RSA and AES Based Secure Data Sharing in Cloud Based Environment

Selvamani K[1] and Velumadhava Rao R[2]

***Abstract*—**Nowadays, Data Outsourcing and Data Backups are maintained by third-party cloud storage service providers so as to reduce the cost for data management. However, Security is a major concern for the outsourced or backup data, since it is maintained by a third-party cloud services. In our proposed protocol, we introduce a mechanism using RSA and AES combination and with access policy control to achieve a secure sharing of data. Also our protocol is built upon with a set of cryptographic key mechanisms which are independent of third-party cloud services.

***Keywords*—**authentication, confidentiality, cloud storage, data sharing, key generation.

I. INTRODUCTION

One of the main characteristics of cloud computing is to use the cloud services in a pay-as-you-go manner [1]. Also cloud offers an infinite storage space for client to store their data. Thus cloud storage provides the way for remote data backup, so that user can able to retrieve the data at any time using the cloud services. Cloud also reduces the financial overload of enterprises and organizations in maintaining their data. There are more case studies that are related to cloud storage for remote data backup [2]. Also individuals can store their personal data to the cloud using Dropbox and Google Drive etc [3, 4]. Nowadays more number of peoples are using tools like Dropbox to store their data in cloud. However, we need to consider the security concerns in storing the sensitive data in cloud which is maintained by third party cloud services. In our proposed work, two security issues are considered particularly. First, we need to ensure that only authorized parties have access to the outsourced data in cloud through efficient key distribution mechanism and access policy. Second, to guarantee secure data access we need to implement cryptography schemes for providing security when users upload/download data from cloud services. In this paper, we used RSA and AES algorithm for achieving the proposed issues. We also perform several cryptography key operations to protect the data which is accessed from the cloud. The proposed protocol is applicable for general storage backups where upload/download of data takes place with the help of backend interface.

Several studies [5] are related to the protection of outsourced data using cryptographic techniques. Wang et al. [6] proposed an auditing system that enables the users to verify the integrity of outsourced data. Wang et al. [7] came up with a secure outsourced data access mechanism with access rights. Yun et al. [8] discussed about the integrity and privacy on an outsourced data using hash-based mechanism. The key shares are stored in Hash Table in a distributed manner. RSA is the most popular public key algorithm. Rivest, Shamir and Adleman [9] invented this algorithm where both public and private key is used for encryption and decryption. All the messages are encrypted using the public key and it is sent to the receiver. The receiver uses the private key to decrypt the message. Yellamma et.al[11] proposed a method to secure data in cloud using RSA. Joan Daemen and Vincent Rijment [10] invented AES a symmetric algorithm. AES uses the same key for both encryption and decryption of messages.

The remaining paper is organized as follows: in section 2, we discuss the proposed works. In section 3, we describe the implementation details and evaluate the performance of our proposed work. Finally, section 4 gives the conclusion of our work.

II. PROPOSED WORK

In our proposed work, we introduce the method of sharing the data in a cloud environment and also securing the data. We made the following contributions Secure Data Sharing, Confidentiality and Authentication. The proposed architecture model is shown in Fig. 1. The following processes were employed in our proposed system.

The following entities are involved in our proposed method

Cloud: Data outsourcing and Data backup are carried out in cloud by the data owner. To protect the data from the unauthorized user, data is stored in a form of encryption in cloud. Confidentiality is achieved by storing the data in an encrypted form. The cryptographic operation and also the upload and download file operation are done using our proposed method. So there is no much involvement of cloud specific operation in our work.

KDM: The Key Distribution Manager (KDM) is acted as trusted third party where all the cryptographic related operations are carried out here. ACL is also maintained in KDM for storing policy for individual files. Whenever user wants to upload or download the file in cloud, first the user has to register with KDM and KDM verifies for authentication. KDM can be maintained by the organization itself to generate trust for the users who are accessing the data.

Data Owner: Data owner will upload the encrypted data to the cloud with the help of KDM. Data Owner sends the Access Policy (A_{Pi}) which is associated with the file along with the list of users who can access this file to KDM. This set of access

[1]Assistant Professor, Anna University, CEG, Chennai
[2]Research Scholar, Anna University, CEG, Chennai.

policies are maintained in ACL file which is part of a KDM. All access policies are associated with each file and the authorized users are allowed to access the files.

Cloud User: Group of users who want to access the file will first registers with the KDM by sending their Group ID. After the verification of group users, the KDM will authenticate the list of users for accessing the file.

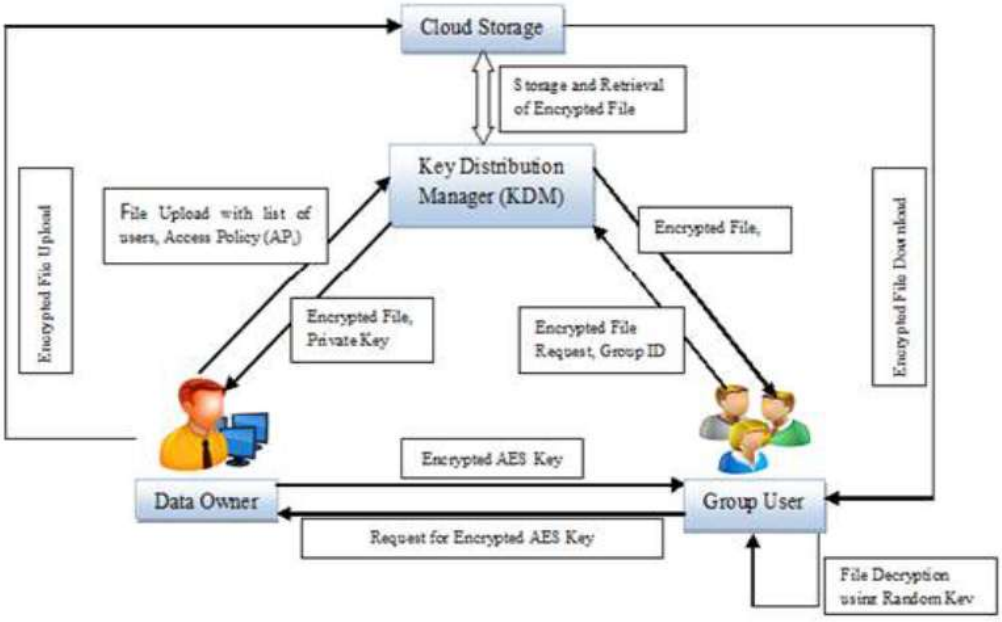

Fig. 1: Proposed system architecture

2.1 User Registration

Data Owner and every other individual user have to first register with KDM. KDM keeps track of list of user and their Group ID (IDi) which is provided by the group members while registration. During the process of encryption and decryption, KDM verifies and authenticate the list of users who wish to access that file.

2.2 Key Generation

The KDM is responsible for RSA key generation of our method which involves both Symmetric and Asymmetric algorithms. In our proposed work, we use RSA algorithm to generate the key pair which is public and private key. The following steps are carried out in the generation of P_U and P_R.

1. Two large prime number x and y are chosen randomly.

2. Compute z = x * y

3. Compute the function f(z) = (x-1) * (y-1)

4. A number e is chosen randomly such that the range as 1 < e < f(z) and also GCD(e, f(z) = 1.

5. With these parameter private key (d) is generated using the equation

$$d= e-1 \bmod f(z) \quad (1)$$

2.3 Encryption

Data owner send the file to KDM for encryption. Encryption is carried our using the Symmetric algorithm AES using the below equation 2.

$$C1 = E_{AES} (File, R_k) \quad (2)$$

The KDM generates the random key (R_k) of the length required for file encryption. Now the file is an encrypted form. The C1 is stored in the cloud by Data owner after it is received from KDM. After the process of file encryption, the AES key generated is encrypted using the public key of RSA using equation (3)

$$C2 = ERSA(R_k, KPu) \quad (3)$$

KDM send the value of C1 (Encrypted File), C2 (Encrypted AES Key) and private key KPr to the Data owner for decryption. The overall process for secure data sharing is shown in Fig. 2. The procedure for the implementation of AES and RSA based encryption and decryption is shown in [12]

2.4 Decryption

When the group member wants to access the file, either the user can download the encrypted file from the cloud or the request can be made to KDM for retrieving the encrypted file. The KDM sends the encrypted file and the private key pair (KPr) to the Data Owner. After verifying the authenticity of the user the KDM sends the Encrypted file to the group members. Once the file is received by the user, then a request is send to the data owner for AES key to decrypt the file. The Data owner uses the private key (KPr) and C2 value received from KDM to regenerate the AES key (Rk) using the equation (4)

$$KAES = DRSA(C2, KPr) \quad (4)$$

As per the request provided by the user and also with respect to user access policy, the generated AES key (KAES) is sent to the group of users. The group members after receiving the AES key will able to decrypt the file using the equation (5)

$$File = DAES(C1, KAES) \quad (5)$$

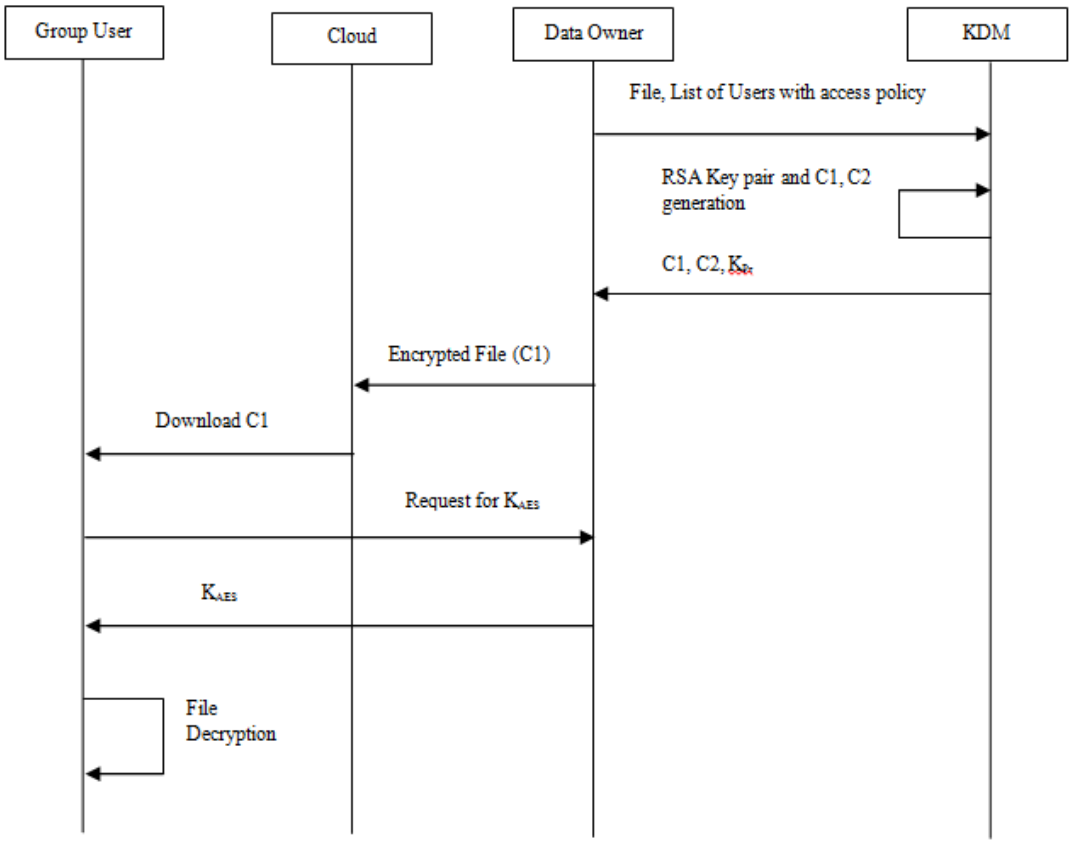

Fig. 2: Process of Proposed Method

2.5 Policy Based Data Access

Every individual file and the users are associated with single access policy. During the file upload the data owner sends the file along with the policy associated with the file and list of users who can access the file. Every individual users needs to access the file is with a same or different policy as it decided by the data owner. The KDM authenticates the users during the file download operation. The policy can vary with different applications. Suppose when there are several employees working in a organization and each one involved in different project we can associate a user-based policy will the files for the user. Whenever the employee leave the organization the KDM will generate different RSA key pair and will re-encrypt the AES key using the new public key. AES key is also regenerated before encryption by RSA public key if necessary. Then KDM will send the new private key to the data owner. The data owner will send the newly generated private key to the group members who have access policy to access that file. The member of group who left will not have any access to that data and the policy associated with the users is also deleted from ACL which is part of KDM. The new user first registers with the KDM. While joining user submits his identity for authentication. When user wants to join the existing group, he sends a request to KDM by providing the group ID for verification. Once KDM verifies the user and a request to add the new member has been sent by the data owner, then KDM adds the member to the group which is reference by group Id. Also ACL update the new user information and the policy is associated to the user with the file that has to be accessed. When the data owner needs to renewal with the new policy, sends a request to KDM to update the policy associated with the file. The KDM updated the ACL list according to the new policy and if needed it again regenerates the new pair of RSA public and private key. When a user is associated with different file and since each file is linked with a single policy, KDM will make use of the Boolean combination of policies for that user.

III. Performance Evaluation

This section evaluates the performance of our proposed system. The proposed method is experimented and implemented in java. For our experiment, we used Google App Engine as cloud storage. We used Google Cloud SQL as a database to store and retrieve data anytime and anywhere. In our proposed work, we use both symmetric and asymmetric key for maintain security. Using RSA key pair generated, AES key is encrypted. The testing has been carried out with multiple key length. Fig.3 and Fig.4 shows that the time taken to generate the private key and public key increases with respect to the increase in the key length.

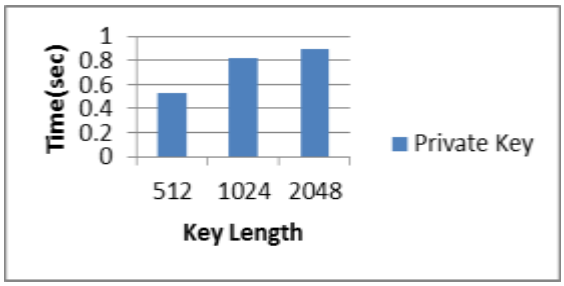

Fig. 3: RSA Private Key Generation Time

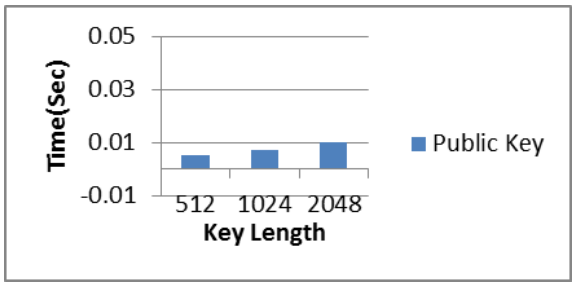

Fig.4: RSA Public Key Generation Time

Our evaluation is based on the time taken to upload and download the file. Fig.5 and Fig.6 indicates the increase in time with respect to increase in the file size for both upload and download.

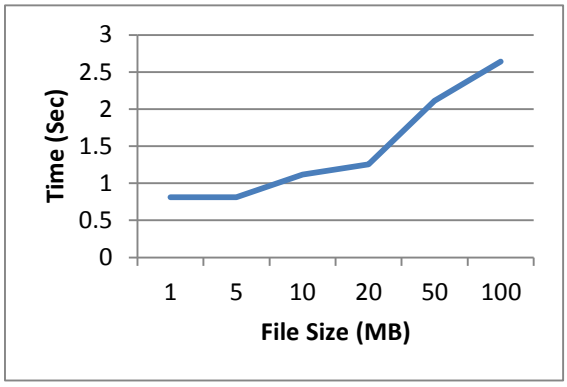

Fig. 5: File upload with respect to time

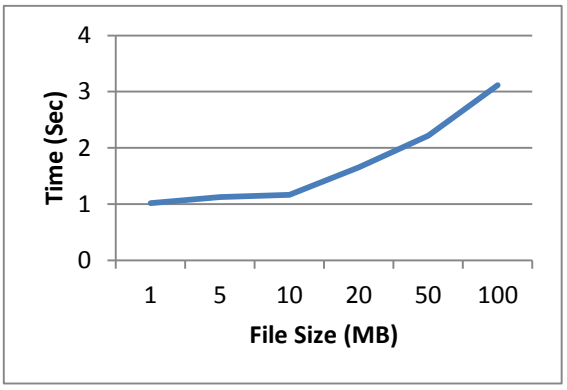

Fig. 6: File download with respect to time

IV. Conclusion

We proposed a secure sharing of data using RSA and AES algorithm to maintain security within cloud server. KDM will be responsible for all key generation and key distribution process in our proposed scheme. The performance is evaluated and the results are obtained based on RSA key generation and AES encryption process. From the result, it is noticed that our proposed method will be applicable for sharing data in cloud securely. We use policy based access mechanism to provide security with the data in cloud and also to provide authentication. In future we can use multiple KDM to handle the data with different access policies to avoid insider attacks.

References

[1] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A View of Cloud Computing." Comm. ACM, vol. 53, no. 4, pp. 50-58, Apr. 2010. https://doi.org/10.1145/1721654.1721672

[2] Amazon, "Case Studies," http://aws.amazon.com/solutions/ case-studies/#backup, 2012.

[3] Dropbox, http://www.dropbox.com, 2010.

[4] Google Drive, http://www.drive.google.com, 2012

[5] S. Kamara and K. Lauter, "Cryptographic Cloud Storage," Proc. 14th Int'l Conf. Financial Cryptography and Data Security, 2010. https://doi.org/10.1007/978-3-642-14992-4_13

[6] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Storage Security in Cloud Computing," Proc. IEEE INFOCOM, Mar. 2010. https://doi.org/10.1109/infcom.2010.5462173

[7] W. Wang, Z. Li, R. Owens, and B. Bhargava, "Secure and Efficient Access to Outsourced Data," Proc. ACM Workshop Cloud Computing Security (CCSW), Nov.2009. https://doi.org/10.1145/1655008.1655016

[8] A. Yun, C. Shi, and Y. Kim, "On Protecting Integrity and Confidentiality of Cryptographic File System for Outsourced Storage," Proc. ACM Workshop Cloud Computing Security (CCSW), Nov. 2009. https://doi.org/10.1145/1655008.1655017

[9] R.L.Rivest, A.Shamir, and L.Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," Communication of the ACM, Volume 21 No. 2, Feb. 1978. https://doi.org/10.1145/359340.359342

[10] Daemen, J., and Rijmen, V. "Rijndael: The Advanced Encryption Standard." Dr. Dobb's Journal, March 2001.

[11] Yellamma, P. , Narasimham, C. , and Sreenivas, V. , " Data security in cloud using RSA ", Proceedings of 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 4-6 July 2013. https://doi.org/10.1109/icccnt.2013.6726471

[12] https://alperkaratepe.wordpress.com/2011/08/18/encrypting-decrypting-data-files-by-using-aes-and-rsa-algorithms/