ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 5, Issue 5, May 2016

Decentralized Access Control For Secure Storage Service Using Message Digest

Rani Shriram Joshi
Computer Science and Engineering
H.V.P.M's C.O.E.T
Amravati, India

Dr.A.B.Raut
Computer Science and Engineering
H.V.P.M's C.O.E.T
Amravati, India

Abstract***: This methodology proposes a new decentralized access control scheme for secure data storage in clouds that supports anonymous authentication. Data often contains sensitive information and should be protected as mandated by various organizational policies and legal regulations. Encryption is a commonly adopted approach to assure data confidentiality. Encryption alone however is not sufficient as organizations often have also to enforce fine-grained access control on the data. Such control is often based on security-relevant properties of users, referred to as identity attributes. In the proposed scheme, the cloud verifies the authenticity of the series without knowing the user's identity before storing data. This methodology also has the added feature of access control in which only valid users are able to decrypt the stored information. The scheme prevents replay attacks and supports creation, modification, and reading data stored in the cloud. It also address user revocation. Moreover, this authentication and access control scheme is decentralized and robust, unlike other access control schemes designed for clouds which are centralized. This methodology used new alogorithm which act as a digital signature and the algorihm used is MD5 alogithm. MD5 is a message digest authentication algorithm developed by RSA, Inc.. It is an augmented version of the MD4 algorithm. MD5 processes a variable-length message into a fixed-length output of 128 bits. This conversion of input and also it is irreversible once the input data is process into fixed length output it will not get back to input makes it differ and more efficient with more authenticate than any other algorithm.+***

Keyword:- Access control ,authentication, attribute-based encryption, cloud storage, decryption, MD5

I. INTRODUCTION

In cloud computing, users can outsource their computation and storage to servers (also called clouds) using Internet. This frees users from the hassles of maintaining resources on-site. Clouds can provide several types of services like applications (e.g., Google Apps, Microsoft online), infrastructures(e.g., Amazon's EC2, Eucalyptus, Nimbus), and platforms to help developers write applications (e.g.,Amazon's S3, Windows Azure).Much of the data stored in clouds is highly sensitive, for example, medical records and social networks. Security and privacy are, thus, very important issues in cloud computing. In one hand, the user should authenticate itself before initiating any transaction, and on the other hand, it must be ensured that the cloud does not tamper with the data that is outsourced. User privacy is also required so that the cloud or other users do not know the identity of the user. The cloud can hold the user accountable for the data it outsources, and likewise, the cloud is itself accountable for the services it provides. The validity of the user who stores the data is also verified.The confidentiality of the content and the privacy of the users are thus not assured if the identity.

The encryption algorithm is used for the security of the data or information which user want to be kept private or secure from various types of hackers or unauthorized access.

The authentication algorithm computes a digest of the entire data of the message, used for authentication. Typically, the message digest is registered with a trusted third-party, or encrypted via other means. The digest is used by the receiver to verify the contents of a message. It can also be used to encrypt the contents of a message, via a second pass over the data by another algorithm. MD5 requires that both the sender and receiver compute the digest of the entire body of a message. MD5 is used for authentication in a number of

protocols. It is also included as an encapsulation mechanism in SIPP, IPv6, and IPv4. The following is a partial list of protocols or protocol options using MD5.

MD5 is a irreversible alogorithm which is once is digited input and convert into 128 bits fixed output no matters how much input given to the MD5 algorithm whether it is 64 bits or 32 bits it will convert it into fixed 128 bits format. MD5 is irreversible because it will act as encryptor but it will not provide any key which will decrypt the MD5 output.

Decentalized access control for secure storage system uses message digest algorithm for providing authentication. MD5 is a one-way function; it is neither encryption nor encoding. It cannot be reversed other than by brute-force attack.

II. LITERATURE REVIEW AND RELATED WORK

ABE was proposed by Sahai and Waters. In ABE, a user has a set of attributes in addition to its unique ID. There are two classes of ABEs. In key-policy ABE or KP-ABE (Goyal et al.), the sender has an access policy to encrypt data. A writer whose attributes and keys have been revoked cannot write back stale information. The receiver receives attributes and secret keys from the attribute authority and is able to decrypt information if it has matching attributes. In Ciphertext-policy, CP-ABE, the receiver has the access policy in the form of a tree, with attributes as leaves and monotonic access structure with AND, OR and other threshold gates.

All the approaches take a centralized approach and allow only one KDC, which is a single point of failure. Chase proposed a multi authority ABE, in which there are several KDC authorities (coordinated by a trusted authority) which distribute attributes and secret keys to users. Multi authority ABE protocol was studied in, which required no trusted authority which requires every user to have attributes from at all the KDCs. Recently, Lewko and Waters proposed a fully decentralized ABE where users could have zero or more attributes from each authority and did not require a trusted server. In all these cases, decryption at user's end is computation intensive.

So, this technique might be inefficient when users access using their mobile devices. To get over this problem, Green proposed to outsource the decryption task to a proxy server, so that the user can compute with minimum resources (for example, hand held devices). However, the presence of one proxy and one KDC makes it less robust than decentralized approaches. Both these approaches had no way to authenticate users, anonymously. Yang presented a modification of, authenticate users, who want to remain anonymous while accessing the cloud.

To ensure anonymous user authentication ABSs were introduced by Maji. This was also a centralized approach. A recent scheme by Maji et al. takes a decentralized approach and provides authentication without disclosing the identity of the users. However, as mentioned earlier in the previous section it is prone to replay attack.

III. PROPOSED METHOD

This paper proposes a method to create an environment framework which will able to transfer data from data owner to data reader with secure model in which there are 4 types of different users are available which will going to communicate with each other using this system. Four different types of users are as follows

1. Trustee: Legitimate user who is responsible by providing tokens.
2. Creator: Creates data which have to share with other users.
3. Reader: Reads data which is shared by creator.
4. Writer: Reads data which is shared by creator and also able to write.

Working of proposed system can be explained with using an example for a case consider that there are three users, a creator, a reader, and writer. Creator Alice receives a token γ from the trustee, who is assumed to be honest. A trustee can be someone like the federal government who manages social insurance numbers etc. On presenting her id (like health/social insurance number), the trustee gives her a tokenγ. There are multiple KDCs (here 2), which can be scattered. For example, these can be servers in different parts of the world. A creator on presenting the token to one or more KDCs receives keys for encryption/decryption and

signing. In the Fig. 1, SKs are secret keys given for decryption, Kx are keys for signing. The message MSG is encrypted under the access policy X.

The access policy decides who can access the data stored in the cloud. The creator decides on a claim policy Y, to prove her authenticity and signs the message under this claim. The ciphertext C with signature is c, and is sent to the cloud. The cloud verifies the signature and stores the cipher text C. When a reader wants to read, the cloud sends C. If the user has attributes matching with access policy, it can decrypt and get back original message. Write proceeds in the same way as file creation.

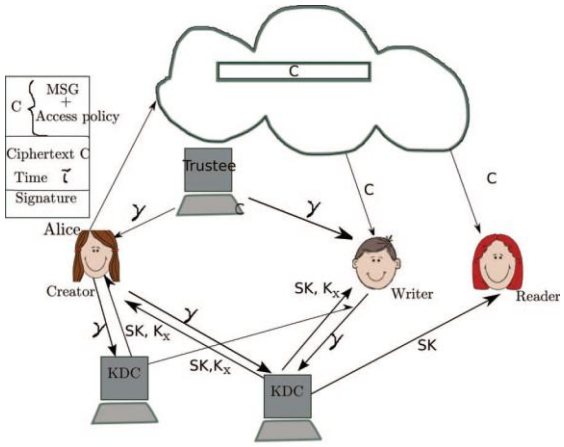

Figure. 1. secure cloud storage model

By designating the verification process to the cloud, it relieves the individual users from time consuming verifications. When a reader wants to read some data stored in the cloud, it tries to decrypt it using the secret keys it receives from the KDCs. If it has enough attributes matching with the access policy, then it decrypts the information stored in the cloud.

In this paper, we put one more algorithm rather than AES and that is MD5 which is used for authentication purpose. The MD5 message-digest algorithm is a widely used cryptographic hash function producing a 128-bit (16-byte) hash value, typically expressed in text format as a 32-digit hexadecimal number. MD5 has been utilized in a wide variety of cryptographic applications and is also commonly used to verify data integrity.

MD5 can be used to store a one-way hash of a password, often with key stretching. Along with other hash functions, it is also used in the field of electronic discovery, in order to provide a unique identifier for each document that is exchanged during the legal discovery process. This method can be used to replace the Bates stamp numbering system that has been used for decades during the exchange of paper documents.

MD5 processes a variable-length message into a fixed-length output of 128 bits. The input message is broken up into chunks of 512-bit blocks (sixteen 32-bit words); the message is padded so that its length is divisible by 512.

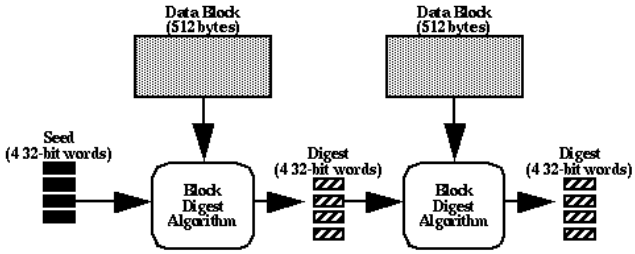

Figure 2.: MD5 block-chained digest algorithm

Check-summing is a well known method for performing integrity checks. Checksums can be computed for disk data and can be stored persistently. Data integrity can be verified by comparing the stored and the newly computed values on every data read. Checksums are generated using a hash function. The use of hash functions has become a standard in Internet applications and protocols.

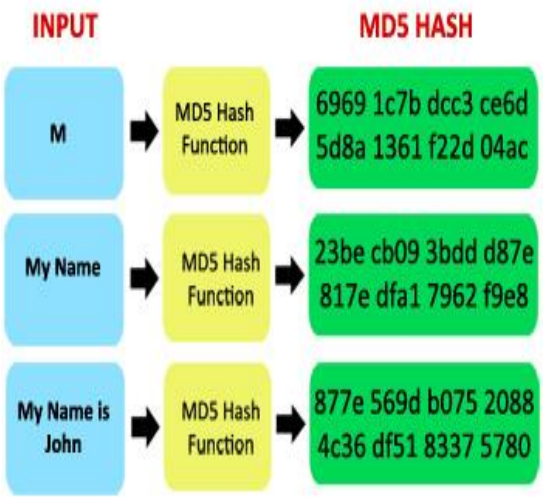

Figure 3: Unique MD5 Hash for each input

Hash functions map strings of different lengths to short fixed size results. These functions are generally designed to be collision resistant,

which means that finding two strings that have the same hash result should be infeasible. In addition to basic collision resistance, functions like MD5 (Message digest) have some properties like randomness. In this project, MD5 hashing algorithm is used to generate Checksum MD5 hashing algorithm is one way encryption in which we able to do only encryption decryption is not possible for MD5 output. MD5 output is unique for each unique input given to Message digest algorithm.

In above figure 2 it shows variable length input like “M”, “My Name” and “My Name is Jhon” and it is given to input for MD5 and it will produces fixed length output and it will be unique for all unique input. In message digest algorithm upper case and lower case input are considered as different characters so characters “My name” is differ from “My Name” [4].

IV. PERFORMANCE ANALYSIS

In this paper, we explored the performance of the popular hash function MD5 — Message Digest Algorithm Version 5. Firstly we introduce the algorithm in detail. Then we check the algorithm’s complexity in two aspects: computational and space complexity. Furthermore we check the security aspects of MD5: whether it is feasible to find collision it using brute force, birthday attack, and cryptanalysis. Available cryptanalysis methods are introduced so that we can find out why MD5 is not considered secure any more.The computational complexity of MD5 can be seen from the following diagram:

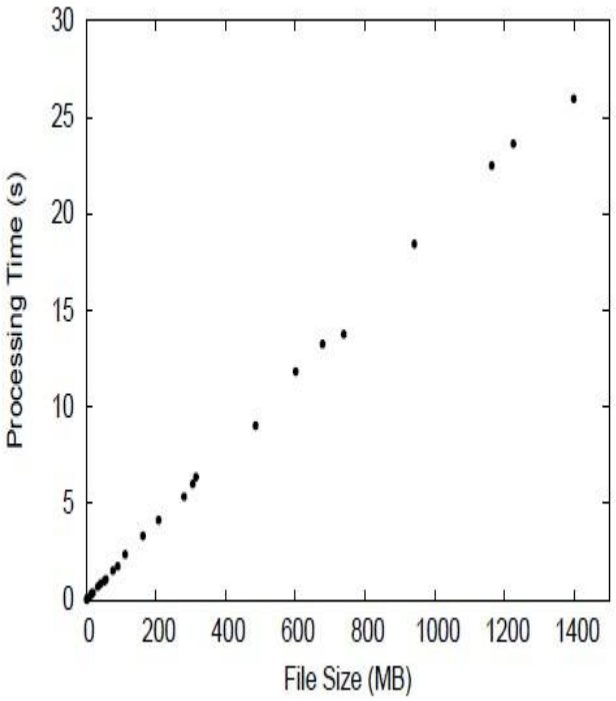

Figure 4.: Performance analysis of MD5

MD5 as per the above figure is more fesible and efficient algorithm we make this decision after checking computational and space complexity.

V. CONCLUSION

Method have presented a decentralized access control technique with anonymous authentication using message digest algorithm, which provides user revocation, prevents replay attacks and because of use of MD5 algorithm it is more authanticated. The cloud does not know the identity of the user who stores information, but only verifies the user’s credentials. Key distribution is done in a decentralized way . MD5 also act as a identity checker of users digital signature. And then allow the user to use the information from clousd storage for the particular authenticated user as per access policy .

Acknowledgment

My thanks to the Guide, Dr. A.B.Raut and Principal Dr. A.B.Marathe, who provided me constructive and positive feedback during the preparation of this paper.

References

[1] Sushmita Ruj, Member, IEEE, Milos Stojmenovic, Member, IEEE, and Amiya Nayak, Senior Member, IEEE "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 Access Control with authentication for Securing Data in Clouds," Proc. IEEE/ACM Int'l Symp.Cluster, Cloud and Grid Computing, pp. 556-563, 2012.

[3] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward Secure 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 Int'l Conf. Financial Cryptography and Data Security, pp. 136-149, 2010.

[6] H. Li, Y. Dai, L. Tian, and H. Yang, "Identity-Based Authenticationfor Cloud Computing," Proc. First Int'l Conf. Cloud Computing(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 Int'l 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] S. Jahid, P. Mittal, and N. Borisov, "EASiER: Encryption-Based Access Control in Social Networks with Efficient Revocation," Proc. ACM Symp. Information, Computer and Comm. Security (ASIACCS), 2011.

[11] R.L. Rivest, A. Shamir, and Y. Tauman, "How to Leak a Secret," Proc. Seventh Int'l Conf. Theory and Application of Cryptology and Information Security (ASIACRYPT), pp. 552-565, 2001.

[12] X. Boyen, "Mesh Signatures," Proc. 26th Ann. Int'l Conf. Advances in Cryptology (EUROCRYPT), pp. 210-227, 2007.

[13] Atkinson, R., "IPv6 Authentication Header," (working draft - draft-ietf-ipngwg-auth-00.txt), February 1995.

[14] Atkinson, R., "IPv6 Security Architecture," (working draft - draft-ietf-ipngwg-sec-00.txt), February 1995.

[15] Atkinson, R., "IPv6 Encapsulating Security Payload (ESP)," (working draft - draft-ietf-ipngwg-esp-00.txt), February 1995.

[16] Baker, F., and Atkinson, R., "OSPF MD5 Authentication," (working draft - draft-ietf-osp5-md5-03.txt), March 1995.

[17] Baker, F., and Atkinson, R., "RIP-II Cryptographic Authentication," (working draft - draft-ietf-ripv2-md5-04.txt), March 1995.

[18] Bradner, S., and Mankin, A., "The Recommendation for the IP Next Generation Protocol," RFC 1752, Harvard University, USC/Information Sciences Institute, January 1995.

[19] Deering, S., "Simple Internet Protocol Plus (SIPP)," (working draft - draft-ietf-sipp-spec-01.txt), July 1994.

Author Profile

Rani Shriram Joshi, received the B.E.degree in Information Technology from H.V.P.M's College Of Engineering And Technology, Amravati in 2014. She is currently pursuing Master's Degree in Computer Science and Engineering from H.V.P.M's College of Engineering And Technology, Amravati.

Dr.Anjali.B.Raut, received the B.E.and M.E degree in Computer Science from Prof. Ram Meghe Institute of Technology, Badnera and PHD from Sant Gadge Baba Amravati University. She is currently working as Head Of Department in Computer Science and Engineering at H.V.P.M's college of Engineering and Technology, Amravati.