ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

Cost-Efficient Multi-Cloud Storage and Data Hosting with Cloud Computing Security Using AES Algorithm

Mitash Sule[1], Prof. Dipmala Salunke[2], Mahesh Upase[3], Jayesh Jhamtani[4], Vedvrat Singh[5]

Department of Information Technology, Rajarshi Shahu College of Engineering, Tathawade, Savitribai Phule Pune University, Pune, India. [1, 2,3,4,5]

ABSTRACT: Recent years have witnessed a rapid development in online data hosting service. Facing the numerous cloud vendors as well as their heterogeneous pricing policies, customers may well be perplexed with which cloud(s) are suitable for storing their data. Also, cloud security is a major concern for customers storing data in cloud. This paper proposes and implements a cost-effective data hosting scheme while providing an appropriate redundancy strategy in multi-cloud. Also, we enhance the cloud computing security using AES algorithm.

KEYWORDS : Multi-cloud, data hosting, cloud storage, cryptography, AES

1. INTRODUCTION

Nowadays, progressively more and more enterprises and organizations are hosting their data into the cloud in order to reduce IT maintenance cost and enhance the data reliability [1], [2], [3]. Cloud Computing enables convenient on-demand network access to shared resources. It provides customers with reliable, scalable and low-cost data hosting functionality. Subscribers have to pay cloud service providers for exactly how much they have used. Existing clouds exhibit heterogeneities in terms of both working performance and pricing policies. Different cloud vendors have distinct pricing policies and charging items. Such diversity leads to observable variations across cloud vendors [4]. Facing numerous cloud vendors as well as their heterogeneous pricing policies, customers may be perplexed with which cloud(s) are suitable for storing their data. The current situation is that customers usually put their data into a single cloud and simply trust to luck. This is liable to the vendor lock-in risk, because customers would be confronted with a dilemma if they want to switch to other cloud vendors. The vendor lock-in risk first lies in that data migration inevitably generates considerable expense [5]. Nirvanix, which has thousands of customers including top 500 companies, suddenly shut down its cloud storage service in September 2013. So clearly, it is unwise for an enterprise or an organization to host all of their data into a single cloud [6], [7]. Also, uncontrolled data availability is another type of vendor lock-in risk. Almost all major cloud vendors experienced service outages in recent years [8], [9], [10]. Cloud computing security includes various numbers of issues like multi-tenancy, data loss and leakage, easy accessibility of cloud, internal threats etc. It is not easy to accomplish all the security measures that meet the security needs of all the users, because different customers may have different security demands based upon their objective of using the cloud services [11], [12]. Multi-cloud data hosting has received vast attention from researchers, customers and startups. The key principle of multi-cloud data hosting is to distribute data across numerous clouds to gain enhanced redundancy.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

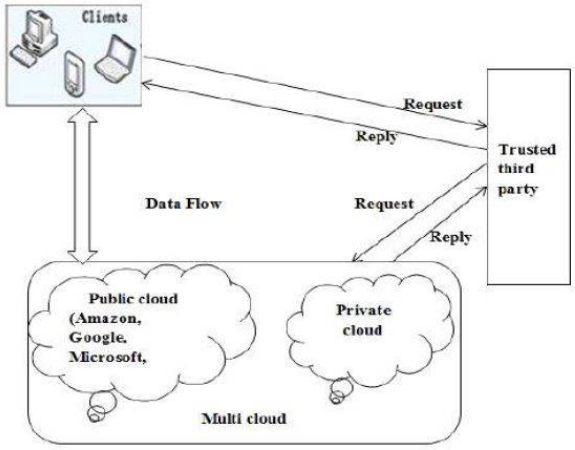

Figure 1:Multi-Cloud Architecture

In multi-cloud architecture, a data storage system involves three different entities as shown in Figure 1. Client is an entity either individual customer or organization which uses cloud to save large amount of data and depend on cloud for conservation and assessing of data. Cloud service providers have significant storage and computation resources to manage client data and offer storage utility to the client. In multi-cloud, service providers organize their resources and provide data storage service to the client. Trusted Third Party is an entity to which verification parameters are stored [13].

II. PROPOSED WORK

In this paper, we propose and implement a novel, cost-efficient data hosting scheme for heterogeneous multi-cloud environments. The system accommodates different pricing strategies and availability requirements. It searches for suitable clouds and suitable redundancy strategy to store data with minimized monetary cost and guaranteed availability. We also implement a flexible migration scheme for the system. It helps user to migrate data among different clouds if necessary.We also implement enhanced security in cloud computing using AES algorithm. Customers' data can be made secure in the cloud using encryption.

III. SYSTEM ARCHITECTURE

In this part, we elaborate a cost-efficient multi-cloud storage and data hosting scheme with cloud computing security using AES algorithm.There are three main components in the system: Searching module, Data Hosting and Security Implementation module and Migration module.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

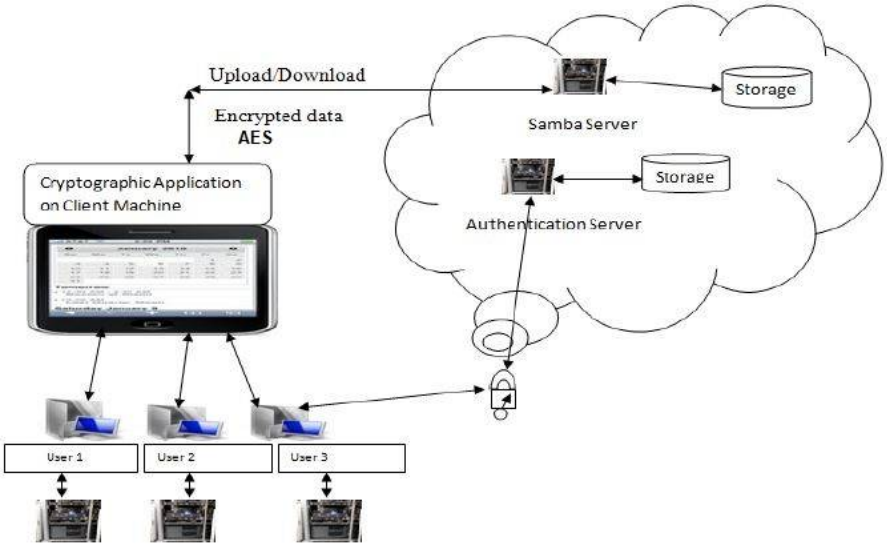

Figure 2: Cloud computing security using AES algorithm

Searching module searches for suitable clouds based on monetary cost and availability in cloud. Data Hosting and Security Implementation module hosts data into cloud and provides security by implementing AES algorithm. Migration module transfers data into another cloud to save the monetary cost of customer.

3.1. SEARCHING TECHNIQUE FOR SELECTING SUITABLE CLOUD TO STORE DATA

A greedy algorithm is method that follows the problem solving heuristic of making the locally best choice at several stages with the prospect of finding a global optimum. In many problems, a greedy strategy does not in general produce best solution, but nonetheless a greedy heuristic may yield locally best solutions that approximate a global excellent solution in a reasonable time.

This algorithm searches for suitable clouds to store data with minimized monetary cost and guaranteed availability. It provides the customer to search for best cloud plan from heterogeneous clouds and their different pricing policies.

3.2 DATA HOSTING AND SECURITY IMPLEMENTATION IN CLOUD USING AES ALGORITHM

Data hosting stores data in cloud and implements AES algorithm to secure the data. The plain text data is never written in any place on cloud. This includes every type of data. This encryption solution is transparent to the application and can be integrated instantaneously and easily without any application changes at all. The key is never stored next to the encrypted data, considering it may compromise the key also. To store the keys, a physical key management server can be installed in the customer's premises. This encryption protects data and encryption keys and guarantees they remain under customer's control, and are never exposed in storage or in transit.

AES, it is a block cipher with a block length of 128 bits. It allows three different key lengths: 128, 192, or 256 bits. We advise AES with 128 bit key length. The encryption technique consists of 10 rounds of processing for 128-bit keys. Except for the final round in every case, all other rounds are similar. 16 byte encryption key, in the form of 4-byte words is extended into a key schedule consisting of 4 x 4 4-byte words. The 4 x 4 matrix of bytes formed from 128-bit input block is mentioned as the state array. Before either round-based processing for encryption can initiated, input state is XORed with the first four words of the schedule.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

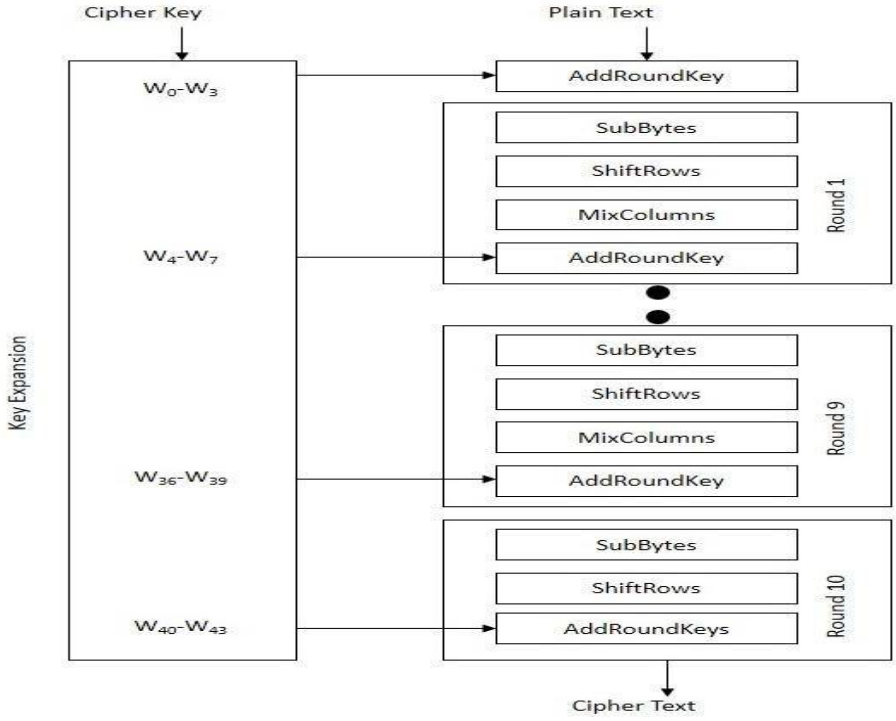

Figure 3: AES Encryption

For encryption, every round consists of the succeeding four steps:
SubBytes – a non-linear substitution step where every byte is replaced with substitute according to a lookup table (S-box).
ShiftRows – a transposition step where every row of the state is shuffled cyclically a certain number of times
MixColumns – a mixing operation which performs on the columns of the state, joining the four bytes in each column.
AddRoundKey – every byte of the state is joined with the round key; every round key is derived from the cipher key using a key schedule.
For the final round only three steps are performed: SubBytes, ShiftRow and AddRoundKey.

3.2.1 SubBytes

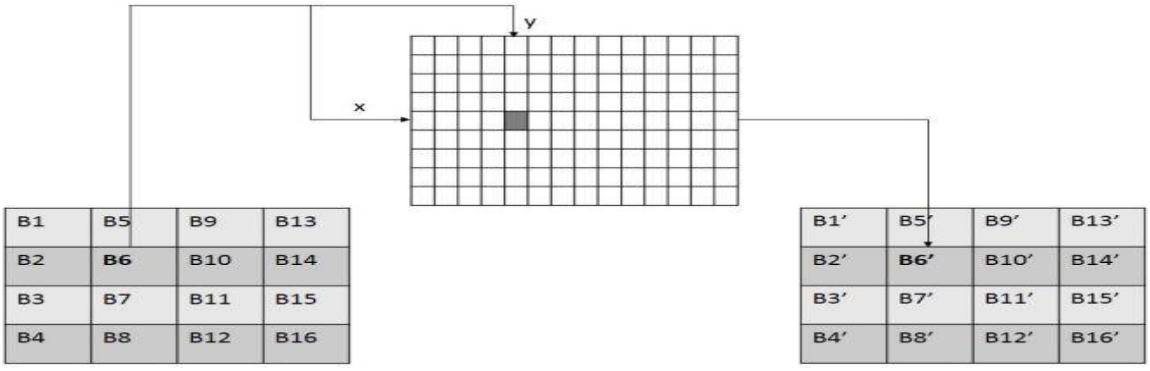

Figure 4: SubBytes Transformation Step

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

The goal of this step is to give sufficient resistance from differential and linear cryptanalysis attacks. This is byte-by-byte substitution where every byte is substituted independently using Substitution table (S-box). Each input byte is divided into 24-bit patterns, depicting an integer value between 0 and 15 which can then be described as hexadecimal values. Left digit specifies the row index and right digit specifies the column index of S-box. At the intersection of row and column, value inferred is substituted. There are sixteen explicit byte-by-byte substitutions. S-box is constructed by a mixture of GF (28) arithmetic and bit mangling.

3.2.2 ShiftRows

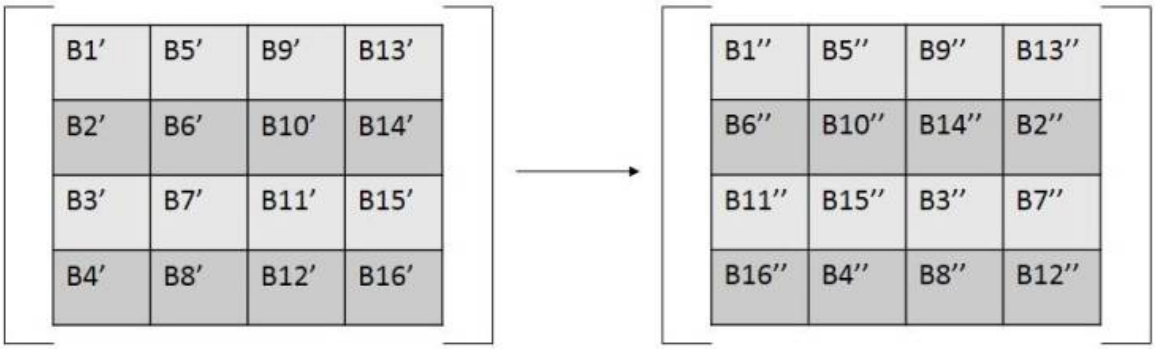

Figure 5: ShiftRows Transformation Step

The purpose of this step is to provide diffusion of the bits over multiple rounds. The row 0 in the matrix is not shifted, row one is circular left shifted by one byte, row two is circular left shifted by two bytes, and row three is circular left shifted by three bytes.

3.2.3MixColumns

Figure 6: MixColumns Transformation Key

Likepreceding step, the goal of this step is to contribute diffusion of the bits over numerous rounds. This is achieved by carrying out multiplication one column at a time. Every value in the column is multiplied against every row value of a standard matrix. The outcome of these multiplications isXORed together. For e.g. value of first byte B1'' is multiplied with 02, 03, 01 and 01 and XORed to acquire new B1''' of resulting matrix. The multiplication continues against one matrix row at a time against every value of a state column. Same steps are used in decryption, as in encryption, the procedure in which the steps are carried out is different.

3.2.4 AddRoundKey

In this operation, the matrix is XORed along with the round key. The primitive key consists of 128 bits/16 bytes which is represented as a 4x4 matrix. This 4 words key where every word is of 4 bytes is changed to a 43 words key. The first four words exhibit W[0], W[1], W[2], and W[3] The rest of expanded key i.e. W[4] to W[43] is produced as follows:-

```
for (i=4; i
{
T = W [i-1];
```

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

if (i mod 4 = = 0)
T = Substitute (Rotate (T)) XOR RConstant [i/4];
W[i] = W [i-4] XOR T;
}
Here
Rotate means - implement a one byte left circular rotation on the 4-byte word.
Substitute means - implement a byte substitution for every byte of the word, utilizing S-box, also used in the SubBytes step.
RConstant means – Round Constant which is XORed with the bytes. The rightmost three bytes of the round constant are nil.
In this way, W [4]… W [43] of the key schedule is produced from the initial 4 words. Although, overall, the similar steps are used in decryption, as in encryption, the sequence in which the steps are executed is different.

3.3 MIGRATION OF DATA

Cloud migration involves shifting data between cloud environments, which is known as cloud-to-cloud migration. The process of shifting to a different cloud provider is known as cloud service migration.
Given the usable clouds including their prices and availability, customer should be able to migrate their data from existing cloud storage to new cloud storage. Migration can save more money. We have to make sure that the cost of migration can be earned back by the new storage. Therefore, the following circumstance has to be met:
M [Se] > M [Sn] + T ,where M [Se] and M [Sn] are the monetary cost of existing cloud storage and monetary cost of new cloud storage respectively. And, T represents the migration cost. Hence, we migrate only if the above condition is met.

IV. PERFORMANCE AND EVALUATION OF SYSTEM

To study performance of our proposal, we chose a text file with the size of 1 MB. It shows the encoding and uploading time versus the number of data fragments set by the customer. From the Figure 7, we can see that when we increment the number of data fragments from 2 to 8, the uploading time decreases significantly; while the encoding time has slightly increased. The considerable performance improvement for uploading is due to the help of multi-threading technique; while the increased number of data fragments along with more checksum pieces results in increased overhead for encoding. However, when the number of data fragments n is further increased, the uploading time and the total processing time become relatively stable. This result is very different from our previous findings, where the uploading time dramatically goes up when the number of data fragments $n \geq 10$. Based on our additional investigation, we notice that the former result is due to a sudden performance decrease at Amazon S3 when the file size of a data fragment becomes less than 16 MB. Note that when n = 10, the size of each data fragment equals 1MB/10, i.e., 0.1MB. With the default configuration, the Transfer Manager from the Amazon S3 API uses a single low-speed thread for uploading files of less than 16 MB. In this study, we modify the configuration to allow multi-part upload of small files. As a result, the sudden performance decrease (when $n \geq 10$) disappeared. As shown in Figure 7, the total processing time becomes constantly less than one minute. Comparing to the optimum performance for uploading the 1 MB file, where Dropbox used 2 minutes 19 seconds, our approach demonstrates a significant performance improvement for uploading the file.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

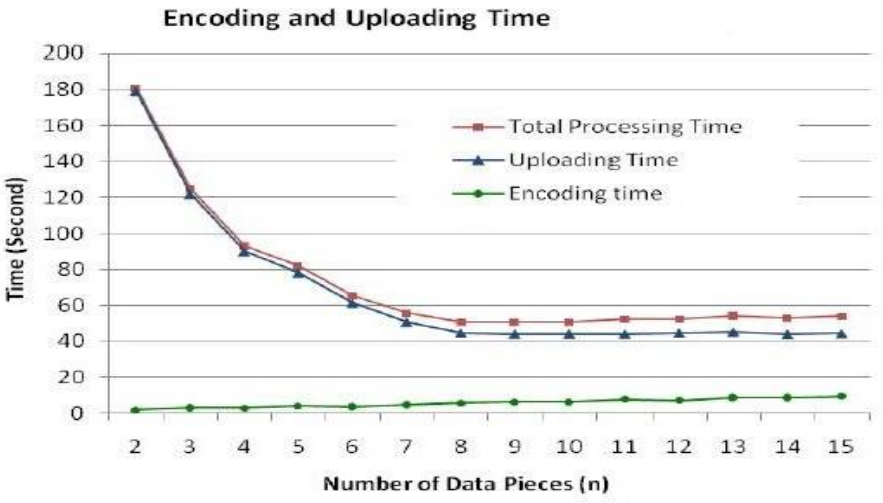

Figure 7. Encoding & uploading time vs. number of data pieces (1MB file)

From the above mentioned experimental results, we can notice that both the uploading and downloading time can be considerably reduced by selecting a reasonable number of data fragments. For example, when the file size is about 1 MB, based on our experiments, the number of data fragments should typically be set to 8 as long as the network condition is excellent, and the client machine has identical performance as the one used in the case study. Note that, when n = 8 and m = 4, the space efficiency e reaches its highest value 0.6667. It is also vital noting that when no service provider fails, the application only requires downloading the data fragments, and no checksum pieces are needed for restoring the original file. In this case, the downloading time can be further reduced, and the decoding time becomes merely the time needed to combine the data fragments into the original file. Therefore, in a usual case with no failures of service providers, the total performance for file retrieval could be better than the outcome demonstrated in Figure 7.

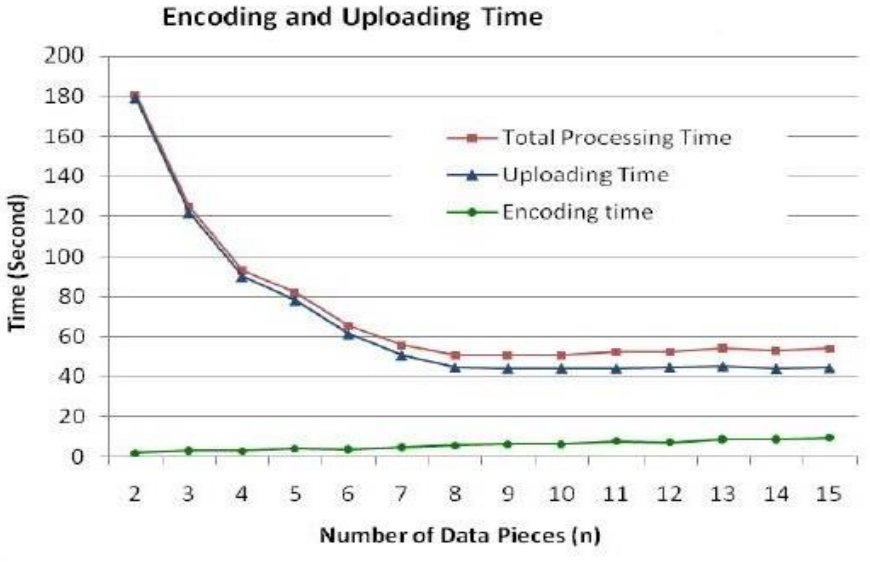

Figure. 8. Downloading and decoding time vs. number of data pieces (1 MB file)

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

To additionally investigate the system performance for larger files, we use video file with a file size of 3 MB for this case study. Figure 8 displays the encoding and uploading time vs. the number of data fragments set by the customer. From the Figure 8, we can see that when we increase the number of data fragments from 2 to 8, the uploading time decreases considerably; while the encoding time has slightly increased. When $n > 8$ the uploading time is decreased and the encoding time is increased by some extent. As shown in the Figure 8, for $n \geq 8$, the total processing time is continually below 100 seconds, i.e., 1 minute and 40 seconds. Comparing to the optimum performance for uploading the 1 MB file, where Dropbox used 4 minutes and 13 seconds to upload the file, our proposal again demonstrates a significant performance improvement for uploading the file.

V. CONCLUSION

Cloudservices are undergoing rapid development and the services based on multi-cloud also become prevailing. Major concerns while moving data into cloud is capital expenditure, availability and security. So, in this paper, we design a system which guides customers to distribute data among clouds cost-effectively. It provides customer with the ability of secured storage under an affordable budget. Implementing AES for security over data provides benefits of less memory consumption and less computation time. AES provides security to cloud users as encrypted data in the cloud is safe from many attacks. No differential and linear cryptanalysis attacks have been yet proved on AES.

REFERENCES

[1] Gartner: Top 10 cloud storage providers. [Online]. Available: http://www.networkworld.com/news/2013/010313-gartner- cloud-storage-265459.html?page=1, 2013.
[2] Z. Li, C. Jin, T. Xu, C. Wilson, Y. Liu, L. Cheng, Y. Liu, Y. Dai, and Z.-L. Zhang, "Towards network-level efficiency for cloud storage services," in Proc. ACM SIGCOMM Internet Meas. Conf., 2014, pp. 115–128.
[3] Z. Li, C. Wilson, Z. Jiang, Y. Liu, B. Y. Zhao, C. Jin, Z.-L. Zhang, and Y. Dai, "Efficient batched synchronization in dropbox-like cloud storage services," in Proc.ACM/IFIP/USENIX 14th Int. Mid- dleware Conf., 2013, pp. 307–327.
[4] A. Li, X. Yang, S. Kandula, and M. Zhang, "CloudCmp: Compar- ing public cloud providers," in Proc. ACM SIGCOMM Internet Meas. Conf., 2010, pp. 1–14.
[5] Quanlu Zhang, Shenglong Li, Zhenhua Li, "CHARM: A Cost-Efficient Multi-Cloud Data Hosting Scheme with High Availability" in IEEE TRANSACTIONS ON CLOUD COMPUTING, VOL. 3, NO. 3, JULY-SEPTEMBER 2015.
[6] Its Official, the Nirvanix Cloud Storage Service is Shutting Down. [Online]. Available: http://techcrunch.com/2013/09/27/its-offi- cial-the-nirvanix-cloud-storage-service-is-shutting-down/, 2013.
[7] Nirvanix Provides Cautionary Tale for Cloud Storage. [Online]. Available: http://www.forbes.com/sites/tomcoughlin/2013/09/ 30/nirvanix-provides-cautionary-tail-for-cloud-storage/, 2013.
[8] Google Outages Damage Cloud Credibility.[Online]. Available: https://www.networkworld.com/news/2009/092409-google- outages-damage-cloud.html, 2009.
[9] Rackspace to issue as much as $3.5M in customer credits after out- age. [Online]. Available: http://www.networkworld.com/news/ 2009/070609-rackspace-outage.html, 2009.
[10] Summary of the amazon EC2 and amazon RDS service disruption in the US east region. [Online]. Available: http://aws.amazon.com/cn/message/65648/, 2011.
[11] AbhaSachdev and MohitBhansali "Enhancing Cloud Computing Security using AES Algorithm" in International Journal of Computer Applications (0975 – 8887) Volume 67– No.9, April 2013.
[12] SachdevAbhaThakral, and MohitBhansali. "Addressing the Cloud Computing Security Menace." IJRET, Volume 2, Issue 2, pp. 126-130, Feb 2013.
[13] "Survey on Cost-Efficient Multi-Cloud Storage and Data Hosting Scheme In Cloud Computing" inInternational Journal of Innovative Research in Computer and Communication EngineeringVol. 4, Issue 2, February 2016.