International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 04, April 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

DATA SECURITY IN CLOUD COMPUTING USING THRESHOLD CRYPTOGRAPHY

Nihal Karale[1], Shrikant Kendre[2], Akshay Koli[3], Likhit Pande[4]

JSPM's Rajarshi Shahu College of Engg. Pune, Maharashtra

ABSTRACT

Currently cloud computing is extremely in style in giant and little scale organization because it can hold on giant amount of knowledge and supply low price service. Thus it has daily new challenges to produce secure authorization, integrity and access management. Some approaches are guaranteeing concerning security however there are some limitations to these approaches and problems. To resolve this issue we have proposed a scheme named threshold cryptography within which information from owner will be divided among its users in cluster and partial key will be shared with all users in the cluster. The partial key will be used by the user for decryption. The proposed scheme uses capability list to control the access. This proposed scheme not only provides the sturdy information confidentiality however additionally reduces the quantity of keys.

Index terms: Outsourced data, malicious outsiders, access control, authentication, capability list, threshold cryptography.

I. INTRODUCTION

As an end user, cloud computing lets you run software applications and access data from any place and time, and from any computer; without the need to ever install, upgrade, troubleshoot software applications physically on a local desktop or server. Cloud computing is a quick growing and new technology in field of computation and storage of information. It provides storage and computing as a service at terribly engaging cost. It provides services consistent with three basic service models: infrastructure as a service (IaaS), platform as a service (PaaS) and software system as a service (SaaS). Storage as a service is essentially a platform as a service. The few characteristics of cloud computing are: on-demand service, self service, location independent, more elasticity and measured scale service.

These characteristics create cloud important. Industries and establishments area unit exploiting these characteristics of cloud computing and increasing their profit and revenue [1]. That's why, industries area unit shifting their businesses towards cloud computing. However, information security may be a major obstacle within the approach of cloud computing. Folks are still fearing to use the cloud computing. Some folks believe that cloud is unsafe place and once you send your information to the cloud, you lose complete privacy over it [8][9]. They're somewhere right. Data of information owners are kept and processed at external servers. So, confidentiality, integrity and access of information become additional vulnerable. Since, external servers area unit are operated by commercial service suppliers, information owner cannot trust on them as they'll use this information for his or her edges and might spoil businesses of information owner [4]. Knowledge owner even cannot trust on users as they will be malicious. Information confidentiality could violet through collusion attack of malicious users and repair suppliers.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

Many schemes are given to confirm these security necessities however they're littered with collusion attack of malicious users and cloud service supplier and significant computation (due to massive no keys). To handle these problems we have a tendency to propose a scheme. During this scheme, there are primarily three entities: Data Owner (DO), Cloud Service Provider (CSP) and Users. Users are divided in terms on some basis like location, project and department and, akin to every cluster, there's one key for encryption and decryption of information. Every user within the cluster shares components of the key. Information is decrypted once a minimum of threshold number of users are present. This scheme not solely provides information confidentiality by all means that however conjointly reduces the amount of keys. To get fine-grained information access control, the approach has used capability list [6]. It's primarily row-based decomposition of access matrix. In capability list operations for a user and authorized data are specified. It is better suit than Access Control List (ACL) [5][10][16] it is because ACL gives users and their allowable operation for every information and file. It is much inefficient that two users need same information and have same operations on that. During this paper, the approach has used the changed Diffie-Hellman algorithmic rule to come up with only one shared session-key between CSP and user to guard the information from outsiders. To confirm data integrity the approach has used MD5 [4].

II. RELATED WORK

Data confidentiality and access control are two basic security necessities for outsourced information in cloud computing. Sometime, after we emphasize more on security of information, we tend to ignore performance of systems (DO, CSP, users). as an example, to secure information, we tend to sometime use too many keys. we all know that keys are confidential, thus there's have to be compelled to secure and maintain these keys that are extra work. These extra works have an effect on the performance of the system. So, it is fascinating to scale back no of keys. So, there is need a scheme that has not only information security however conjointly maintain the performance. several schemes are recommended to satisfy these necessities.

The scheme proposed in [13] is the group-key scheme. In group-key scheme, there's one key to every cluster of users for decryption method and all users of the cluster know that key. Here, range of keys is reduced however there's a problem of collusion attack of CSP and a user because a single malicious user can leak whole information of the cluster to CSP. we all know that CSP is not trustworthy party. It will use data owner's information for its business advantages.

The scheme proposed in [4] tried to achieve information confidentiality and access control. in this scheme, information are encrypted by symmetrical/bilaterally symmetric keys and symmetric keys are known solely to data owner and corresponding data users. The encrypted information are stored at CSP. CSP cannot see information stored at it as information are encrypted. information are further encrypted by one time secrete session-key shared between CSP and user by the changed Diffie-Hellman protocol to safeguard information from outsiders during the transmission between CSP and user. This scheme no doubt provides whole data security however there's associated a key corresponding to every user and users could also be massive in number in some applications. So, number of keys might increase. Hence, increases the maintenance and security issues of keys Communication model of the proposed scheme somehow matches with it [4] however proposed scheme is safer and reduces number of keys. The proposed scheme is helpful for those applications wherever works are done in team and group such as in software industries. you may assume proposed scheme has limited

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

applications however it is not as such. it's applicable all where you can group users on some basis and can apply threshold cryptography technique. such as software and hardware industries, institutes, banks and medicals fields. there is provision of hierarchy of access in this scheme which makes this scheme more helpful and realistic. for example, an university has vice-chancellor, hods, teachers, staff and students. each has totally different level of access right.

III. MODEL AND ASSUMPTIONS

We suppose that our model is composed of three entities: a CSP, a DO and many users related to DO. Initially, all users are registered at DO. during registration users send their credentials to DO. we tend to assume that user's credentials are sent securely to DO. DO then divides users in groups and provides encryption keys, tokens, algorithm (MD5) and other necessary things for secure communication to user groups in response of registration. A user can get information from CSP in a very confidential manner after successful authentication of himself at CSP. we assume that CSP includes a massive capacity and machine power. we additionally assume that nobody can breach the security of CSP. further we assume that the algorithm which is used to generate the secrete keys for encryption, is secure at DO. DO has storage capacity to store some files and information and, he can execute programs also at CSP to manage his files and information. we are using modified Diffie-Hellman and public key cryptography to secure communication between CSP and user. modified Diffie-Hellman protocol is used to create one time session-key between CSP and user. Fig.1 illustrates the secure communication between entities in the proposed scheme.

IV. TECHNOLOGIES USED

This is a complete model for secure communication between different entities and secure access to data. There are four algorithms in the proposed scheme. Algorithm 1 describes secure communication of data between DO and CSP moreover this algorithm insures data confidentiality and, authentication of DO and CSP. Algorithm 2 describes procedures which DO and CSP apply after a new file creation in respect. Algorithm 3 describes about secure communication of data between CSP and user. In this algorithm user's authorization is also checked. Algorithm 4 describes the threshold cryptography technique for decryption of a user's file. Algorithm 4 is applied at user side where number of keys is reduced (one key corresponding to one group) and no threat of collusion attack as in group-key scheme.

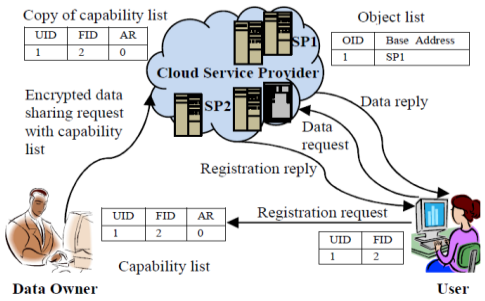

Fig: System Architecture

To understand proposed scheme better we take an example of real life scenario, DO may be a software industry who stores its data on to the CSP and the users may be its employees who view their data from the CSP. DO

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

divides users in groups on some basis such as project basis and encrypts the data of each group with a single symmetric key (KT) and, it gives parts of the symmetric key (KT) to each user of the group. DO computes digest of data by using 128-bit MD5 hash algorithm and then encapsulates the digest and data using the symmetric key (KT). This in turn, provides strong data confidentiality and integrity. DO then fills the entries such as UID, FID and AR in Capability List corresponding to each new user. DO then encrypts Capability List and encapsulated things with its private key after that public key of CSP and, then sends all things to CSP. These encryptions ensure confidentiality and authentication between DO and CSP.

Algorithm 1: Procedure to be followed by CSP after getting encrypted File and Capability List from DO

Step 1: CSP stores Encrypted Data and Capability List which are received from DO Array ← Rece(EkPuCSP(EkPrDO (Ekkt (Fi)) || (CPList))
CPList || Ekkt (Fi) ← DkPrCSP(DkPuDO((Array))

Step 2: CSP updates the Encrypted File List Encptd. File List ← Encptd. File List (FID, Base Adds.)

Step 3: CSP updates Capability List CPList ← CPList(UID, FID, AR)

Algorithm 1 describes the process what CSP do after getting encrypted data and Capability List from the DO. CSP decrypts the message using its own private key and the public key of data owner and stores the encrypted data and Capability List in its storage. CSP then updates the encrypted File List and Capability List. Since, data are encrypted using symmetric key (KT) which is known only to DO and respected user group, CSP can't see data even though user's credential comes through it.

Algorithm 2: Procedure to be followed after a new File creation

Step 1: DO updates Capability List CPList ← Add.(CPList, (UID, FID, AR))

Step 2: Now, DO encrypts the CPList, Encrypted File, symmetric key and sends these to the CSP Send (EkPuCSP(EkPrDO(CPList, (Fi), EkPrDO (EkPuUSR(KT, N+1, TimeStamp)))))

Step 3: CSP Updates its copy of the Capability List, Encrypted File List and sends symmetric key to indented user group Send (EkPuUSR(EkPrDO(EkPuUSR(KT,N+1,TimeStamp))))

Step 4: Now, the user can send actual access request for that File directly to CSP

Algorithm 2 illustrates the procedure required after a new File creation. When a new File is created, DO fills entries for that File in Capability List containing UID, FID and AR. DO generates a symmetric key (KT) and encrypts File with that symmetric key (KT). Now, DO encrypts the updated CPList, Encrypted File and symmetric key (KT) with its private key after that public key of CSP and sends these to the CSP. When CSP receives these, it updates Capability List, Encrypted File List and sends encrypted symmetric key (KT) to respective user group. Users of the user group then decrypt the message and get their own parts of the

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

symmetric key (KT). To avoid man-in-middle and replay attack we use nonce and timestamp in each message. After getting the details, user can request to CSP for data.

Algorithm 3: Algorithm for secure data exchange between CSP and User by using Modified D-H key exchange

Step 1: User sends data access request to CSP Send (UID, FID, AR))

Step 2: CSP matches UID, FID, AR with CPList stored at it. If(match)
Go to step (3) else
Go to step (6)

Step 3: CSP initiates D-H exchange with that User and shares one time shared session key(KS)

Step 4: CSP encrypts the encrypted File with shared session key and sends it to User Send (((Fi)))

Step 5: User decrypts the File and calculates the message digest of that File If Calculated digest matches with stored digest then File is original else File is modified and User sends Error Notification to DO

Step 6: CSP sends 'invalid request' message to User

Algorithm 3 describes how data are exchanged securely between CSP and the user by use of modified Diffie-Hellman algorithm. We called it modified D-H algorithm as we encrypt the D-H parameters using the public key of one side and, using nonce in each direction during session key (KS) generation and data transfer. It helps to counter the man-in-the middle attack. After available of keys and tokens, the user may request for data to CSP. CSP initiates modified D-H key exchange with the user, if request is authentic. We assume that the session key (KS) is shared between CSP and the user by modified Diffie-Hellman algorithm. Now, CSP encrypts the encrypted File (Fi) and its digest (Di) with the shared session key (KS) and sends it to the user. This over encryption ensures the confidentiality of the message between cloud service provider and the user. The user then decrypts the message (user decrypts the message according to algorithm 4) and calculates the digest of File and then matches it with stored digest. If digest matches, File is original otherwise File is modified by outsiders and user then sends an error notification message to DO.

Algorithm 4: Algorithm for Decryption of a File for User 1

Step 1: User 1 receives Encrypted File M ← Rece.((Fi))

Step 2: Initially, all bits of PKS Vector is zero. Here, PKS Vector indicates parts of the key. User 1 will update this PKS Vector with the components he has PKS = PKSOR A1

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

Step 3: User 1 forwards M and PKS to ith user of the group. Who will decrypt it and update the PKS Vector M = (M) PKS = PKSOR Ai The ith user then forwards M and PKS to next user in the group. The next user performs same operations as ith user did. This process is continued.

Step 4: if (PKS = = 11111.....up to d bits)

Go to step (5)

else

Go to step (3)

Step 5: Forward M to the User 1 User 1 then decrypts message (M) and get File Fi = (M)

Algorithm 4 which resembles the threshold cryptography technique, describes the procedure how a File is decrypting for User 1. After getting encrypted message, user's main concern how to decrypt it because he alone can't decrypt. So, he first updates PKS Vector (Initially, all bits of it are zero) with his key component and then sends PKS Vector and encrypted message to next user of same group. The next user then decrypts the message and updates the PKS Vector with his key component. This is continuing until all bits of PKS Vector are one. Here, we can see that application is not using all key components (Only threshold no of key components). After this, data are sent back to initiator user. Initiator user then decrypts the message and gets it. Initially, User 1does not decrypt message (M), he just updates the PKS Vector.

V. CONCLUSION

In this paper, we presented a new approach which provides Security for data outsourced at CSP. Some approaches are given to secure outsourced data but they are suffering from having large number of keys and collusion attack. By employing the threshold cryptography at the user side, we protect outsourced data from collusion attack. Since, DO stores its data at CSP in encrypted form and, keys are known only to DO and respected users group, data confidentiality is ensured. To ensure fine-grained access control of outsourced.

REFERENCES

[1] Proxy Re-encryption for Data Confidentiality in Cloud Computing Environments," Computers, Networks, Systems and Industrial Engineering (CNSI), 2011 First ACIS/JNU International Conference on, vol., no., pp.248-251, 23-25 May 2011.

[2] A. Shamir, “How to share a secret," Communications of the ACM, v.22 n.11,p.612-613,Nov.1979.[Online].Available: http://portal.acm.org/citation.cfm?id=359168.359176.

[3] N. Bennani, E. Damiani, and S. Cimato, “Toward Cloud-Based Key Management for Outsourced Databases," Computer Software and Applications Conference Workshops (COMPSACW), 2010 IEEE 34th Annual, vol., no., pp.232-236, 19-23 July 2010.

[4] S. Sanka, C. Hota, and M. Rajarajan, “Secure data access in cloud computing," Internet Multimedia Services Architecture and application (IMSAA), 2010 IEEE 4th International Conference on, vol., no., pp.1-6, 15-17 Dec. 2010.

[5] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, “Improved proxy re-encryption schemes with applications to secure distributed storage," in Proc. of NDSS’05, 2005.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

[6] C. Hota, S. Sanka, M. Rajarajan, and S. Nair, "Capability-Based Cryptographic Data Access Control in Cloud Computing," Int. J. Advanced Networking and Applications Volume: 01 Issue: 01 Page: (2011).

[7] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," Association for Computing Machinery, in Proc. of CCS'06, 2006

[8] T. Mather, S. Kumaraswamy, and S. Latif, "Cloud Security and Privacy," O'Reilly Media, Sep. 2009.

[9] A. T. Velte, T. J. Velte, and R. Elsenpeter, "Cloud computing a practical approach," Tata McGraw-Hill Edition, 2010, ISBN-13:978-0-07-068351-8.

[10] W. Stallings, "Cryptography and network security," LPE Forth Edition, ISBN-978-81-7758-774-6.

[11] G. Miklau, and D. Suciu, "Controlling access to published data using cryptography," in Proc. of 29th VLDB, Germany, Sept 2003.

[12] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing," in Proc. of IEEE INFOCOM 2010, 2010.

[13] H. Zhong, and H. Zhen, "An Efficient Authenticated Group Key Agreement Protocol," Security Technology, 2007 41st Annual IEEE International Carnahan Conference on, vol., no., pp.250-254, 8-11 Oct. 2007.

[14] S. K. Harit, S. K. Saini, N. Tyagi, and K. K. Mishra, "RSA Threshold Signature Based Node Eviction in Vehicular Ad Hoc Network,"Information Technology Journal, 2012, ISSN 1812-5638, in Asian Network for Scientific Information.

[15] R.S. Fabry, "Capability-Based Addressing," in Communications of the ACM, 17(7), July 1974, pp. 403-412.

[16] S. D. C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Over-encryption: Management of access control evolution on outsourced data," in Proc. of VLDB'07, 2007.