Light Weight Encryption Technique for Group Communication in Cloud Computing Environment

Dinesh C. Verma
NCICS, Israna (Panipat)

A.K. Mohapatra
IGIT, New Delhi

Kaleem Usmani
Head of CERT-MU, Mauritius

ABSTRACT
Cloud computing is a typical example of distributed computing and emerged as a new paradigm that moves computing and data away from desktop and portable PCs into large data centers. Cloud services have three broad categories based on the fundamental nature of the cloud-based solution they provide: infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), or software-as-aservice (SaaS). In this work we have focused on the SaaS services provided by the Cloud service providers (CSP). The issue with SaaS is data security and confidentiality that makes the cloud user reluctant towards the cloud services. Data confidentiality can be achieved by encrypted outsourced content before outsourcing to cloud servers. But due to excessive computation of existing cryptographic algorithms and distributed nature of cloud computing, there is a need of a light weight cryptographic technique that has less computational overhead and high throughput. In this paper a light weight encryption technique is proposed which has less computational time and overall good performance. In order to prove it, the proposed algorithm is compared with the existing encryption techniques and results are analyzed. The key distribution of the shared key and secret key between the two group members is also handled efficiently using the same algorithm. Enormous overhead due to the large key size has been effectively ruled out in this paper. Light weight nature of proposed algorithm is well suited for distributed nature of cloud servers for an efficient processing with greatly enhanced user's confidence in cloud computing.

General Terms
Information Security, Distributed Computing.

Keywords
Distributed computing, Cloud computing, Encryption, Software as a service (SaaS), Data security.

1. INTRODUCTION
Cloud computing is emerging as a key computing platform for sharing resources that include software, infrastructure, application, and business process. Cloud computing has recently emerged as a new paradigm for hosting and delivering services over the Internet. With this emerging trend a organisation can start with small resources and increase only when there is a increase in service demand [1] [20]. Thus, a cloud model promotes availability and is composed of five essential characteristics as follows [2] [3] [4]:

- On-demand self-service
- Ubiquitous network access
- Location-independent resource pooling
- Rapid expansion
- Metered service

Cloud is designed to be available everywhere, all the time. By using the feature like geo-replication and redundancy, services are available even during hardware failures including full data centre failures [12].

Cloud services are offers three types of deployment models which are as follows [9]:

Public Cloud: Public clouds or hosted clouds are external or publicly available environments that are accessible to multiple tenants.

Private Cloud: Private clouds are typically custom-made with dedicated virtualized resources for the particular organizations. It also refers to internal data centres, not available to general public.

Hybrid Cloud: Hybrid clouds are the combination of both public as well as private cloud and tailored for a particular group of customers.

Although there's no well defined service, several IT experts have classified cloud computing vendors into three broad categories based on the fundamental nature of the cloud-based solution they provide: infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), or software-as-aservice (SaaS) [13][7].

Software-as-a-Service (SaaS): In this type of a cloud computing model, a provider's made software runs on a hardware cloud infrastructure and it is allowed to access by the customers with the help of a thin client interface such as a Web browser. Some examples are Customer Relationship Management (CRM) software, Salesforce.com's human-resource applications.

Plateform-as-a-Service (PaaS): PaaS allows customers to use programming environments of service provider to access and utilize additional application building blocks.

Infrastructure-as-a-Service (IaaS): This type of service is very useful for small enterprises that are not in position to invest for infrastructure. When a vendor rents out infrastructure components on demand—such as servers, storage components, file systems, virtualization technologies, and network hardware—the vendor is delivering an IaaS service.

The aim of this work is to focus on SaaS services in which the owner's data is under the control of some third party cloud service provider(CSP). In order to make it secure and confidential a mechanism is proposed here that provide storage of encrypted data on remotely located servers with maximum possible security and with acceptable performance

in group communication. In order to protect the access to data, we propose to combine both the symmetric and the asymmetric cryptography. The main contribution of the paper is the specification of such a technique. In order to implement it, we choose one time pad encryption technique in which text will be encrypted through a secret key and encrypting the secret key through the symmetric key. The key idea of the framework is that it combines symmetric and asymmetric data encryption to ensure data confidentiality and privacy while mini-mizing the cost for data shipping and computation. The proposed technique and its implementation guidelines specified in this paper can put into practice in while storing data in cloud environment. We are also demonstrating the feasibility of the proposed approach.

The remainder of the paper is organized as follows: after a short introduction of background concepts in section II and III, we describe, from a high level point of view, the need of data security in cloud environment in section IV. Relate work will be discussed in section V and proposed technique described in section VI. An initial performance evaluation is given in VII and possible future work is given in section VIII.

2. CLOUD STORAGE TECHNIQUES

Cloud computing is a typical example of distributed computing paradigm where the data are stored on cloud servers. In cloud data storage, a user stores his data through a Cloud Service Provider (CSP) into a set of cloud servers, which are running in a simultaneous, cooperated and distributed manner. Instead of storing information to computer's hard drive or other local storage device, data owner saves it to a remote database. Cloud storage has several advantages over traditional data storage. For example, data stored on a cloud storage system, user can get that data from any location that has Internet access. No need to setup a new infrastructure to cater the increasing demand of a user. It brings convenience to the user at the same time poses threat to the privacy of data. A common approach to data protection is to encourage users to store their encrypted data on servers. However, as the amount of encrypted data in the cloud grows, retrieval becomes problematic. One component of cloud infrastructure is the data center, which provides safe, reliable data services. Searching over the distributed nature of data is not an easy task and requires special attention. It may be stored in multiple computers located in the same physical location, or may be dispersed over a network of interconnected computers. In such a system, responsibility for data management is delegated to the distributed file system such as NFS, Netware, LAN-Manager, and AFS (Andrew File System) and its operational staff.

3. ARCHITECTURE OF A CRYPTOGRAPHIC STORAGE SERVICE

To ensure the privacy and confidentiality of sensitive data from, a user herself may encrypt the sensitive data before uploading the data into cloud data storage [11]. In order to store user's encrypted data, cryptographic storage required in a cloud environment. The advantage of cryptographic storage is that no unauthorized users would be able to access the data until some kind of permission is granted by the owner of data [2, 24].

Cryptographic cloud architecture is illustrated in Figure 1. In the given architecture three parties are considered: Data owner Alice that store his data in the cloud; a group member Bob with whom Alice wants to share her data; and a cloud storage provider that stores Alice's data.

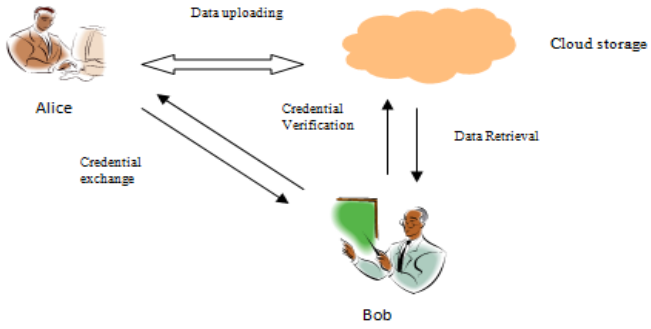

Figure 1: Cloud cryptographic architecture

The architecture consists of three components: a Data Encryption (DE), that process and encrypt data before it is sent to the cloud; a Credential Exchange mechanism (CE), that generates credentials and enable various parties to retrieve segments of customer data (these credentials will enable the parties to decrypt encrypted files according to the policy); Credential verification and data retrieval (CV), that enable various parties to download the files from cloud providers.

Cryptographic storage is an important aspect of cloud services for building trust in cloud vendors. Secure encryption and distributed data make documents random, unreadable and difficult to search by users of cloud [27].

4. DATA SECURITY

Data security is the core of cloud computing security problems. Data security is mainly about the data confidentiality, integrity, availability and so on. As cloud computing brings with it new deployment and associated adversarial models and vulnerabilities, it is imperative that security takes center stage [1]. IDC conducted a survey of 244 IT executives about cloud services. As figure 2 shows, security concerns are the number one issue facing cloud computing.

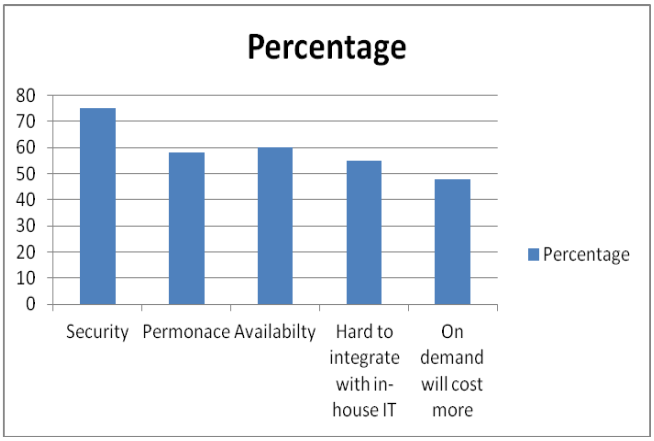

Figure 2: IDC's finding on cloud issues [2]

This is especially true as cloud computing services are being used for e-commerce applications, medical record services, and back-office business applications, all of which require strong confidentiality guarantees with secure and efficient retrieval mechanism. The infrastructure provider, in this context, must achieve the following objectives [20] [21]:

(1) Confidentiality: A significant barrier to the adoption of cloud services is the users' fear of confidential data

(particularly financial and health data) leakage and loss of privacy in the cloud. Confidentiality is usually achieved using cryptographic techniques. It becomes very difficult for malicious user as well as the curious servers to tamper an encrypted data

(2) Auditability: Auditability is a mechanism to confirmation of security setting of applications and like attestation of whether it has been tampered or not. Auditability can be achieved using remote attestation techniques.

In this paper the main focus in on confidentiality of data and it alone can be achieved by encrypting the outsourced content before outsourcing to potentially access curious servers [8]. However, the encrypted data cannot be easily processed by servers [23]. In particular improper and excessive encryption of data limits the data uses with increased read, write and other data base operation times [16].

5. RELATED WORK

Data security is stated in terms of Service level agreements (SLAs), that are an industry standard approach for controlling risk, and so is a more natural starting point [22]. SLA is assurance about the security of data but merely documents assurances are not sufficient to achieve high level security. In order to achieve high level security encryption techniques are practiced.

To keep user data confidential against an untrusted Cloud Service Provider (CSP), a natural way is to apply cryptographic approaches, by disclosing the data decryption key only to authorized users. However, such a simple encryption techniques are not efficient in terms of processing and may introduce following problems:

- It depletes too much CPU capability and memory power of the client during the encryption and decryption.
- The CSP cannot determine which emails or files contain keywords specified by a user if the encryption is not searchable, and can only return all the encrypted emails [18].

Generally, a legitimate client has only limited bandwidth, CPU, and memory, therefore improper and excessive encryption of the data degrades the performance of the cloud application [5]. So there is a need of encryption technique which supports fast encryption and decryption process.

Qin Liu et al. investigate the characteristics of cloud storage services and propose a secure and privacy preserving keyword searching (SPKS) scheme, which allows the CSP to participate in the decipherment, and to return only files containing certain keywords specified by the users, so as to reduce both the computational and communication overhead in decryption for users, on the condition of preserving user data privacy and user querying privacy [18].

Hassan Takabi , James B.D. Joshi and Gail-Joon highlighted the major issues in a cloud environment and also suggested the possible approaches to tackle these issues. They worked out on the multi - tenancy nature of cloud computing and given stress on sharing responsibility between provider and customer.

Ji Hu, Andreas Klein also observed penalty caused by excessive encryption on data and proved that there will be increase in write and read time and also the space requirement to store encrypted data [7].

Kai Hwang et al. [10] proposed data coloring and software marking technique to protect shared data objects and massively distributed software modules. This method enables single sign on cloud and tightens access control to access control for sensitive data in public cloud. The computational complexity of this method is much lower than that performed in conventional encryption and decryption calculations in PKI services.

Mark Townsend [14] proposed that emphasized on multistep approach for data security in cloud environment. Relying on security policy alone will not ensure data security in any environment. In most cases, to ensure data security, utilizing a multi-step approach is considered best. For instance encrypting data, both while it's stored on a cloud vendor's servers and being transmitted to end users, mitigate some of the privacy risks associated with accidental or malicious exposure of the information.

Kevin Hamlen et al. [15] used a secure co-processor (SCP) and Hadoop distributed file system (HDFS) as part of the cloud infrastructure to enable efficient encrypted and large amount of sensitive data storage. By embedding a secure co-processor (SCP) into the cloud infrastructure, the system can handle encrypted data efficiently. Basically, SCP is a tamper-resistant hardware capable of limited general-purpose computation. One drawback of SMC is the high computational costs, heat dissipation.

Li Xiong et al. [26] proposed a framework that use data partitioning, encryption, and data reduction to ensure data confidentiality and privacy while minimizing the cost for data shipping and computation. Fragmentation consists of partitioning data records (horizontal partitioning) or attributes (vertical partitioning) in subsets such that only records or attributes in the same fragment are visible together.

Diaa Salama Abd Elminaam et al. [25] also observed that larger key size causes an increase in power and time consumption. In case of AES, three different key sizes possible i.e., 128-bit, 192-bit and 256-bit keys. In case of AES it can be seen that higher key size leads to clear change in the battery and time consumption. It can be seen that going from 128-bit key to 192-bit causes an increase in power and time consumption about 8% and in 256-bit key causes an increase of 16%.

From the literature review it is clear that data security is the main issue of cloud computing. An extensive work is being carried out to develop an encryption technique having less encryption and decryption time with enough security in distributed environment.

6. PROPOSED WORK

A lightweight encryption technique that is highly secure and less computational cost is proposed in this section. Proposed encryption also consume less power due it's smaller key size. The proposed algorithm has the following components:

1. Secret key generation: The first step is to generate a random number which is used as a secret key to encrypt the text. This is one time key because the same key will not be used again for encryption purpose.
2. Encryption: The given file will be encrypted before uploading to a third party service provider using the secret key generated in step 1. The secret key is also encrypted using the shared key and inserted in the encrypted text.
3. Data uploading: Encrypted data (Text and secret key) is now can be uploaded to the remote servers. Encrypted data in the control of service provider are now safe from the service providers and malicious users.
4. Credential exchange and decryption: By exchanging the credentials, only legitimate users can download and decrypt the desired file the remote servers. For exchanging the credentials any standard method like Difie-Hellman can be used in advance.

This whole process is divided into three different algorithms: (1) Secret key generation and encryption of text (2) encryption of a secret key and merging with encrypted text (3)

decryption of text using a secret key. The flow diagram of proposed encryption technique is described in figure 3 and figure 4. The approach follows the one time pad technique where the secret key generated by a random number generator. In order to make it more secure secret key is sent through the encrypted file itself.

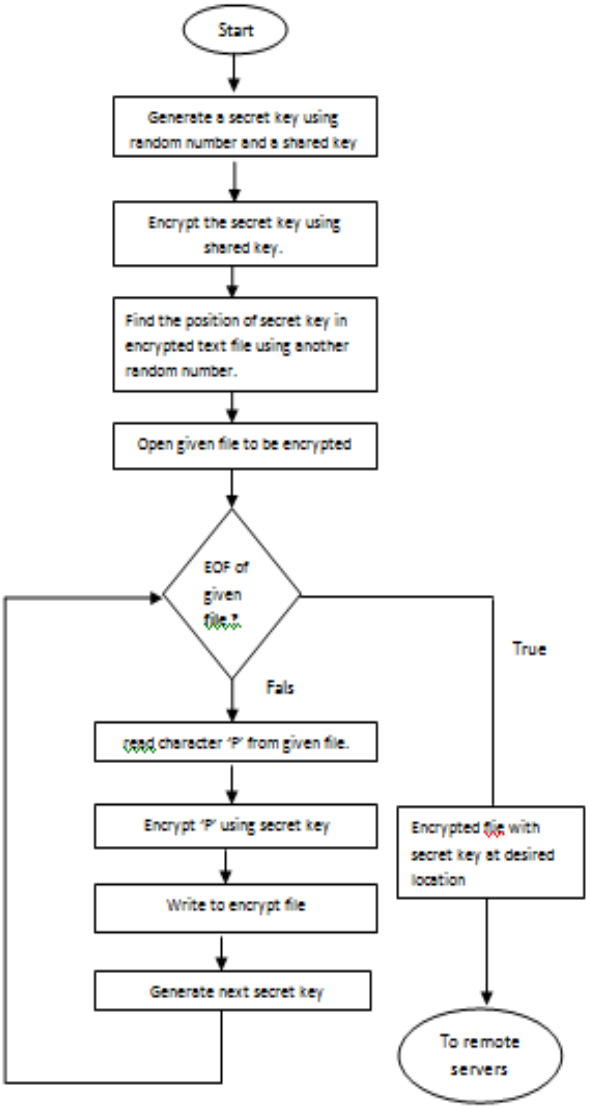

Figure 3: Encryption and file uploading process

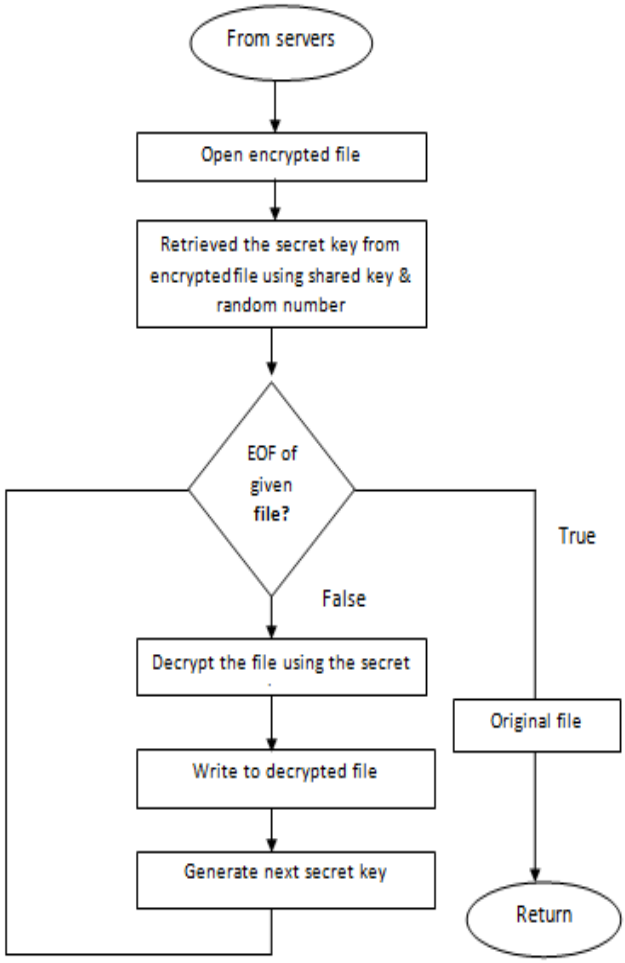

Figure 4: Decryption of encrypted file downloaded from the cloud server

Secret key generation and encryption Algorithm (E):

// Shared key K_{pu} is known in advance.

Step 1. Generate a secret key K_{pr} using random number.
Step 2. Generate another random number *'N'*.
Step 3. Determine location to store secret key in encrypted data file using formula

Loc = |shared key(K_{pu}) – reverse of shared key ($K^{'}_{pu}$)| + 'N'.

Step 4. Convert '*N*' in to its binary equivalent 'B_n'.
Step 5. Repeat steps 6 and 9 till EOF.
Step 6. Read a character P_i from the plain text file.
Step 7. Encrypt it and store its cipher text (C_p) value in the encrypted data file.

$$C_p = E(K_{pr}(P_i)) \qquad \text{where i= 1,2…n}$$

Step 8. Encrypt secret key using a shared key.

$$C_k = E(K_{pu}(K_{pr}))$$

Step 9. Increment secret key by one.

Step 9. Write encrypted secret key at position Loc in the encrypted data file.
Step 10. Insert binary of *'N'* in the starting of encrypted file.
Step 11. Upload this whole encrypted file to remote server.

$$C = B_n\ Cp \ldots\ Cp\ Ck\ Cp\ \ldots . Cp$$

Key retrieval and Decryption Algorithm (D):

Step 1. Read 4 characters (B_n) from encrypted data file and convert into decimal number say *'N'*.
Step 2. Determine location of the secret key in encrypted data file using formula

$$Loc = |shared\ key(K_{pu}) - reverse\ of\ shared\ key\ (K'_{pu})| + 'N'$$

Step 3. Set file pointer=Loc.
Step 4. Repeat step 5 for i=1 to 4.
Step 5. Read 9 characters and decrypts them by shared key.

K_{pr} =D(K_{pu} (C_k)) //secret key has been determined.

Step 6. Set file pointer at 5th position. //First 4 positions are occupied by random number.
Step 7. Repeat steps 8 and 9 till EOF of the encrypted data file.
Step 8. If (file pointer != Loc) then
Read 9 characters one by one and decrypts it by secret key.
$P_i = D(K_{pr}(C_P))$
else
file pointer= Loc+36

Step 9. Concatenation of decrypted characters gives plain text *(P)*.

Algorithm to encrypt a character 'P_i' and secret key.

// Secret key varies for each character.

Step 1. Determine ASCII value of a given character *'P_i'*.
$A_i = ascii('P_i')$
Step 2. Convert ASCII value into 8 bit binary code.
$P_{bin} = tobinary(A_i)$
Step 3. Reverse the 8-bit ASCII value.
P_{rev} = reverse(P_{bin})
Step 4. Generate a random secret key.
$K_{pr} = randomize()$
Step 5. Divide 8-bit ASCII value by 4-bit secret key to determine 4-bit remainder and 5-bit quotient.
$P_{rev}\ mod\ K_{pr}$
$Q_i = quotient$
$R_i = remainder$
Step 6. Concatenation of the remainder and quotient is 9-bit encrypted code corresponding to a character *'P_i'*.

$$C = \sum_{i=1}^{n} R_i\ Q_i$$

7. PERFORMANCE EVALUATION

The strength of any encryption technique is its security, integrity and authenticity. In the proposed method random bit string acts as the key. The resulting ciphertext cannot be broken, because every possible plaintext is an equally probable candidate. The ciphertext gives the cryptanalyst no information at all. In a sufficiently large sample of ciphertext, each letter will occur equally open. The secret key is also encrypted with the help of shared key and given algorithm i.e. $C_k = E(K_{pu}(K_{pr}))$ and transmitted along with ciphertext that is very difficult to distinguish. Location of secret key i.e. $Loc = |shared\ key(K_{pu}) - reverse\ of\ shared\ key\ (K'_{pu})| + 'N'$ in the ciphertext is also changed for every file which is very difficult to find in the ciphertext.
The proposed method is highly authentic because no one other than the authorized user can use this algorithm to encrypt the plain text. From the given algorithm it is clear that to encrypt a secret key i.e. $C_k = E(K_{pu}(K_{pr}))$ one need know public key, encryption algorithm. Similarly for decryption K_{pr} =D(K_{pu} (C_k)) one must be the authenticated user of cloud service provider (CSP) and familiar the shared key and decryption algorithm. In the absence of proper credential like public key, encryption algorithm, secret key, decryption algorithm and the authentication process by the CSP one cannot decrypt the message. Merely knowing one algorithm, it is not possible to complete the encryption and decryption process. So the proposed algorithm is authentic in all sense.
The integrity of the proposed system is related to the authentication of the proposed system. It is not easy to tamper the by any malicious user or even by the remote servers. Data is uploaded in encrypted form in the premises of CSP, so any change in encrypted data will not make any sense. Further only an authorized person is allowed to upload and download the file, so the tampering with data is not an easy task.
For our experiment, we use a laptop i3-350M processor 2.26 GHz, in which performance data is collected. The overall performance of the proposed algorithm is compared with the existing encryption techniques. Several parameters like 1) Encryption time 2) throughput 3) Decryption time are considered. Comparison of proposed and existing algorithms is shown in figure 5, figure 6 and figure 7.

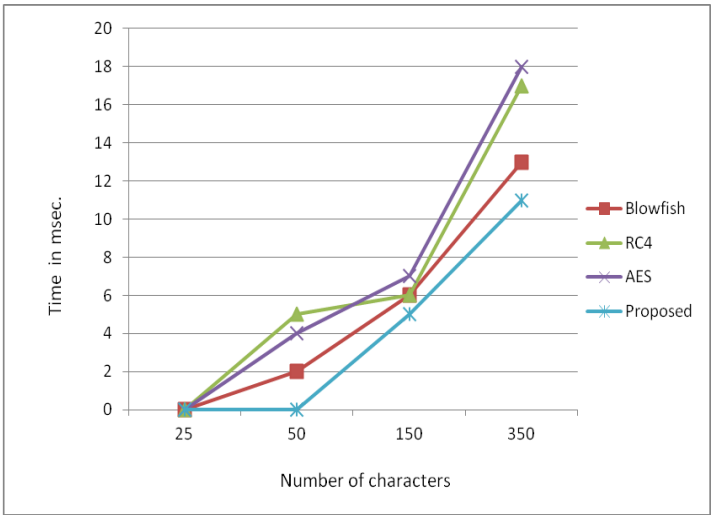

Figure 5: Comparison of encryption time

Form literature survey it is clear that the Blowfish algorithm is superior in term of encryption time, decryption and throughput. But from the experimental result it is clear that the proposed method is taking a bit less time in comparison of Blowfish. We conducted this experiment and we find that the proposed method is better option due it is simplicity and security. In figure 6 throughputs of all these algorithms has

been plotted and observed that the proposed algorithm has better throughput as compared to other algorithms.

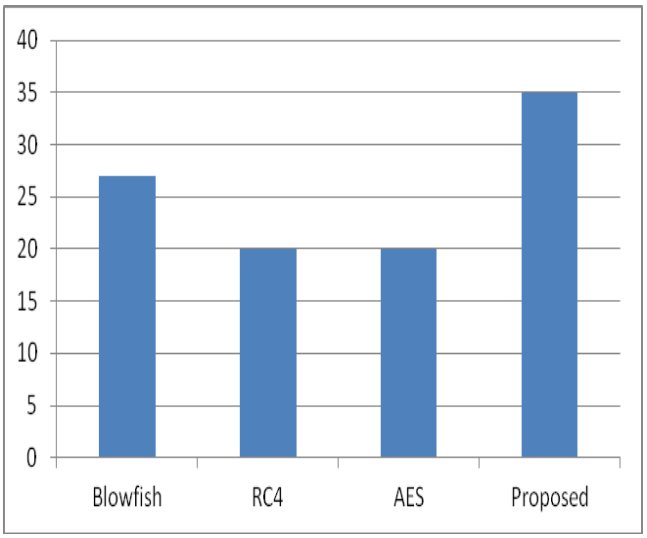

Figure 6: Comparison of throughput

Comparison of decryption time of algorithms is shown in the figure which shows that the proposed algorithm is as good as blowfish algorithm.

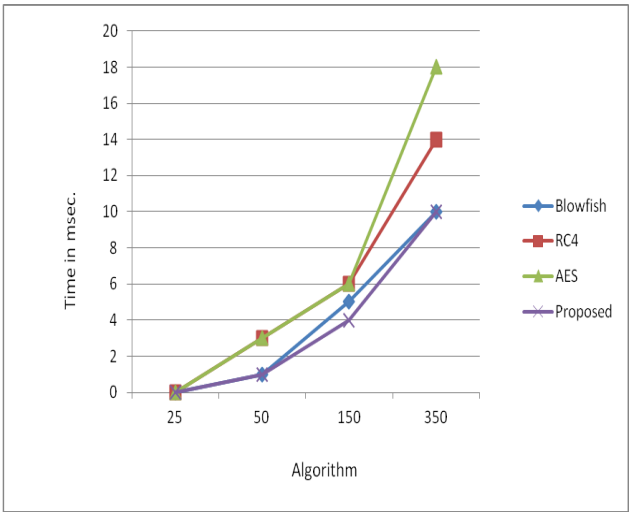

Figure 7: Comparison of decryption time

Figure 7 shows the visual representation of different encryption algorithm's key sizes. It has to be noted that our algorithm occupies the lower position in the key size. Though the key size is less, it provides an equivalent security to the information in a distributed environment. Our method provides an equal challenge as other encryption algorithms with less number of bits used for the secret key. The proposed encryption technique is less computational and provides higher speed in a secure manner.

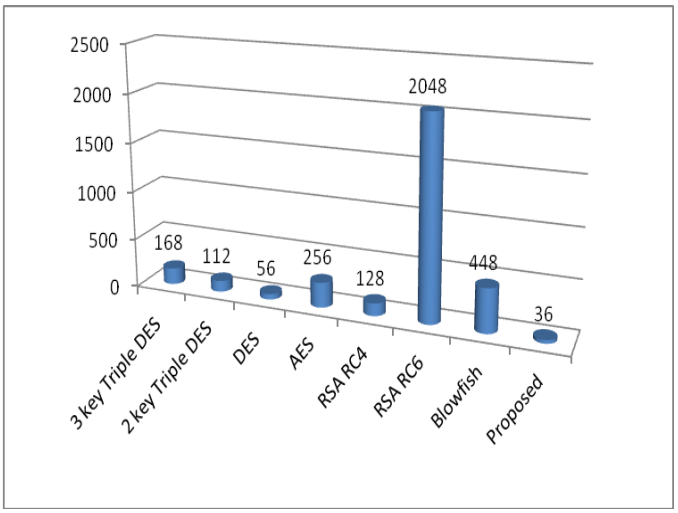

Figure 7: Comparison of key size

On the basis of above result analysis we can say that the proposed algorithm is lightweight in terms of encryption time, decryption time and throughput. Due to these properties our algorithm can be good choice for uploading of encrypted file on remote server and downloading of file from the servers.

8. CONCLUSION AND FUTURE WORK

In order to achieve the confidentiality, encryption techniques are used in a cloud environment. The overall performance of existing encryption techniques for a small amount of data is comparatively less due to long key size and high degree of calculations. Our method provides an equal challenge as other encryption algorithms with less number of bits used for the secret key. The proposed encryption technique is less computational and provides higher speed in a secure manner. Our enciphering approach also provides a high level of authentication between the end users with only a small key size, which is fixed. The key management for the sharing of the secret key between the two parties is also efficiently handled using the same algorithm. The enormous overhead due to the large key size has been effectively ruled out in this paper. Due to a small key size and gained knowledge from literature survey the power consumption very less as compared to existing algorithms.. Various projections on the performance of our algorithm have also been discussed in this paper. The distributed nature of cloud computing is a major hurdle in the construction of an efficient encryption technique with searching capability. In the future, we will focus to extend this method to encrypt the images and to provide the searching capability for encrypted data. This system can be further optimized by considering the processing power of co-processor in a distributed environment and by combining with other existing encryption algorithms for exchanging their keys.

9. REFERENCES

[1] Mehmet Yildiz et. al., "A Layered Security Approach for Cloud Computing Infrastructure", 10th IEEE International Symposium on Pervasive Systems, Algorithms, and Networks, pp 763-767.

[2] Anthony T. Velte, Toby J. Velte Robert Elsenpeter "Cloud Computing A Practical Approach" Tata McGRAW-HILL EDITION, pp 35.

[3] Ik Rae Jeong et al., "Constructing PEKS schemes secure against keyword guessing attacks is possible?", Elsevier Computer Communications 32 (2009) 394–396.

[4] Bo Zhang et al., "An efficient public key encryption with conjunctive-subset keywords search", Elsevier Journal of Network and Computer Applications 34 (2011) 262–267.

[5] Ji Hu, Andreas Klein, "A Benchmark of Transparent Data Encryption for Migration of Web Applications in the Cloud", 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, IEEE Computer Spciety, pp 735-740.

[6] G. Russello et al., "providing data confidentiality against malicious hosts in shared data spaces", Elsevire Science of computer programming 75(2010) 426-439

[7] S. Subashini, V.Kavitha, "A survey on security issues in service delivery models of cloud computing", Elsevier Journal of Network and Computer Applications 34, 2011, pp 1-11

[8] Weichao Wang et. al., "Secure and Efficient Access to Outsourced Data", CCSW'09, November 13, 2009, ACM, pp. 55-64.

[9] L. Grossman, "The case for cloud computing", IEEE Journal computer.org/ITPro, March/April 2009, pp 23-27

[10] Kai Hwang, Deyi Li, "Trusted Cloud Computing with Secure Resources and Data Coloring "IEEE INTERNET COMPUTING 2010, Pg 14-22

[11] Patrick McDaniel, Sean W. Smith, "Outlook: Cloudy with a Chance of Security Challenges and Improvements" IEEE computer and reliability societies, January/February 2010, pp 77-80

[12] Shyam Kumar Doddavula, Amit Wasudeo Gawande, "SETLabs briefings on business innovation through technology", Infosys Research, Vol 7 No.7, pp 14.

[13] Michael Armbrust, Armando Fox et. al., "A View of Cloud Computing", communications of the ACM April 2010, vol. 53 no. 4, pp 50-58.

[14] Mark Townsend, "Managing a Security Program in a Cloud Computing Environment", ACM InfoSecCD '09, September 25-26, 2009, pg 128-133

[15] Kevin Hamlen et al., " Security Issues for Cloud Computing", International Journal of Information Security and Privacy, 4(2), 39-51, April-June 2010

[16] Richard Chow at. Al., "Controlling Data in the Cloud: Outsourcing Computation without Outsourcing Control", CCSW'09, ACM, November 13, 2009, pp 85-90

[17] William Stallings,"Cyptography and Network Security: Principal and Practices", Forth Edition, Pearsons Prentice Hall.

[18] Qin Liu et al., "Secure and privacy preserving keyword searching for cloud storage services" Journal of Network and Computer Applications, Elsevier, 2011.

[19] Seny Kamara and Kristin Lauter, "Cryptographic Cloud Storage", LNCS 6054, Springer, 2010, pp. 136-149.

[20] Qi Zhang, Lu Cheng, Raouf Boutaba," Cloud computing: state-of-the-art and research challenges", J Internet Serv Appl (2010) 1, Springer, pp 7–18

[21] Dan Lin, Anna Squicciarini, "Data Protection Models for Service Provisioning in the Cloud", SACMAT'10, June 9–11, 2010, ACM, pp 183-192.

[22] S. Creese et al., "Data Protection-Aware Design for Cloud Services", CloudCom 2009, Springer LNCS 5931, pp. 119–130, F009.

[23] Yao Chen, Radu Sion, "On Securing Untrusted Clouds with Cryptography", WPES'10, October 4, 2010, Chicago, ACM, pp 109-114.

[24] Christian Henrich et el., "Brief Announcement: Towards Secure Cloud Computing" Springer, SSS 2009, LNCS 5873, pp. 785–786, 2009.

[25] Diaa Salama Abd Elminaam et al., "Evaluating The Performance of Symmetric Encryption Algorithms", International Journal of Network Security, Vol.10, No.3, PP.216–222, May 2010.

[26] Li Xiong et al., "Adaptive, Secure, and Scalable Distributed Data Outsourcing: A Vision Paper ", ACM 3DAPAS'11, June , 2011

[27] Jin Wook Byun et al., "On a security model of conjunctive keyword search over encrypted relational database", The Journal of Systems and Software 84 (2011) 1364–1372.

[28] Dong Yuan et al., "A data placement strategy in scientific cloud workflows", Elsevier Future Generation Computer Systems 26 (2010) 1200-1214.