UTF Base 64 Methods Used in Securing Cloud Data Storage

O. A. Alabi* and I.U. Oghenekaro
Department of Computer Science
University of Port Harcourt, Nigeria

*Corresponding author's email: Akinity2000 [AT] yahoo.com

Abstract— **The different exemplifications here-in give a technique and framework to secure information stockpiling and sharing over a cloud based system, Disseminated information thusly builds the danger of unapproved physical access to the information, Loss of control to the outsourced information, the quantities of the general population with access to the information. who could be traded off significantly increments. The Protest Arranged Investigation and Outline (OOAD) technique is picked with a specific end goal to satisfy, all things considered, the objective of this examination on the grounds that the investigation is pointed on outlining and building up an enhance secure model for cloud information stockpiling, which addresses information stockpiling that is information very still and in movement. The technique includes introducing a customer application on a client gadget, validating a customer application client, removing content from an information source, acquiring content sharing data from a substance stockpiling supplier, sending a substance conveyance list and a substance use arrangement to an application server, encryption the substance by the customer application making and sharing a protected substance record, unscrambling the substance document, getting a refreshed substance use approach from the application server, confirmation instrument, checking the character of the substance beneficiary utilizing a personality determination component, rendering the safe substance record to the beneficiary, upholding the substance use strategy and sending content use logs to the application. To secure these data, we implement a mechanism called UTFbase 64 which combines of UTF base8 and UTF base 16 function as a mechanism that uses zero to secured it from unauthorized user (hacker). The result is extremely appreciative because of the two function combined as one to secured stored data**

Keywords- Cloud computing, UTF base 8,16,64, storage

I. INTRODUCTION

Distributed computing is a characteristic advancement of the far reaching reception of virtualization, benefit situated design, autonomic, and utility processing (Jeremy Geelan 2009). This new range of figuring is in help of making another level of utilizations that can keep running on an expansive number of equipment gadgets, which is comprehensive of portable or PDAs and tablets or Individual Advanced Associates (PDAs), when information is put away on the cloud.

Distributed computing is the blend of an innovation, stage that gives facilitating and capacity benefit on the Web (Abhinay B et al 2013). Distributed computing plans to give adaptable and modest on - request figuring foundations with great nature of administration levels. Distributed computing manages actualizing building principals to get excellent applications by means of the Web. Principle objective of the distributed computing is to give adaptable and reasonable on-request registering frameworks with great nature of administration levels (Harjit S and Gurdev S. 2012). It is an outcome and impact of the straightforward entry to remote figuring locales gave by the Web. This every now and again appears as online instruments or applications that clients can approach and make brimming with it by means of a web program in a way that the client will have an inclination that the program is introduced in their host frameworks.

II. MATERIALS AND METHODS

How the algorithm is used and its link

This project work is based on secure cloud data for proper storage, a hash service data integrity verification, encryption / decryption service, and provision for defining list of people which can access data securely is provided by a trusted 3rd party which is separate from the storage cloud provider. We proposed the used of UTF base64 algorithms which combined UTF base8 and UTF base16. Base64 is a gathering of comparative twofold to-content encoding plans

that speak to parallel information in an ASCII string position by making an interpretation of it into a radix-64 portrayal. The term Base64 starts from a particular Emulate content exchange encoding which is represented as follows in the table

Length	**Input**	**Length**	**Output**	**Padding**
20	*any carnal plea**sure.***	28	YW55IGNhcm5hbCBwbGVh**c3VyZS4=**	1
19	*any carnal plea**sure***	28	YW55IGNhcm5hbCBwbGVh**c3VyZQ==**	2
18	*any carnal plea**sur***	24	YW55IGNhcm5hbCBwbGVh**c3Vy**	0
17	*any carnal plea**su***	24	YW55IGNhcm5hbCBwbGVh**c3U=**	1
16	*any carnal plea**s***	24	YW55IGNhcm5hbCBwbGVh**cw==**	2

The same characters will be encoded differently depending on their position within the three-octet group which is encoded to produce the four characters. For example

Input	**Output**
*plea**sure.***	cGxlY**XN1cmUu**
*lea**sure.***	bGVh**c3VyZS4=**
*ea**sure.***	ZWF**zdXJlLg==**
asure.	Y**XN1cmUu**
sure.	**c3VyZS4=**

The proportion of yield bytes to include bytes is 4:3 (33% overhead). In particular, given a contribution of n bytes, the yield will be bytes long, including cushioning characters.

In principle, the cushioning character isn't required for unraveling, since the quantity of missing bytes can be ascertained from the quantity of Base64 digits. In a few executions, the cushioning character is obligatory, while for others it isn't utilized. One case in which cushioning characters are required is linking numerous Base64 encoded records.

Interpreting Base64 with cushioning

When disentangling Base64 content, four characters are commonly changed over back to three bytes. The main exemptions are when cushioning characters exist. A solitary '=' demonstrates that the four characters will interpret to just two bytes, while '==' shows that the four characters will unravel to just a solitary byte. For instance:

Encoded	**Padding**	**Length**	**Decoded**
YW55IGNhcm5hbCBwbGVh**cw==**	two '='s	1	*any carnal plea**s***
YW55IGNhcm5hbCBwbGVh**c3U=**	one '='	2	*any carnal plea**su***
YW55IGNhcm5hbCBwbGVh**c3Vy**	no '='s	3	*any carnal plea**sur***

Deciphering Base64 without cushioning

Without cushioning, after typical disentangling of four characters to three bytes again and again, under four encoded characters may remain. In this circumstance just a few characters might remain. A solitary outstanding encoded character isn't conceivable (in light of the fact that a solitary base 64 character just contains 6 bits, and 8 bits are required to make a byte, so at least 2 base 64 characters are required : the principal character contributes 6 bits, and the second character contributes its initial 2 bits) . For instance:

Length	**Encoded**	**Length**	**Decoded**
2	YW55IGNhcm5hbCBwbGVh**cw**	1	*any carnal plea**s***
3	YW55IGNhcm5hbCBwbGVh**c3U**	2	*any carnal plea**su***
4	YW55IGNhcm5hbCBwbGVh**c3Vy**	3	*any carnal plea**sur***

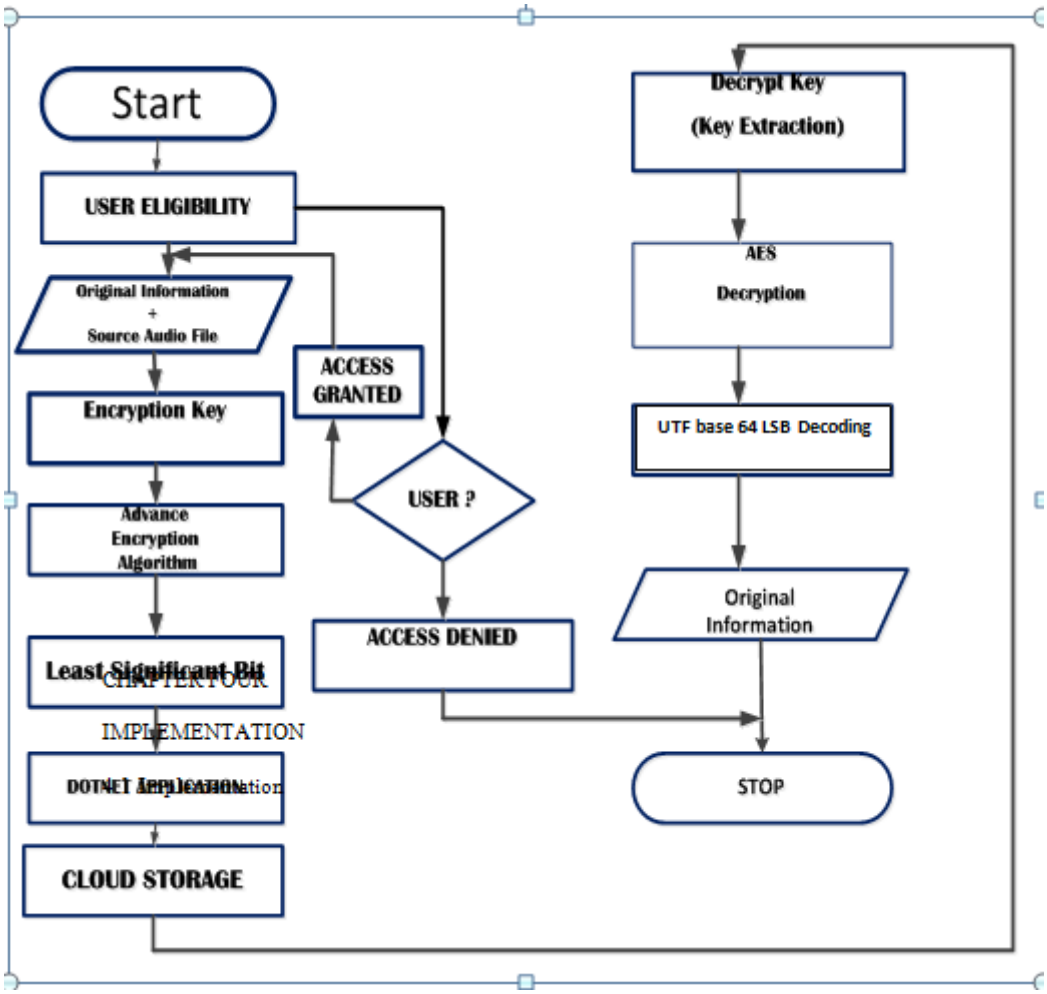
The above sample(tables) is what is utilized to explained and secured information in the cloud and influence the outsider to have full sure on the Information put away in the cloud.

Curtmola et al. [2011] meant to guarantee information ownership of numerous imitations over the scattered stockpiling framework. They expanded the ("provable information ownership") PDP plan to cover numerous imitations without encoding every reproduction independently, giving certification that different duplicates of information are really kept up.

Geambasu, et al (2009): take a shot at Vanish Framework and Focused on the objective of making information that self destruct naturally after it is not any more valuable. It ought to do as such with no unequivocal activity by the clients or any gathering putting away or filing that information, such that all duplicates of the information vanish at the same time from all locales, on the web or disconnected.

Ekong and Ejiofor, 2016: chip away at pointless framework that secured put away information utilizing AES and DES as an encryption instrument. They additionally actualize the model such that it will identify unapproved part to get to put away information.

III. RESULT


Start
USER ELIGIBILITY
Original Information
+
Source Audio File
Encryption Key
Advance
Encryption
Algorithm
Least Significant Bit
IMPLEMENTATION
CLOUD STORAGE
ACCESS
GRANTED
USER ?
ACCESS DENIED
Decrypt Key
(Key Extraction)
AES
Decryption
UTF base 64 LSB Decoding
Original
Information
STOP

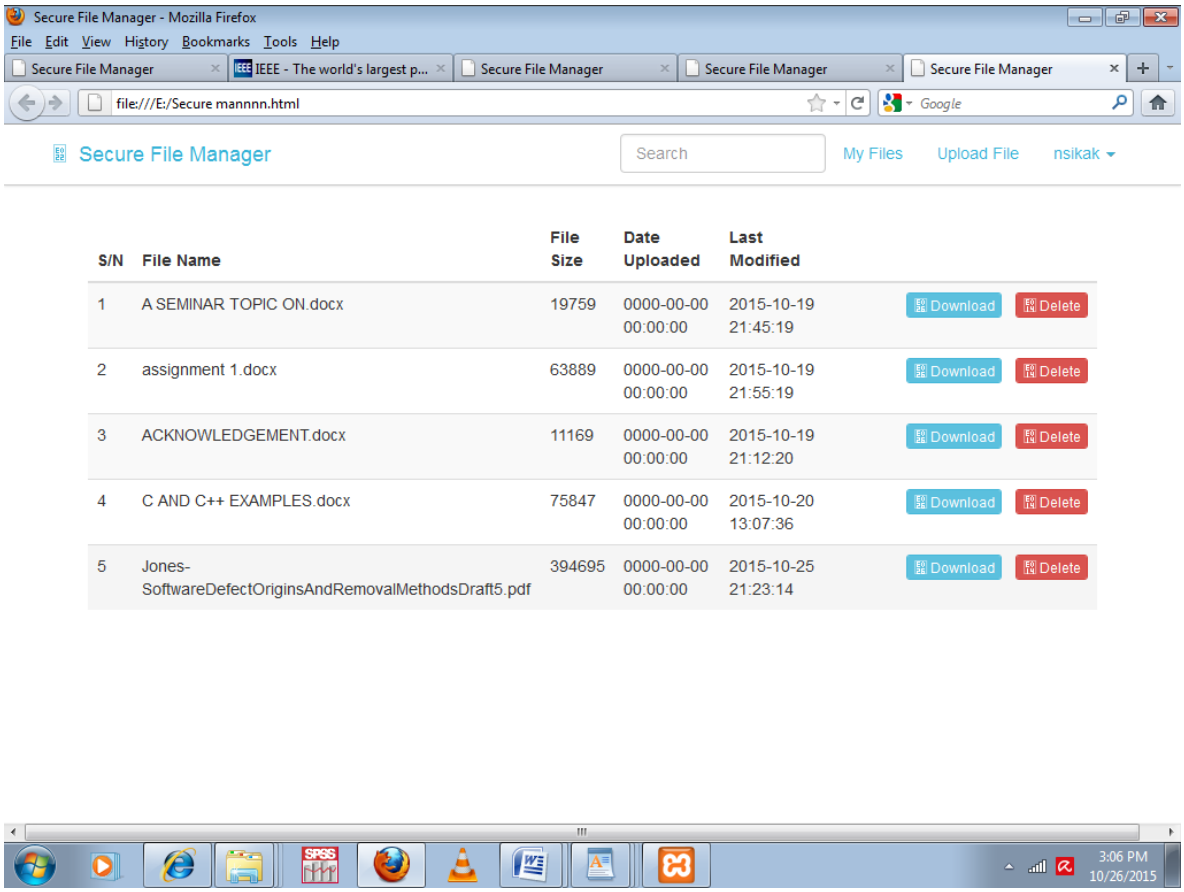


Figure 2: list of file uploaded and secured

Execution Assessment

The execution assessment of a product is measured with the lattices utilized. A few networks can be utilized as a part of the assessment of a product extending from speed, time, effectiveness et cetera. In UTF base64 calculation is utilized As a part of Securing Cloud Information Stockpiling framework, the grids utilized as a part of the assessment testing are the record estimate, the ideal opportunity for transfer and download, and the throughput of the propelled calculations utilized as a part of securing the information put away in the cloud. This area clarified in points of interest how the execution for the transfer time and download time for the proposed framework is quicker than the current framework and diverse keys are produced (both Symmetric and Unbalanced key are created to clients contingent upon the time you Enrolled as a part), which are appointed to enlisted Clients in view of time go (0;00hours-04.00hrs suggests symmetric, 04 and is more safe for clients to have self-assured on there information put away in the cloud.

IV. DISCUSSION

Result Table

Table 1: Performance evaluation for uploading

FILE SIZE	EXISTING	PROPOSED
16MB	40s	1s
64MB	60s	0s
124MB	70s	0s
500MB	80s	0s
1GIG	90s	5s

Fig 2.3 demonstrates the throughput for transfer. X hub speaks to the throughput, Y hub speaks to the record measure. Throughput is how much megabyte is computed in one moment. In existing framework, 16mb was transferred in 50s, where as in proposed framework it takes 10s to transfer a 16mb document. Contrast with the current framework the execution of proposed framework is higher. This diagram obviously demonstrates the proposed framework decreases the throughput over the current framework by a normal of 15.5% and up to 55% for the transferring.

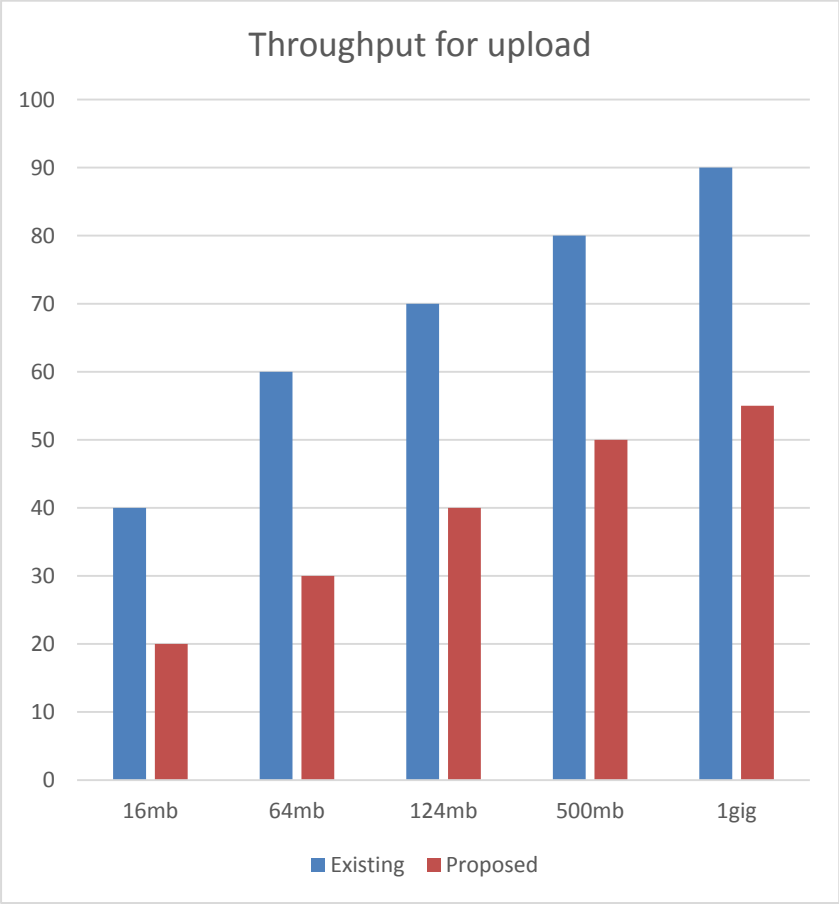


Fig 2.3Throughput for Upload

Result Table 2:

Performance evaluation for downloading

FILE SIZE	EXISTING	PROPOSED
16MB	47s	0s
64MB	59s	0s
164MB	73s	0s
500MB	86s	0s
1GIG	96s	0s

In Result, Fig.2.4 demonstrates the throughput for download. X hub speaks to the throughput, Y pivot speaks to the document measure. In existing framework, 16mb was downloaded in 50s, where as in proposed framework it takes 18s to download a 16mb record. Contrast with the current framework the execution of proposed framework is higher. It demonstrates that proposed framework lessens the throughput over the current framework by a normal of 18% and up to 50.75% for the downloading. The figures demonstrates the throughput comes about for the diverse plans. The throughput diminishes in light of the fact that .

transfer/download forms require significantly more CPU calculation and completing encryption/unscrambling forms in the proposed framework, contrasted and the current framework. Table1 demonstrates the execution assessment for transferring. In existing framework, 16mb was transferred in 50s, where as in proposed framework it takes 20s to transfer a 16mb record. Contrast with the current framework the execution of proposed framework is higher. Table 2 demonstrates the execution assessment for downloading. In existing framework, 16mb was transferred in 47s, where as in proposed framework it takes 25s to download a 16mb document. Contrast with the current framework the execution of proposed framework is higher.

Table1 demonstrates the execution assessment for transferring. In existing framework, 16mb was transferred in 40s, where as in proposed framework it takes 20s to transfer a 16mb document. Contrast with the current framework the execution of proposed framework is higher. Table 2 demonstrates the execution assessment for downloading. In existing framework, 16mb was transferred in 47s, where as in proposed framework it takes 25s to download a 16mb record. Contrast with the current framework the execution of proposed framework is higher

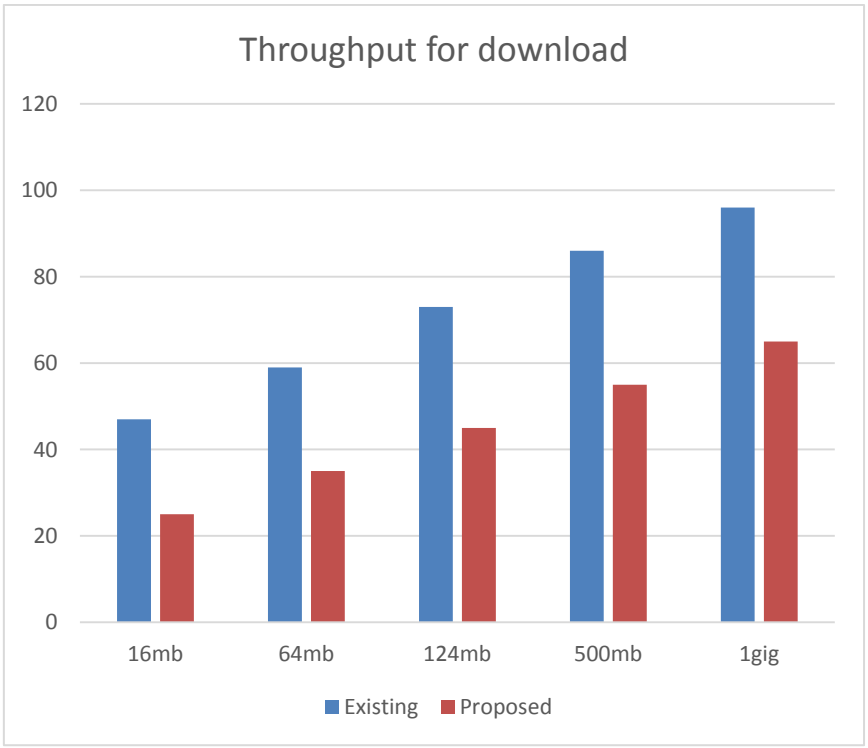


Fig 4.9.2 Throughput for Download

REFERENCES

- Abhinay B.Angadi, Akshata B.Angadi, Karuna C.Gull (2013): "Security Issues with Possible Solutions in Cloud Computing-A Survey"; ISSN: 2278 – 1323 International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) 2(2) 345-567
- Curtmola N. Gupta H., Bharti C., Tanvi A., and Charul D., (2011): Cloud Computing: Comparison with Previous Technique and Research Challenges, International Journal of Computer Applications (0975 – 8887, volume 85- No 8 January 2014
- Ekong P., and Ejiofor U C, 2016: securing of stored data usingself destructive mechanism, ISSN: 2278 – 1323 International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) 2(2) 345-354.
- **Geambasu** F., Vadym Mukhin J., and Artem V., (2009): "Security Risk Analysis for Cloud Computing Systems" The 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Prague, Czech Republic, 15-17 3(3), ISSN 2319 – 4847
- Harjit S. Lamba C. andGurdev S. (2012): , "Cloud Computing-Future Framework for management of NGO's", Department Of Computer Science, Eternal University, Baru Sahib, HP, India. IJoAT, ISSN 0976-4860, 2(3)23-56.
- Jeremy Geelan, (2009): "Twenty-One Experts Define Cloud Computing", cloud computing journal published. IEEE 1(3)12-23.