ADV MATH SCI JOURNAL
Advances in Mathematics: Scientific Journal **9** (2020), no.7, 4485–4493
ISSN: 1857-8365 (printed); 1857-8438 (electronic)
https://doi.org/10.37418/amsj.9.7.17 Spec. Issue on RDESTM-2020

# DESIGN AND IMPLEMENTATION OF CRYPTCLOUD SYSTEM FOR SECURING FILES IN CLOUD

P. RAMESH NAIDU[1], N. GURUPRASAD, AND DANKAN GOWDA V

ABSTRACT. This research paper deals with an efficient method of securing files and folders in cloud using the method of cryptography. Some features such as Advanced Encryption System (AES) algorithm is used to convert plain text to cipher text and cipher text to plain text of the confidential data using one of the most promising and secured technique known as Cipher text-Policy Attribute-Based Encryption (CP-ABE). In this research paper, we will be analyzing the different ways of misusing credentials to access: one among is the cloud user and the other is on the semi-trusted network authorization side. To reduce risk, we recommend the CryptCloud which is secure and more liable trusted authority and able to cancel CPABE based cloud storage system with trace and audit the white-box. We demonstrate how to analyze security issues and further show the usage of our system through experiments, an enhanced AES cryptosystem to handle the failures decryption of the original AES. Then we will be developing a reliable and accurate scheme based on the enhanced AES and confidential distribution of key for each file. As per the data owner new access policy is specified, the stored cipher text will be updated directly in the cloud server without decryption and the update at the cloud will be able to verify by data owner. The proposed scheme can verify the distributed confidential information to avoid the users from Hacking and can reject various attacks such as the collusion attack.

[1]*corresponding author*

2010 *Mathematics Subject Classification.* 68P25, 94A60.

*Key words and phrases.* Advanced Encryption System, CryptCloud, CPABE, Public key Encryption.

## 1. INTRODUCTION

This paper is about a stepwise walk through to solve a problem of securing files and folders in cloud using cryptography.Some time data which is stored in the cloud server is sensitive that can be hacked because we cannot trust cloud server so prominently [1]. So that, the cipher text data will be stored in the cloud. When we are Updating the stored cipher text which is in the cloud when a data owner chosen a new policy to access data and user need authenticity to access data[2] and to verify data of a user to access cipher text data. The data owner and cloud user first login to the cloud using their credentials. Owner uploads a file in cloud [3] and user requests to that file. File gets uploaded in cloud assigning a unique key to each file when its uploaded. Key is generated using AES algorithm which is of 128 bits. The time of file uploading gets recorded in the database[4]. The user needs the key to access the file and it can be accessed for a certain time limit. User requests the access of file to the admin, admin either accepts the requests or it can be denied [5]. User can download the file and view it using a private key of the file. This paper comprises of 5 major sections [6], which will guide you step by step to the research and findings. These are: 1) Introduction 2) Research Elaborations 3) Results or Finding 4) Conclusions

## 2. RESEARCH ELABORATIONS

This section of the paper involves with different strategies and approaches for encrypting and decrypting the generalized model of the cloud [7] architecture using symmetric key encryption system. The steps involved in the development [8] are as mentioned below which will be discussed in detailed. The list are mentioned below:

(1) Uploading and encryption of file
(2) Key Generation
(3) Decryption
(4) Request and Response

2.1. **Uploading and encryption of file.** As the owner uploads the file in cloud after login in, a unique private key is assigned to a file. Key generation and file gets encrypted [9] using Advanced Encryption System (AES) algorithm at the time of uploading a file. The generic approach for file uploading [10] and

validating is described below using flow diagram. Encryption Using AES:

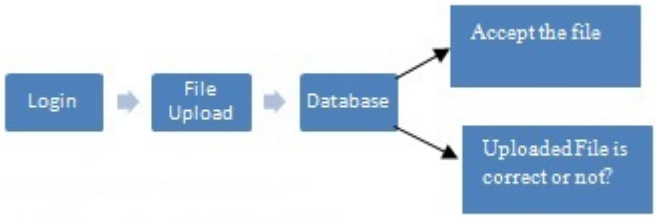


FIGURE 1. Uploading and validating

Cipherfun(byte in[4*Nbs], byte put[4*Nbs], word
wd[Nbs1*(Nrs+1)])
begin
byte stateone[4,Nb]
stateone = get
AddRKey(stateone, wd[0, Nbs-1])
for round = 1 step 1 to Nrs–1
SbBytes(stateone)
Rowshift(stateone)
Columnsmix(stateone)
AddRKey(stateone, w[round*Nbs, (round+1)*Nbs-1])
end for
SbBytes(stateone)
Rowshift(stateone)
AddRKey(stateone, wd[Nrs*Nbs, (Nrs+1)*Nbs-1])
out = state
end

Initially, decrypted data is taken as an input and send to state array element. After AddRKey(add round key), stateone array is converted to implement a function round of 10,12 or 14 times, with final little different [11] from first Nrs H1 rounds. The final Stateone output is then copied and displayed. The function round key is used in terms of parameterized key that consists of single array of

4 bytes of words will be generated by using Key generation technique. The individual transformations - SbBytes(), Rowshift(), Columnsmix(), and AddRKey() process the state array and are described as follow:

a. SbBytes()-SubBytes uses S-box which is applied [12] independently on each byte of 4x4 matrix. First Four bit represents the row number and next four bit represents column number .Size of S-box is 16x16.Get the corresponding value from the S-box and then covert it into 8 bits to substitute it into the block. Consider first four bit 0000 as row number which 0 and next four bit 0101 as column number which is 5.0th row and 5th column in s-box has value of 52.it is again converted in 8 bits. This will be stored in state array.
b. Rowshift()-ShiftRows function performs circular right shift operation of rows of state array. There will be no change in 0th row. Row 1 ,2,3 will be shifted 1bits,2bits,3bits respectively.]
c. Columnsmix()- Output of Rowshift() will be considered as input to MixColumns().Consider each word as one column and apply multiplication operation with predefined 4x4 matrix. The result will be one word which is again stored in state array.
d. AddRKey()-AddRoundKey function perform xor operation. Output from Columnsmix is xored With key .First column of output of columnmix xored with first column of the key and the resultant is stored in state array.

2.2. **Key Generation.** Key generation routine is applied on cipher key K in AES algorithm. Generate the words of Nbs (Nrss + 1) using key expansion. Initially there will be Nbs words requiring Nr rounds. The resultant will be array of 4 bytes words key schedule which is denoted as [wi],i covers range from 0 to Nb (Nrs + 1).Key Expansion algorithm consists of two functions Rotword() and Sbwords. Based on input of Rotwords function it performs cyclic permutation and produces an output. Subwords function performs S box on each 4 byte of input to produce an output word.

Key Expansion :

ExpansionKey(byte keys[4*Nks], word wd[Nbs*(Nrs+1)], Nks)
begin
word tempval

```
i = 0
while (i < Nks)
wd[i] = word(key[4*j], key[4*j+1], key[4*j+2], key[4*j+3])
j = j+1
end while
j= Nks
while (j < Nbs * (Nrs+1)]
tempval = wd[j-1]
if (j mod Nks = 0)
tempval = SubWord(RotWord(tempval)) xor Rcon[i/Nks]
else if (Nks > 6 and i mod Nks = 4)
tempval = SubWord(tempval)
end if
wd[j] = wd[j-Nks] xor tempval
j = j + 1
end while
end
```

2.3. **Decryption or Inverse cipher.** The transformation of the encryption can be reversed and then compiled in the back order to create the inverted Cipher of the AES algorithm. The Inverse Cipher functions used in are - InverseShiftRows (), InverseSubBytes (), InverseMixColumns () and AddRKey ().

Inverse Cipher:

```
InverseCipher(byte in[4*Nbs], byte put[4*Nbs],
word wd[Nbs*(Nrs+1)])
begin
byte stateone[4,Nbs]
stateone= get
AddRKey(stateone, wd[Nrs*Nbs, (Nrs+1)*Nbs-1])
for round = Nrs-1 step -1 goto 1
InverseShiftRows(stateone)
InverseSubBytes(stateone)
AddRKey(stateone, wd[round*Nbs, (round+1)*Nbs-1])
InverseMixColumns(stateone)
end for
```

InverseShiftRows(stateone)
InverseSubBytes(stateone)
AddRKey(stateone, wd[0, Nbs-1])
put = stateone
end
The individual transformations - InverseSubBytes(), InverseShiftRows(), InverseMix-Columns(), and Inverse of AddRKey() process the state array and are described as follow:

a. InverseShiftRows()-function performs circular inverse right shift operation of rows of state array. There will be no change in 0th row. Row 1 ,2,3 will be shifted 1bits,2bits,3bits respectively.
b. InverseSubBytes()-InverseSubBytes uses InverseS-box which is added independently on each byte of 4x4 matrix of state array.
c. InverseMixColumns()-It is inverse of MixColumns() .InverseMixColumns () Follow column column by column, treat four-term polynomial as a each column.
d. Inverse of AddRoundKey()-Inverse of AddRoundKey function perform Inverse xor operation. Output from InverseMixColumns is xored With key .First column of output of InverseMixcolumn xored with first column of the key and the resultant is stored in state array .

2.4. **Request and Response.** The general architectural design or the workflow of request and response of the user and admin. User requests for accessing the file and admin sends the response towards it.

## 3. Results And Findings

Each file is getting encrypted and a key is assigned to every file through which user can access the file. We have used the time scheduling algorithm, so the user can download the file only during the specified time period, and if the user fails to download the file within the time period then the pass-code expires and the user again needs to send the request to the admin and again the admin needs to accept the request.

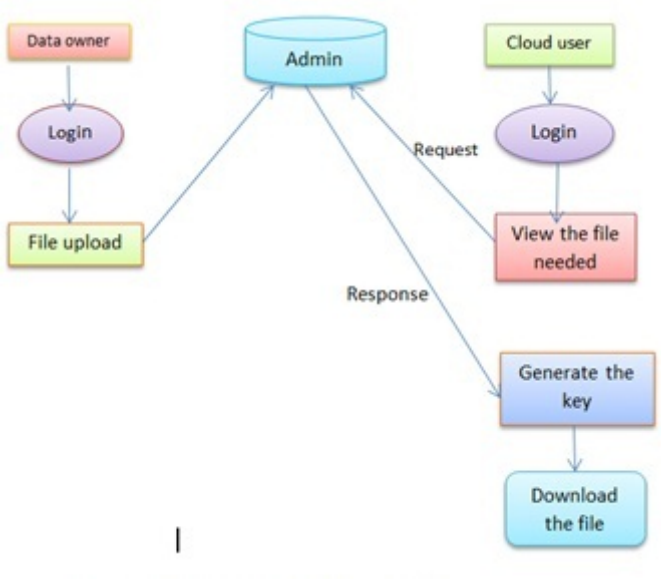


FIGURE 2. The request and response


FIGURE 3. Users request accepted by the admin

## 4. CONCLUSION

When the data is being stored by the admin in the cloud there is always an issue if the data is accessed securely by the user or not. There are number of exterior threats that can lead to data leaks, including harmful deceptions from cloud service providers or cloud user accounts. The best way is to encode files which can be achieved with use of hybrid cryptography. Hybrid cryptography with the help of AES algorithm provides security to the files stored in the cloud and make it less vulnerable to threats. It can be seen that AES is the most secure

FIGURE 4. Passkey entered by the user to access the file

FIGURE 5. File downloaded after entering the passkey

symmetric encryption technology. User can access files from cloud only if they have the key which is sent by the admin after secure authorization. It guarantees a method for safe cloud environment.

Future work:

The main aim for future is to implement code through which user can get to know if there request has been accepted or not. We can extend our scope by including backup and recovery actions to help prevent data loss in the event of an attack.

## REFERENCES

[1] M. ALI, R. DHAMOTHARAN: *Sedasc: Secure data sharing in clouds.* , IEEE Systems Journal, **1**(2) (2017), 395–404.

[2] M. ARMBRUST, A. FOX: *Security in cloud computing: Opportunities and challenges*, Information Science and Technology Journal, **2** (3) (2015), 357–383.

[3] M. ALI , S. U. KHAN: *A view of cloud computing*, Communications of the ACM Journal, **53** (4) (2010), 50–58.

[4] N. ATTRAPADUNG, H. IMAI: *Attribute-based encryption supporting direct/indirect revocation modes*, Journal of Cryptography and Coding, **3** (14) (2009), 278–300.

[5] D. V. KISHORE, D. V. GOWDA : *MANET topology for disaster management using wireless sensor network*, 2016 International Conference on Communication and Signal Processing (ICCSP), (2016), 0736–0740.

[6] D. V. GOWDA, D. V. KISHORE : *Optimization of motorcycle pitch with non linear control*, 2016 International Conference on Recent Trends in Electronics, Information and Communication Technology (RTEICT), (2016), 1656–1660.

[7] M. PENNA, D. V. GOWDA : *Design and implementation of automatic medicine dispensing machine*, 2017 International Conference on Recent Trends in Electronics, Information and Communication Technology (RTEICT), (2017), 1962–1966.

[8] D. V. GOWDA, C. A. VARUN : *Implementation of swarm intelligence in obstacle avoidance*, 2017 International Conference on Recent Trends in Electronics, Information and Communication Technology (RTEICT), (2017), 525–528.

[9] D. V. GOWDA, A. C. RAMACHANRDA: *Synthesis and Modeling of Antilock Braking System Using Sliding Mode Controller*, Journal of Advanced Research in Dynamical and Control Systems, **11**(12) (2018), 208–221.

[10] D. V. GOWDA, A. C. RAMACHANRDA: *Modelling and Performance Evaluation Of Anti-Lock Braking System*, Journal of Engineering Science and Technology, **14**(5) (2019), 3028–3045.

[11] D. BONEH, X. BOYEN: *Short signatures without random oracles* , Journal of EUROCRYPT, **2**(1) (2004), 56–73.

[12] H. CAI, B. XU: *Iot-based big data storage systems in cloud computing*, Perspectives and challenges. IEEE Internet of Things Journal, **4**(1) (2017), 75–87.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY, VISVESHWARAYA TECHNOLOGICAL UNIVERSITY, BANGALORE, KARNATAKA STATE, INDIA

*Email address*: ramesh.naidu@nmit.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, GLOBAL ACADEMY OF TECHNOLOGY, VISVESHWARAYA TECHNOLOGICAL UNIVERSITY, BANGALORE, KARNATAKA STATE, INDIA

*Email address*: nguruprasad18@gmail.com

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING, B.M.S INSTITUTE OF TECHNOLOGY AND MANAGEMENT, VISVESHWARAYA TECHNOLOGICAL UNIVERSITY, ADDRESS: BANGALORE, KARNATAKA STATE, INDIA

*Email address*: dankan.v@bmsit.in