Privacy - Preserving Public Authentication for Shared Data in Cloud

Priyanka R Chaudhari
Computer Engineering Dept,
BSIOTR
Savitribai Phule Pune University,
Pune, India

Archana C. Lomte
Computer Engineering Dept,
BSIOTR
Savitribai Phule Puneuniversity,
Pune, India

ABSTRACT
Data security is an important in cloud. Access control is major issue to secure the stored data in cloud. Thus the proposing privacy preserving authenticated access control scheme for securing shared data in cloud. In the proposed scheme supports authenticated access control for anonymous user to share data on cloud also additional feature in access control which provides authentication for the user, in which only valid users are able to decrypt the stored information. User authentication and access control scheme are introduced in decentralized, which prevent replay attacks and also supports to creation, , and reading, modification data stored in the cloud. Implementation also provide feature user revocation.

Keywords
Access control, Authentication, key distribution centers (KDCs).

1. INTRODUCTION
Now a day's Cloud computing is a growing paradigm which gives facility of data storage and access for cloud users, Clouds offers multiple services like applications (e.g., Google Apps), and platforms to help developers write apps (e.g. Amazons S3, Windows Azure), infrastructures (e.g., Amazons EC2, Eucalyptus, Nimbus), Much of the data stored in clouds is sensitive, for e.g., medical records and social networks. Secured access control must provided to this sensitive information which can often related to health, important documents (as in Google Docs or Drop box) or even personal data (as in social networking)[1].

Access control is gaining importance in online social networking where users (members) store their personal information, pictures, and videos and share them with selected groups of users or communities they belong to.

Security and privacy are vital issues in cloud computing. First the user ought to authenticate itself before initiating any transaction, but on the other side, it should be ensured that the cloud or other users do not know the identity or credentials of the user. There are three objectives to protect the data.

Confidentiality – Confidentiality is roughly equal to privacy. Measures undertake to make sure confidentiality are designed to prevent sensitive information access and disclosure. Access is being restricted to authorized user.

Integrity – Integrity involves maintaining the consistency, accuracy, and trustworthiness of data. It provides security against improper data modification or destruction.

Availability – It provides consistent access and use of data.

Authentication has significant to different fields. In anthropology, antiques, and art, a common crisis is verifying that a given arti-fact was formed by a particular person or was formed at a certain place. In computer science, verifying a person's identity is often required to secure access to confidential data or systems. In the Existing System, data are accessed in centralized form on the basis of key distributed center. Key distributed center does not support for authentication. A single failure of KDC can affect the maximum number of data in cloud storage. It is most difficult to maintain the large number of data in cloud for centralized form. In clouds where is a very difficult task and involves technical facts and law enforcement. Clouds and users, No one unable to deny any requested or performed operations. Due to use of single key distribution center, single point failure may be occurred and at that it is very difficult to manage because of multiple numbers of users those are supported in a cloud. But in those system having disadvantages like, data storage are only predicated on centralized form. And additionally it affects the maintenance of astronomically immense number of data storage in cloud. It does not fortify the authentication control.

The schemes [11], [15], [12], [17] on access in cloud are centralized manner. All schemes use attribute based encryption (ABE) excepting [11]. In scheme [11], they apply symmetric key approach and that scheme is not provide authentication. The schemes [11], [15], [19] are not provide authentication. The Scheme by Zhao [17] provides privacy preserving authenticated access control in cloud. Though, the authors take a centralized loom where a single key distribution center (KDC) distributes secret keys and attributes to all users. Regrettably, a single Key Distribution Center is a single point of failure and also difficult to maintain because of the large number of users that are uses cloud environment. Our scheme uses decentralized approach to distribute secret keys to users. And also provide many KDCs in different locations in the world. Yang [18] proposed a decentralized approach; their scheme does not authenticate users, who want to stay anonymous for access control in cloud. In work, Ruj [19] projected a distributed access control scheme in clouds. But, the scheme do not provide user authentication. The other disadvantage is that a user can only create and store a file and other users can only read the file. Write access is not permitted to users other than the creator. In paper [2], support to authenticate the validity of message without hiding the identity of the user who has stored data in the cloud. In our scheme Implementation also provide user

revocation, that was not addressed in [2]. Our scheme is resistant to replay attacks, in which a user can replace new data with stale data from earlier write, even if it no longer has valid claim policy. This is an important assets because a user, revoked of their attributes, might unable to write to the cloud. Our scheme also allows writing multiple times which was not allowable in work [19].

The primary goals of our scheme are:

1. Distributed access control of data held in cloud to make sure only authorized users with valid attributes can access them.

2. Authentication of users who store and modify their data in the cloud.

3. The identity of a person is protected in the cloud during authentication.

4. The architecture is decentralized, for instance there may be numerous key distribution center (KDCs) for key management.

5. The access control and authentication are both collusion resistant, for instance no two users are able to collude and access data or authenticate themselves, cons independently not authorized.

6. Revoked users cannot access data after they've been revoked.

7. The proposed scheme is resilient to replay attacks. A writer whose keys and attributes have been completely revoked cannot write back stale information.

8. The protocol supports multiple read and writes on your data held in the cloud.

9. The expense is akin to the current centralized approaches, and thus the costly operations made for professionals done by the cloud.

2. ALGORITHM

2.1 Blowfish Algorithm

The Blowfish Algorithm is used for Encryption and Decryption, respectively. The working of the algorithm is given below. Blowfish Algorithm is a symmetric block cipher technique that is used for encryption of data. It takes a variable length keys an input, from 32 bits - 448 bits, to secure data.

Blowfish is combination of key-dependent S-Boxes, Feistel network, and a non-invertible F function 16 times. The input is a 64-bit data element, can be any data size upto 448 bits. That algorithm is faster than other encryption algorithms when implemented with large data also. The Blowfish Encryption/Decryption Algorithm is manipulates data in 64-bit block size. The algorithm consists of a key-expansion and a data encryption. Key expansion converts a variable length key of 56 bytes (448 bits) into several subkey arrays of 4168 bytes. Fig 5.1 Feistel structure of blow fish algorithm.

Blowfish algorithm uses 16 rounds Feistel structure. Each round of it is consist of a key–data dependent substitution, key dependent permutation. Blowfish uses a large no's of sub keys. The P-array includes 18 32bit sub keys- P1, P2, P3... P18. There are four S-boxes of 32-bit with 256 entries each box:

S1,0, S1,1,S1,2,..., S1,255;
S2,0, S2,1,S2,2..., S2,255;
S3,0, S3,1,S3,2..., S3,255;
S4,0, S4,1,S3,2.., S4,255.

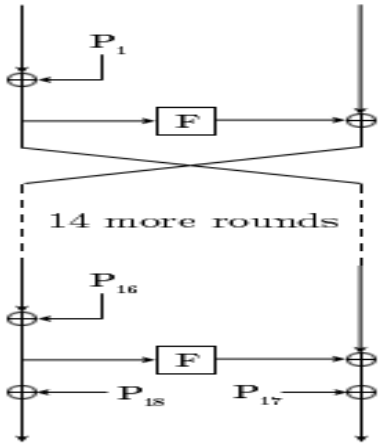


Fig 1. Feistel Structure of Blowfish Algorithm

Steps involved in Blowfish Algorithm

1) INPUT

- The input is a 64-bit data element, x

2) PROCESS

- Divide x into two 32-bit halves:- xL, and xR.
- Then, Also for i = 1 to 16:
  - xL = xL XOR Pi
  - xR = F(xL) XOR xR
  - Swap xL and xR
- After completion of 16th round, swap xL and xR again to undo the last swap.
- Then, xR = xR XOR P17 and xL = xL XOR P18
- Combine xL and xR.

3) OUTPUT

- Blowfish uses a large number of sub keys. These keys should be pre-computed before encryption or decryption of data.
- Cipher text (C) is produced in result.

Decryption is exactly the same as encryption, only the use of P1, P2, P3……., P18 in the reverse order. A common is to use inverse order of encryption as decryption algorithm.

2.2 RSA

RSA is a public-key cryptosystems and is utilized for secured data transmission. The encryption key is public and different from the decryption key which is reserved secret. In RSA is predicted on factoring the product of 2prime numbers.

A user of RSA creates & then publishes a public key predicted on the 2 prime numbers, along with an auxiliary cost. The prime numbers should be reserved secret. Anyone is able to use the public key to encrypt data, but with presently published technique, if the public key is large enough, only some users with knowledge of the prime no's can possibly decode the dada.

The RSA algorithm [22] involves:

• Key generation

RSA includes a public key and a private key. The public key is known by everybody and is utilized for encrypting data. Data encrypted with the public key is able to decrypt in a realistic amount of time using the private key. The keys are generated the following way:

1. Choose 2 dissimilar prime no. p and q.

For security, the integers p and q s should be culled at random, and should be of same bit-length.

2. Compute n = p q.

n = the modulus for public key and private key.

3. Compute $\varphi(n) = \varphi(p)\varphi(q) = (p - 1)(q - 1) = n - (p + q - 1)$,

Where,

φ = Euler's totient function. This value is put private.

4. Choose an integer e such that

$1 < e < \varphi(n)$,

$\gcd(e, \varphi(n)) = 1$;

where, e and $\varphi(n)$ are co-prime.

e = public key exponent.

e having a short bit-length and small Hamming weight results in more efficient encryption – most commonly 216 + 1 = 65,537. However, much smaller values of e (such as 3) have been shown to be less secure in some settings. [8]

5. Determine d

$d \equiv e{-}1 \pmod{\varphi(n)}$

Where d= modular multiplicative inverse of e (modulo $\varphi(n)$).

Also this is: solve for d given

$d \cdot e \equiv 1 \pmod{\varphi(n)}$

This is often computed using the extended Euclidean algorithm. Using the pseudo code in the Modular integers section, inputs a and n correspond to e and $\varphi(n)$, respectively.

d = private key exponent.

The public key consists of the modulus n and the public (or encryption) exponent e. The private key consists of the modulus n and the private (or decryption) exponent d, which must be kept secret. p, q, and $\varphi(n)$ must also be kept secret because they can be used to calculate d.

RSA Signature scheme:

1. Compute the message digest H of the message,

2. H = Hash(M)

3. Form the byte string, T, from the message digest, H, according to the message digest algorithm, Hash, as follows where T is an ASN.1 value of type DigestInfo encoded using the Distinguished Encoding Rules (DER).

4. Form the k-byte encoded message block, EB,

5. EB = 00 || 01 || PS || 00 || T where || denotes concatenation and PS is a string of bytes all of value 0xFF of such length so that |EB|=k.

6. Convert the byte string, EB, to an integer m, most significant byte first,

7. Sign with the RSA algorithm

3. PROPOSED SYSTEM

An access control model to impose controlled data sharing in cloud is proposed. In the scheme, maintaining the large number of data in cloud, decentralized access control approaches is proposed. Involving distribution of secret keys to users. KDC generate key which is encrypted using SHA2 algorithm. Authentication will be done by using attribute based encryption. Authentication access control only allows the user for reading purpose. Accessing the data by user only satisfying the access policy and authentication. Distributed access control of data stored in cloud so that only authorized users with valid attributes can access that data. Authentication of users who wants to store and modify their data on the cloud. The Privacy - Preserving Public Authentication for Shared Data in Cloud identity of the user is protected from the cloud during authentication. The scheme uses decentralized architecture, i.e. Multiple Key Distribution centers are used for key management. Both access control and authentication are collusion resistant; two users are not able to collude and access data or authenticate themselves, if they are independently not authorized. And once user is revoked, that users cannot access data the proposed system is prevent replay attacks.

The protocol supports multiple read and writes on the data stored in the cloud. The expenses are comparable to the existing centralized approaches, and the high-priced operations are mostly done by the cloud. Access control with authentication is provided on the basis of attribute based access control. It accessed on decentralized form of approach by satisfying the access policy. It avoids the data loss and only knows the user's policy not their privacy.

Trustee: Trustee, who is assumed to be honest A trustee can be someone like the federal government who manages social insurance numbers etc. On presenting user id (like health/social insurance number), the trustee gives him a key.

KDCs: There are multiple KDCs (here 2), which can be scattered. i.e., these can be server in different parts of the world. A creator on presenting the key to one or more KDCs receives keys for encryption/decryption.

A. Store a File on cloud

If a user wants to store a file on cloud then first creator receives group key from Key generation centre. Distribution and dispatching of generated group keys is processed. Then user gets that key and apply digital signature on the file using RSA algorithm. The file is encrypted using the blowfish algorithm and store on cloud.

B. Reading a File from the Cloud

When user requests data from cloud then first, the cloud validates that user whether user is revoked or non-revoked. If the user is non-revoked then cloud sends encrypted data if request is validate. The decryption is processed using blowfish algorithm.

C. Writing or modifying a File to Cloud

If a user wants to write or modify an already existing file then user must send its message with the claim policy as done during file creation. The cloud verifies the claim policy, and

only if the user is authentic. Then user allowed writing on the file.

D. User revocation

Revoked users cannot access data after they have been revoked. It should be ensured that users must not have the ability to access data, even if they have matching set of attributes and accessing policy. The owners should change the stored data and send updated information to the non-revoked users only. All non-revoked users change their stored data that have matching set of attributes Updated information is not stored in the cloud but transmitted to the non-revoked users who have matching attributes and access policy. This prevents a revoked user to decrypt updated information and get back.

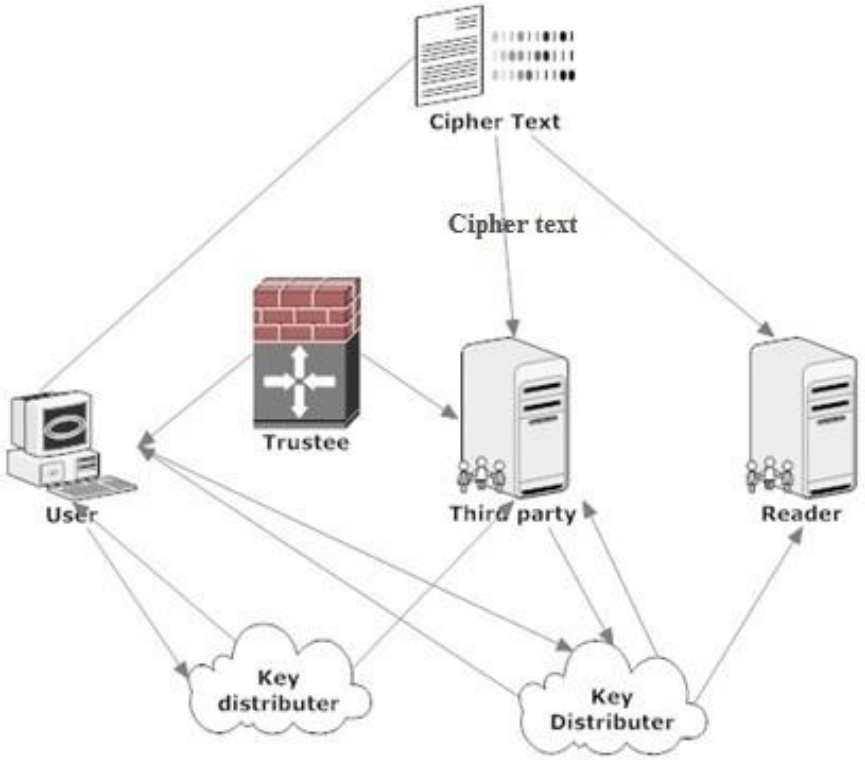


Fig 1: System Architecture

Schemes	**Fine-Grained Access Control**	**Centralized/ Decentralized**	**Read/Write Access**	**Type Of Access Control**	**Privacy Preserving Authentication**	**User Revocation**
Secure And Efficient Access [11]	Yes	Centralized	1-W-M-R	Symmetric Key Cryptography	No Authentication	No
Patient-Centric And Fine-Grained Data Access	Yes	Centralized	1-W-M-R	ABE	No Authentication	No
Attribute Based Sharing Data With Attribute	Yes	Centralized	1-W-M-R	ABE	No Authentication	No
Distributed Access Control In Clouds : DACC	Yes	Decentralized	1-W-M-R	ABE	No Authentication	Yes
DAC-MACs [18]	Yes	Decentralized	1-W-M-R	Abe	Not Privacy Preserving	Yes
Fine-Grained And Flexible Access Control	Yes	Centralized	M-W-M-R	ABE	Authentication	No
Our Scheme	Yes	Decentralized	M-W-M-R	Blowfish	Authentication	Yes

Table I: Comparative Study our scheme with other Access Control Scheme

4. SECURITY OF THE SCHEME

Implemented scheme authenticate a user who wants to write on the cloud. A user must only write access provided

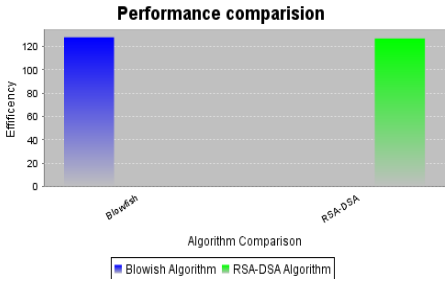
The cloud can validate it access to the claim. An invalid user is to unable receiving the attributes and access from a KDC, if it does not have the credentials from the trustee. Once user's credentials are revoked, then that user is unable to replace data with previous data, thus preventing replay attacks.

Theorem 1. Our access control scheme is secure, collusion resistant and permits access to authorized users only.

Theorem 2. Our authentication data is correct, collusion secure, resistant to the replay of attacks, and protects privacy of the user..

5. RESULT AND DISSCUSSIONS

There are `n` number of user who can be creator, reader and writer on cloud. Cloud verifies the authenticity of the user without knowing the user's identity before storing information on cloud. Creator stores their data with digital signature for authentication. Blow fish algorithm must be applied before storing data to convert it into cipher text.. When any user wants to read and write on stored data, they have to decrypt data with proper decryption key. Readers can read data but readers are not able to modify data. Writer must authorize to modify or rewrite the creator's data. Writer who must be non revoked user, then writer modify or rewrite the creator's data. User who don't having access policy i.e. unauthorized user gets revoked. In this way scheme provide authentication of user, security to stored data in cloud.



Blowfish algorithm is more efficient than existing encryption algorithm. Access control Performance of existing encryption algorithm and blowfish algorithm is calculated shown in fig. Implemented scheme shown in below figure.

Fig: User Registration

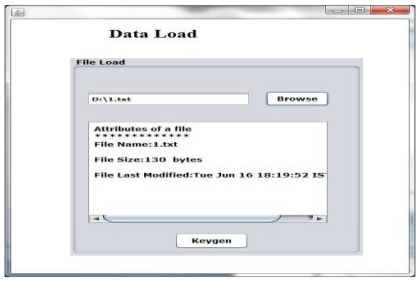


Fig: Store Data

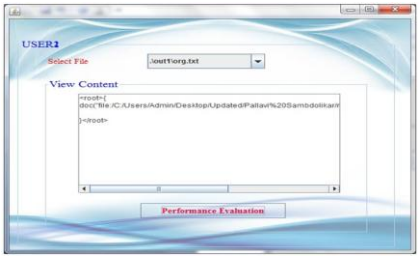


Fig: Read Data on Cloud

Fig: Update Data Received from Cloud

6. COMPARATIVE STUDY

In Table I, the comparison of implemented scheme with other access control schemes and show that our scheme supports many features that the other schemes did not support. 1-W-M-R i.e. only one user can write while many users can read. M-W-M-R means that many users can write and read data. Most of schemes do not support many writes which is supported by our scheme. Our scheme is decentralized but most of the other schemes are centralized manner. Our scheme also support for privacy preserving authentication, which is not supported by others schemes. Our scheme also supports to user revocation, than other schemes.

7. CONCLUSION

The presented Privacy - Preserving Public Authentication for Shared Data in Cloud. Privacy preserving access control technique is decentralized manner. Our scheme supports decentralized access control technique and also supports anonymous authentication. Cloud knows only the access policy of the stored information. The cloud authenticates the user by verifying the credentials even without knowing the original identity of the user. In our scheme, also address the user revocation and also scheme prevents replay attacks. Cloud doesn't know about used details, it only verifies the user information.

8. ACKNOWLEDGMENTS

This is a great pleasure immense satisfaction to express my deepest sense of gratitude thanks to my college, JSPM's Bhivarabai Sawant Institute Of Technology and Research College Of Engineering, Pune and my department of Computer Engineering which has provided the support. I express my heartfelt gratitude to my guide Prof. Archana Lomte who has supported me throughout my research with their patience and knowledge..

9. REFERENCES

[1] Sushmita Ruj, Milos Stojmenovic and Amiya Nayak ,"Decentralized Access Control with Anonymous Authentication of Data Stored in Clouds", IEEE Transactions on Parallel and Distributed Systems, Vol. 25, No. 2, February 2014.

[2] S. Ruj, M. Stojmenovic, and A. Nayak, "Privacy Preserving AccessControl with Authentication for Securing Data in Clouds", Proc.IEEE/ACM Intl Symp. Cluster, Cloud and Grid Computing, pp. 556-563, 2012.

[3] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou,"TowardSecure and Dependable Storage Services in Cloud Computing," IEEE Trans. Services Computing, vol. 5, no. 2, pp. 220-232, Apr.-June 2012.

[4] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "FuzzyKeyword Search Over Encrypted Data in Cloud Computing," Proc. IEEE INFOCOM, pp. 441-445, 2010.

[5] S. Kamara and K. Lauter, "Cryptographic Cloud Storage," Proc.14th Intl Conf. Financial Cryptography and Data Security, pp. 136-149, 2010.

[6] H. Li, Y. Dai, L. Tian, and H. Yang, "Identity-Based Authentication for Cloud Com-puting," Proc. First Intl Conf. Cloud omputing(CloudCom), pp. 157-166, 2009.

[7] C. Gentry, "A Fully Homomorphic Encryption Scheme," PhD dissertation, Stanford Univ., http://www.crypto.stanford.edu/craig, 2009.

[8] A.-R. Sadeghi, T. Schneider, and M. Winandy, "Token-BasedCloud Computing," Proc. Third Intl Conf. Trust and TrustworthyComputing (TRUST), pp. 417-429, 2010.

[9] R.K.L. Ko, P. Jagadpramana, M. Mowbray, S. Pearson, M.Kirchberg,Q. Liang, and B.S. Lee, "Trustcloud: A Frameworkfor Accountability and Trust in Cloud Computing," HP TechnicalReport HPL-2011-38, http://www.hpl.hp.com/techreports/2011/HPL-2011-38.html, 2013.

[10] R. Lu, X. Lin, X. Liang, and X. Shen, "Secure Provenance: The Essential of Bread and Butter of Data Forensics in CloudComputing," Proc. Fifth ACM Symp. Information, Computer andComm. Security (ASIACCS), pp. 282-292, 2010.

[11] W. Wang, Z. Li, R. Owens, and B. Bhargava, "Secure and e_cient access to outsourced data,," in ACM Cloud Computing Security Workshop (CCSW), 2009.

[12] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in ACM ASIACCS, pp. 261270, 2010.

[13] D.F. Ferraiolo and D.R. Kuhn, "Role-Based Access Controls," Proc.15th Natl Computer Security Conf., 1992.

[14] D.R. Kuhn, E.J. Coyne, and T.R. Weil, "Adding Attributes to Role-Based AccessControl," IEEE Computer, vol. 43, no. 6, pp. 79-81,June 2010.

[15] M. Li, S. Yu, K. Ren, and W. Lou, "Securing personal health records in cloud computing: Patient-centric and _ne-grained data access control in multiowner settings,"in Secure Comm, pp. 89106, 2010.

[16] A. B. Lewko and B. Waters, "Decentralizing attribute-based encryption," in EUROCRYPT, ser. Lecture Notes in Computer Science, vol. 6632. Springer, pp. 568588, 2011.

[17] F. Zhao, T. Nishide, and K. Sakurai, "Realizing _ne-grained and flexible access control to outsourced data with attribute-based cryptosystems," in ISPEC, ser.Lecture Notes in Computer Science, vol. 6672. Springer, pp. 8397, 2011.

[18] Kan Yang, Xiaohua Jia and Kui Ren, "DAC-MACS: E_ective Data Access Control for Multi-Authority Cloud Storage Systems," IACR Cryptology ePrint Archive, 419, 2012.

[19] S. Ruj, A. Nayak, and I. Stojmenovic, "DACC: Distributed access control in clouds," in IEEE TrustCom, 2011.

[20] M.Chase, "Multi-authority attribute based ncryption,"inTCC,ser.Lecture Notes in Computer Science, vol. 4392. Springer, pp. 515534, 2007

[21] G. Wang, Q. Liu, and J. Wu,"Hierarchical attribute-based encryption for fine grained access control in cloud storage services," in ACM CCS, , pp. 735737, 2010.

[22] https://en.wikipedia.org/wiki/RSA_%28cryptosystem%29

[23] Priyanka R. Chaudhari and Prof. Archana C. Lomte "Secure Access Control on Shared Data In Cloud", International Journal of Engineering Technology and Computer Research (IJETCR)Volume 2; Issue 6; Page No.148-153.