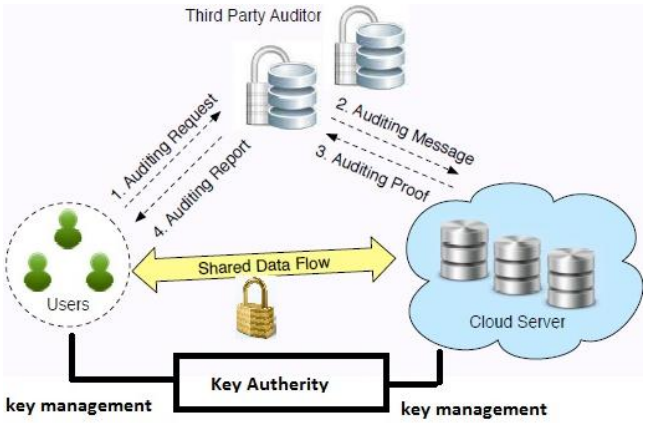

Figure 2: Architecture of proposal system

With our mechanism, the identity of the signer on each block in shared data is kept private from a multiple public verifier, who is still able to publicly verify the integrity of shared data without retrieving the entire file.Along with to provide security for sharing the data between user and cloud. Our experimental results demonstrate the effectiveness and efficiency of our proposed mechanism when auditing shared data.

5. Proposed Algorithms

Propose In the addition to this the mechanism is able to perform multiple auditing tasks simultaneously instead of verifying them one by one and using multiple public verifier to improve the efficiency of auditing task and also using CP-ABE Algorithm between user and cloud server to and the protect user confidential data.The experimental results when sharing data integrity auditing effectiveness and demonstrate the efficiency of this system.for this purpose using three algorithms such as AES,SHA and CP-ABE.In that AES and SHA are used for to generate the public key and ring signature for each file.CP-ABE Algorithm between user and cloud server to and the protect user confidential data.This Algorithm does not allowed the Unauthorized users to access the file from cloud server.

5.1 CP-ABE Algorithms

Two methods in the fine-grained access control based on ABE first one is KP-ABE and second is CP-ABE. In KP-ABE, each of the private key attribute is link with an access structure is specifies which type of cipher texts the key is used to decrypt, and cipher text is labelled with a sets of attributes. In a CP-ABE system, a user's key is link with a set of attributes and an encrypt the cipher text will specifies an access policy over attributes. In KP-ABE constitution to realized the monotones access structures for key policies. The constitution is only proved the secure for the generic group model. To overcome this problem presented another constitution that is proved to be secure under the drastic standard model. Attribute-based encryption [ABE] means that encrypted access control. Cipher texts not necessary for encrypt to one particular user[9]. User private keys and cipher texts link with a set of attributes or a policy over the attributes. When A match in between the user's private key and the cipher text, then only decryption is possible.

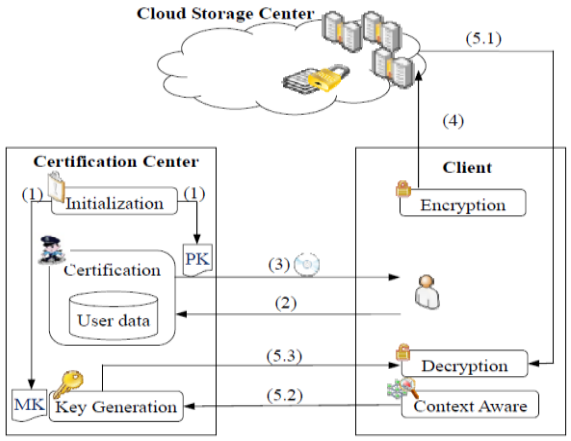

Figure 3: CP-ABE Algorithm structure.

a) **Initialized system**: The First the certification centre can generate the system security parameters like PK and MK, after transmitted PK to the user and MK to the key generation.
- First need to register for the user inputs ID and password to be log in the system.
- The user authentication unit should be verifies the CC in the first in first out manner, then distributes the security parameter
- like PK and the client element to the authorized users.
- The user can accessing or receiving the data or resources from the cloud storage centre.
- The context collector can collects the user's current contexts and transport or move to the key generation of the CC.
- The key generator of the CC generates decryption key according to MK and the get the contexts, and sends it to the decryption unit in the client. After decryption unit can be decrypt the resource.

b) **Context-aware:** Access control is Policy based on CP-ABE algorithm with context its awareness.

5.2 Secure Hash Algorithm

Secure Hash Algorithm is one of the most important cryptographic hash functions and SHA is short. File verification purpose use SHA-1 is smart to contrast the checksums created after running the algorithms and require two files for contrasting purpose. This SHA-1 is the second iteration of the cryptographic hash function and destroy the last SHA-0. An SHA-2 cryptographic hash function is prepared and also SHA-3 is being developed. SHA-256 transport an input messages into the 256 bits message digest[10]. SHA is use to generate the key signature.SHA having six steps.

Stage 1: Packaging Message: Input binary message is 1 and filled is 0 until length = 448 mod 512.Then attached 64-bit binary number. The filled message length is a Multiple of 512 bits, which decides how many '0' to be filled.
Stage 2: Parsing: The padded message is separate into the N 512-bit blocks.
Stage 3: Message Extension (Scheduler): Each 512 bit block can be split into 16 –bit words 32-bit words

Stage 4: Message squeezing: The words from scheduler stage are then passed to the SHA squeezing function.
Stage 5: The algorithm is implemented by 64-cycle repeated each block.
***Stage 6*:** After 64 emphasis of the squeezing function, an median the hash value.

5.3 AES- Advanced Encryption Standard

In cryptography, the Advanced Encryption Standard (AES) is a symmetric-key encryption standard. AES algorithm is used to provide security on the data stored. AES is symmetric encryption algorithm in which to encrypt the message sender uses public key of receiver and its private key is used by receiver to decrypt the message.

Begin
1. Data is stored on cloud server
2. Data is encoded
3. Verification of data is should be done by CSP using AES
4. **If** data is valid

Go To Module T

Else
Invalid data

End

Module T:

Begin
1. Checked the data stored.
2. **If** proof = direct then
Report = direct access

Else
Return {1, 0}
1: if integrity of data is verified as accurate
0: if integrity of data verified is improper

END

- AES is preferred over DES algorithm as it is more secure.
- AES data encryption is mathematically more efficient and elegant cryptographic algorithm. Key length option is the main strength of the algorithm. Time required to crack an encryption algorithm is directly related to the length of the key used to secure the communication. AES gives an option to choose a 128-bit, 192-bit or 256-bit key, making it exponentially stronger as compared to the 56-bit key of DES.
- Block size of DES is small compared to AES
- A balanced Feistel structure is used by DES while substitution-permutation is used by AES.

6. Results

User can upload his file on cloud server and public verifier can generate the keys for each file.

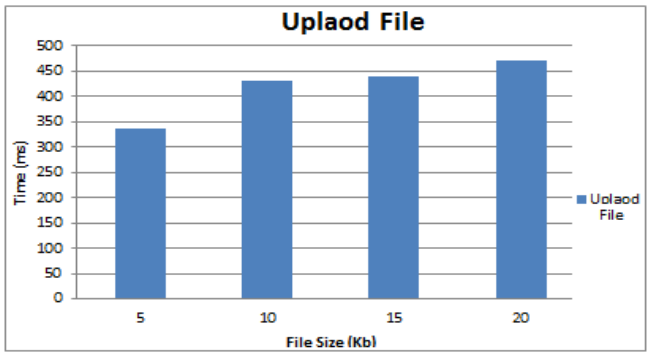

Figure 4: Upload the file in Cloud Server Graph

Public verifier can do the batch auditing and give keys for each file.

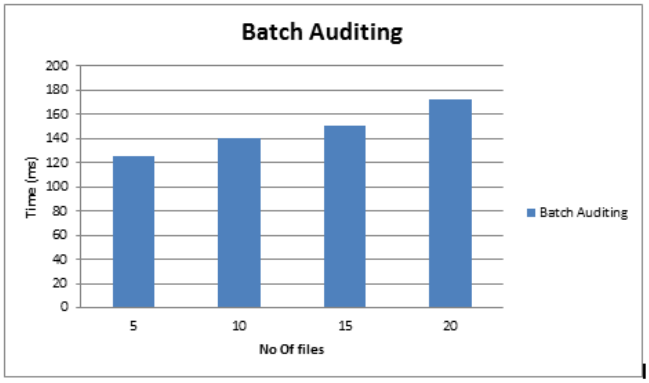

Figure 5: Batch Audting Graph

Public verifier can do the batch auditing and give keys for each file.using these keys user can easily encrypt the his file from cloud server.

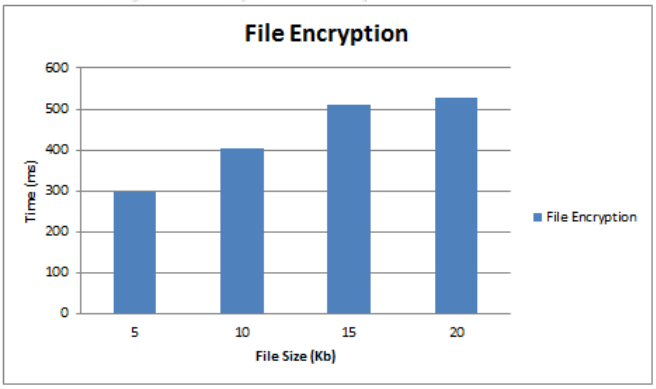

Figure 6: File Encryption Graph

Only valid user can decrypt the file from cloud server

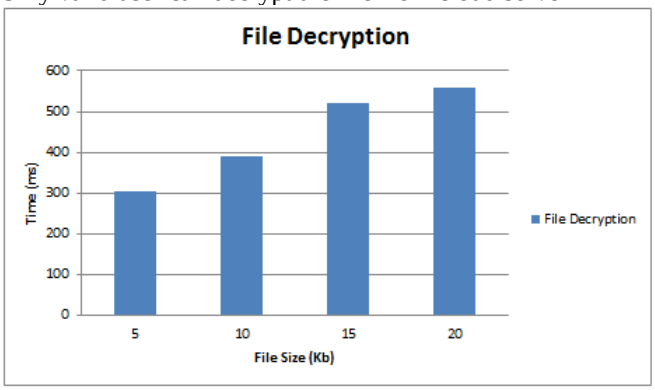

Figure 7: File Decryption Graph

CP-ABE Algorithm will take less time for Encryption as compare AES Algorithm.

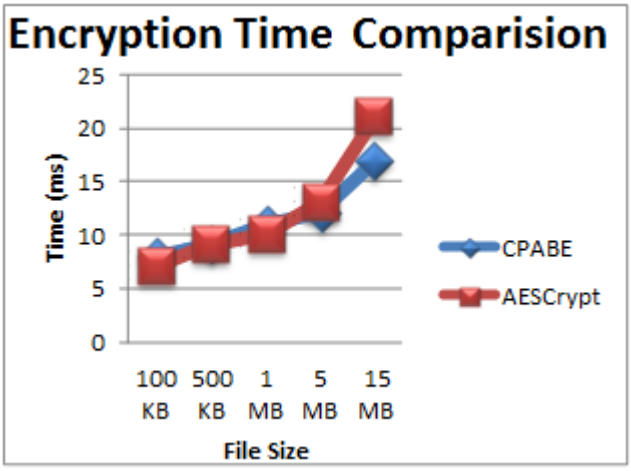

Figure 8: CP-ABE and AES Encrypt Graph

CP-ABE Algorithm will take less time for Decryption as compare AES Algorithm.

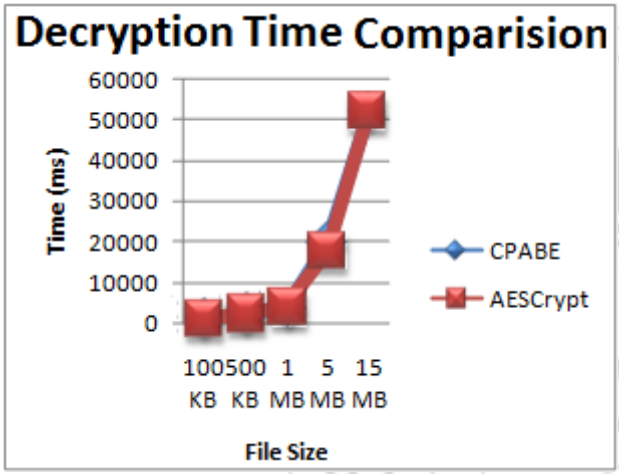

Figure 8: CP-ABE and AES Decrypt Graph

7. Conclusion

With our mechanism, the identity of the signer on each block in shared data is kept private from a multiple public verifier, who is still able to publicly verify the integrity of shared data without retrieving the entire file. using multiple public verifier to improve the efficiency of auditing task and also using CP-ABE Algorithm between user and cloud server to protect the user confidential data

References

[1] B. Wang, B. Li, and H. Li, "Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud," Proc. IEEE Fifth Int'l Conf. Cloud Computing, pp. 295-302, 2012.

[2] K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69-73, 2012.

[3] D. Song, E. Shi, I. Fischer, and U. Shankar, "Cloud Data Protection for the Masses," Computer, vol. 45, no. 1, pp. 39-45, 2012.

[4] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," Proc. IEEE INFOCOM, pp. 525-533, 2010.

[5] B. Wang, M. Li, S.S. Chow, and H. Li, "Computing Encrypted Cloud Data Efficiently under Multiple Keys," Proc. IEEE Conf. Comm. and Network Security (CNS '13), pp. 90-99, 2013.

[6] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," Comm. ACM, vol. 21, no. 2, pp. 120-126, 1978.

[7] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. 14th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '08), pp. 90-107, 2008.

[8] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamic for Storage Security in Cloud Computing," Proc. 14th European Conf. Research in Computer Security (ESORICS'09), pp. 355-370, 2009.

[9] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring Data Storage Security in Cloud Computing," Proc. 17th Int'l Workshop Quality of Service (IWQoS'09), pp. 1-9, 2009.

[10] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S.S Yau, "Dynamic Audit Services for Integrity Verification of Outsourced Storages in Clouds," Proc. ACM Symp. Applied Computing (SAC'11), pp. 1550-1557, 2011.

[11] N. Cao, S. Yu, Z. Yang, W. Lou, and Y.T. Hou, "LT Codes-Based Secure and Reliable Cloud Storage Service," Proc. IEEE INFOCOM, 2012.

[12] B. Wang, B. Li, and H. Li, "Certificate less Public Auditing for Data Integrity in the Cloud," Proc. IEEE Conf. Comm. and Network Security (CNS'13), pp. 276-284, 2013.

[13] C. Wang, S.S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Trans. Computers, vol. 62, no. 2, pp. 362-375, Feb. 2013.