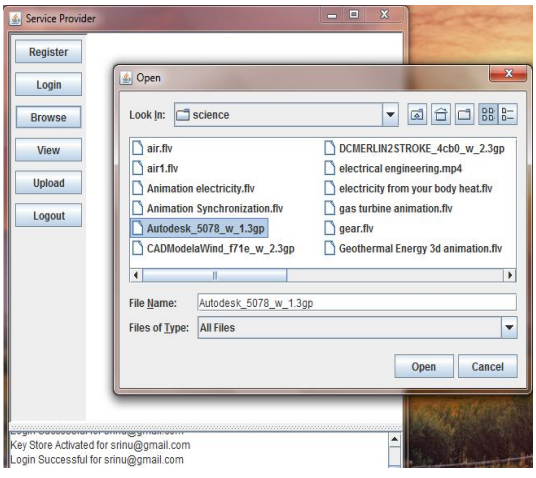

Fig 4: Service Provider browsing the data

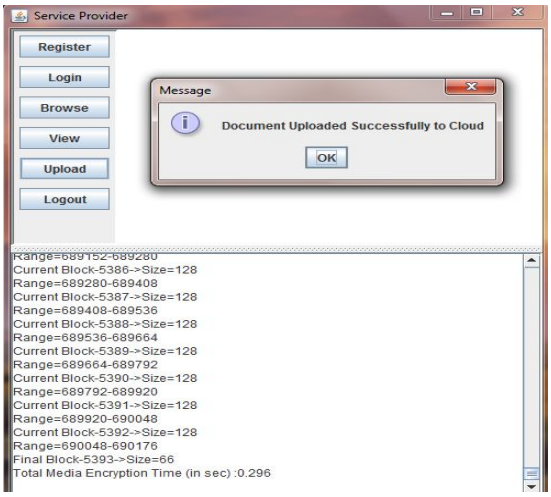

Figure 5: Service Provider uploading data to cloud

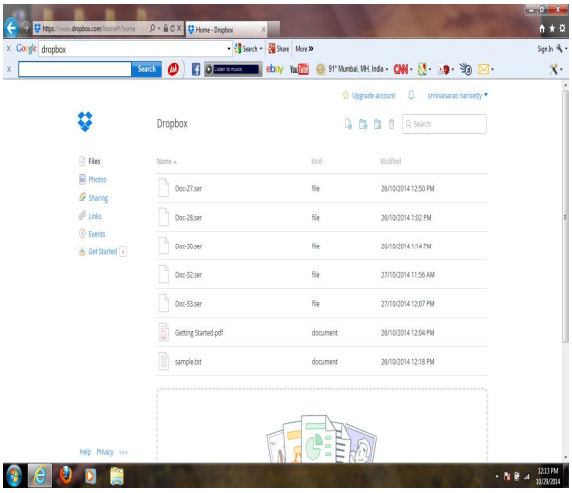

Figure 6: Data is securely stored in the cloud

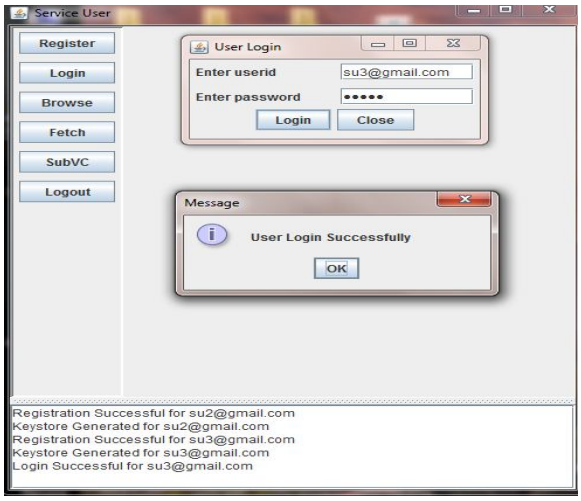

Figure 7: Service User Login

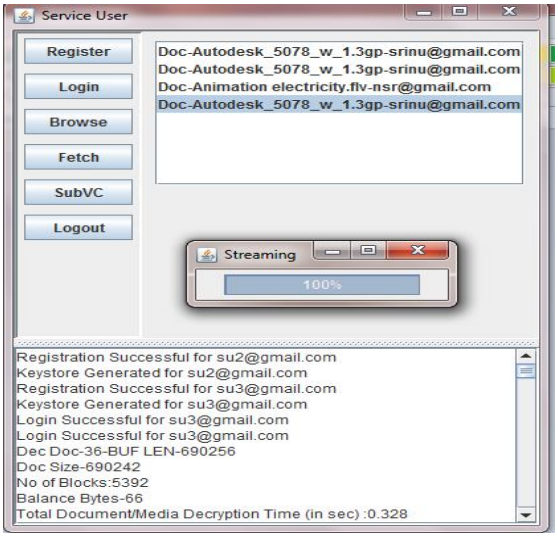

Figure 8: Service User Fetch/download data

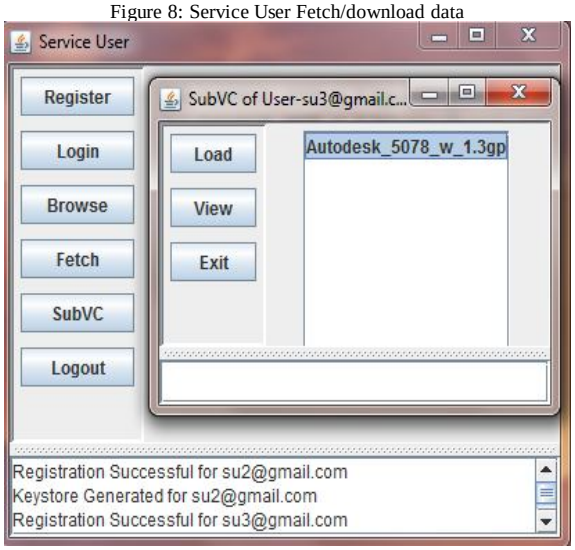

Figure 9: Every Service User is having his own Sub user database

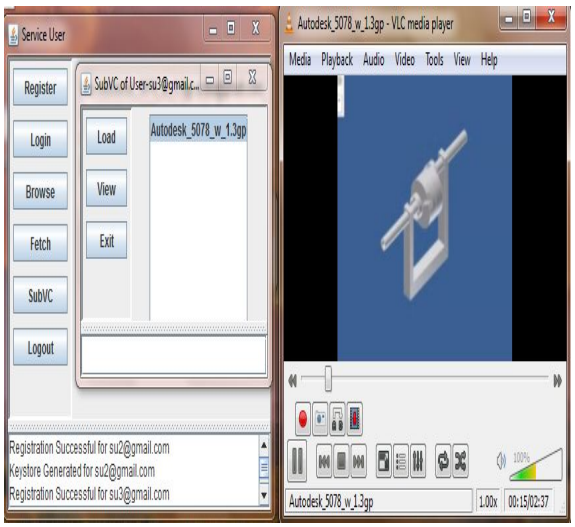

Figure 10: Sub User Database to load and View the Contents Available

VI. Conclusion & Future work

Finally it can be concluded that, the proposed method is used to secure the multimedia data in the cloud environment with homomorphic crypto system along with good buffering techniques by dividing the multimedia data in to number of blocks/chunks. Homomorphic encryption is one of the best cryptographic techniques to secure the multimedia data in the cloud, which is used to decrypt the data without knowing private key. In future it is possible to implement the same application for any real time business logics.

References

[1] Homomorphic tallying with pailliers crypto system, E-voting system, sancar choinyambuu-MSC student 12-6-2009.

[2] Multimedia Guardian Service for Multimedia Streams in the Public Cloud, N. Srinivasa Rao, S.Rama Sree, S.N.S.V.S.C Ramesh, sept-2014.

[3] P. Paillier, ``Public-key crypto systems based on composite degree residuosity classes, " in Advances in Cryptology_EUROCRYPT 1999.

[4] Homomorphic Encryption Applied to the Cloud Computing Security, Maha TEBAA, Saïd EL HAJJI, Abdellatif EL GHAZI-2012.

[5] The Paillier Cryptosystem, A Look Into The Cryptosystem and Its Potential Application By Michael O ' Keeffe the College of New Jersey Mathematics Department April 18, 2008.

[6] Homomorphic Encryption-based Secure SIFT for Privacy-Preserving Feature Extraction ChaoYung Hsu, Chun-Shien Lu, Soo-Chang Pei.

[7] Z. Huang, C. Mei, L. E. Li, and T. Woo, "Cloud Stream: Delivering high- quality streaming videos through a cloud-based SVC proxy," in Proc. IEEE INFOCOM Mini-conf., 2011, pp. 201–205.

[8] M. Wien, R. Cazoulat, A. Graffunder, A. Hutter, and P. Amon,"Real-time system for adaptive video streaming based on SVC," IEEE Trans. Circuits Syst. Video Technol., vol. 17, no. 9, pp.1227–1237.

[9] Ronald L. Rivest, Leonard Adleman, and Michael L. Dertouzos. On Data Banks and Privacy Homomorphisms, chapter On Data Banks and Privacy Homomorphisms, pages 169-180. Academic Press, 1978.

[10] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF formulas on ciphertexts. In Theory of Cryptography Conference, TCC'2005, volume 3378 of Lecture Notes in Computer Science, pages 325-341. Springer, 2005.

[11] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, 469-472, 1985.

Author's profile

Mr. N.SrinivasaRao is a student of Aditya Engineering College, Surampalem. Presently he is pursuing his M.Tech [CSE] from this college and he is received his B.Tech degree from Eswar College of Engineering, Affiliated to JNTUK Kakinada University in the year 2012. His area of interest includes Web Technologies and Object oriented Programming Languages, all current trends and technologies in Computer Science.

Mrs. S.Rama Sree obtained her B.Tech. Degree in Computer Science & Engineering from Koneru Lakshmaiah College of Engineering, affiliated to Jawaharlal Nehru Technological University, Kakinada in the year 2001 and M.Tech Degree in Computer Science from Jawaharlal Nehru Technological University Kakinada in the year 2006. She is currently a Research Scholar and working as a Head of the Department of Computer Science & Engineering at Aditya Engineering College, Surampalem, India. She has 18 International Journal Papers and 5 National/ International Conferences to her credit. Her Research interests include Software Engineering, Cost Estimation, Fuzzy Logic, Neural Networks and Neuro Fuzzy Systems.