Highly Secure Method based on Ciphertext Policy Attribute based Encryption in Hadoop System

Priyanka Rajput
Research Scholar
Department of Computer Science & Engineering,
Patel College of Science & Technology, Bhopal, MP, India

Pankaj Kawadkar
Professor
Department of Computer Science & Engineering,
Patel College of Science & Technology, Bhopal, MP, India

ABSTRACT
Cloud computing is one of today's most exciting technologies due to its ability to reduce costs associated with computing while increasing flexibility and scalability for computer processes. Cloud computing is set of resources and services offered through the Internet. The rapid growth in field of "cloud computing" also increases severe security concerns. Lack of security is the only hurdle in wide adoption of cloud computing. Distribution and sharing of File is one of the most critical service of cloud computing.

This paper examines security for "big data" environments, reviewing built-in protections and weaknesses of these systems. In this paper, we use the Hadoop based file system and secure it using a highly secure algorithm based on ciphertext policy attributes based encryption with pairing (CP-ABE-WP with pairing). The solution aims at providing optimal performance in cloud environment and also ensures security by using a session based approach. The proposed method has maintained a hierarchy of users to ensure that files are used by their appropriate users.

Keywords
Cloud Storage; Access Control; Attribute-Based Encryption; Document Sharing; Ciphertext-Policy.

1. INTRODUCTION
Cloud storage is a service based on cloud computing technology [1]. Storage virtualization consolidates different storage resources that can be accessed through a single user interface via the Internet without exposing the physical details of the underlying infrastructure. Cloud storage has the capability of providing almost unlimited flexible storage capacity and rapid provisioning to users, as well as dramatically reducing the costs of IT ownership and maintenance.

Data outsourcing to third party cloud storage providers presents a number of issues. In order to provide virtually unlimited storage resources to end users, a cloud storage service usually spans multiple domains. Thus, data from different logical domains may be hosted at the same physical or virtual server, or the data may be segmented and stored on multiple servers across different security domains. Since virtualization hides the details of physical resources, the location of stored data becomes uncertain to users, which has a potential to result in mistrust of cloud storage service providers [2], [3].

From a data security perspective, data owners should take responsibility for protecting their own data. This data owner-centric protection approach typically requires the following characteristics:

a. Fine-grained protection. Data access policy can be defined at data item level. The data access policy should be enforced at each access attempt with or without the data owner's involvement.

b. Dynamic access rights management. The granting or revoking of access rights to a particular item of data is straightforward to conduct and can, ideally, be performed almost instantaneously.

c. Efficient key management. Critical key management operation such as key establishment, key refreshing and key revocation are conducted in an efficient manner that scales well and is appropriate for the highly dynamic and heterogeneous nature of a cloud storage environment.

The emerging data architecture most commonly seen introduces Apache Hadoop to handle these new types of data in an efficient and cost-effective manner. The Emerging Big Data Architecture is Fig 1. Hadoop does not replace the traditional data repositories used in the enterprise, but rather is a complement.

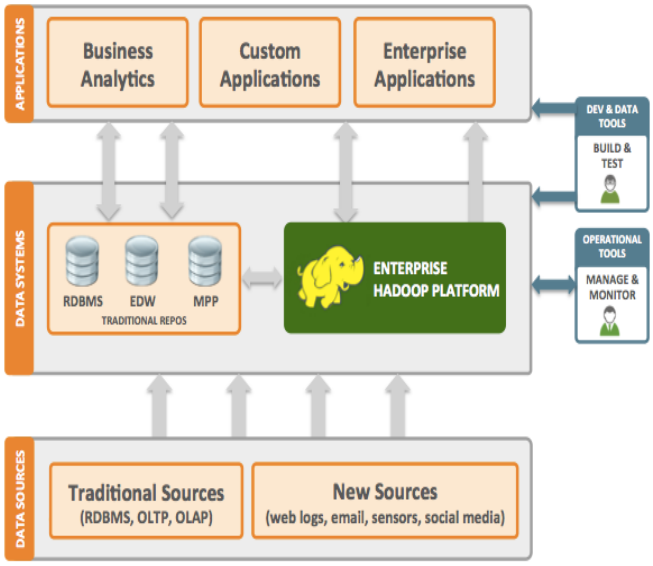

Fig 1: The Emerging Big Data Architecture

Hadoop [4] is an open source large scale distributed file system modeled after the Google File System [5]. It provides extreme scalability on commodity hardware, streaming data access, built-in replication, and active storage. Hadoop is well known for its extensive use in web search by companies such as Yahoo! as well as by researchers for data intensive tasks like genome assembly [6]. A key feature of Hadoop is its resiliency to hardware failure. The designers of Hadoop

expect hardware failure to be normal during operation rather than exceptional [7]. Hadoop is a ”rack-aware” file system that by default replicates all data blocks three times across storage components to minimize the possibility of data loss. Hadoop and Hadoop Distributed File System (HDFS) have a weak security model; in particular the communication between Data nodes and between clients and data nodes is not encrypted. This paper proposes a new technique to improve security in Hadoop environment.

Recently, attribute-based encryption (ABE) has been developed as a cryptographic primitive for the provision of fine-grained access control to encrypted data. In ABE, a set of system attributes are used to define user access rights or data access policies. ABE thus appears to be a promising tool for the protection of data in cloud storage environments. However, existing ABE schemes have some practical limitations with respect to the efficiency and scalability of certain operations that are critical to cloud storage environments. This paper proposes a deployment model for session based CP-ABE-WP with pairing which enables management of access rights. This model can be generically adapted to suit ciphertext-policy ABE (CP-ABE) schemes.

The rest of paper is organized as follows. Section 2 provides a brief background of Cloud Storage Trust Model. Section 3 discussed about ABE and discussion of related work is happened in section 4. The Proposed Method- Session Based Cipher Policy-Attribute Based Encryption is explained in Section 5 and applies it to an existing CP-ABE and proposed SB-CP-ABE scheme in Section 6. Conclusions are drawn in Section 7.

2. CLOUD STORAGE TRUST MODEL

In our model, the attribute authority is the central trusted component that is responsible for generating attribute key shares, publishing system public parameters and maintaining the master secret. The cloud storage provider, which includes a proxy server, is a semi-trusted entity. It is responsible for re-encrypting data owners’ cipher texts before they are sent to users. Data owners are responsible for protecting their data by defining access policies, managing user revocation lists, and encrypting data before it is sent to the cloud storage provider. Users are untrusted entities whose attributes need to comply with the access policy before the data is able to be decrypted. All the communication channels need to be encrypted for data transmission.

3. ATTRIBUTE BASED ENCRYPTION

ABE was first introduced by Sahai and Waters [8]. There are two major classes of ABE schemes. In key policy ABE (KP-ABE) [9], [10] cipher texts are labeled with sets of attributes and private keys are associated with access policies. A user can decrypt a ciphertext if the attributes associated with the ciphertext satisfy the access policy associated with the private decryption key. In ciphertext-policy ABE (CP-ABE) [11], [12], [13] an access policy is associated with each ciphertext. The private decryption key can be reconstructed correctly if a user’s attributes satisfy the access policy. Although KP-ABE and CP-ABE both achieve fine-grained access control, CPABE is more suitable for data-owner-centric protection in outsourcing systems. User can decrypt the cipher text with one attribute and with more than one attribute in without pairing and pairing CP-ABE.

4. RELATED WORK

There has been some prior research into dealing with the practical problems with implementation of ABE, particularly with respect to revocation issues.

In [14] it was suggested that attributes could be associated with an expiry time. This idea was enhanced by

[11] Who suggested associating private key shares with an expiry time. Both the resulting schemes require users to periodically contact the attribute authority for generation of new key shares. This raises potential issues of scalability with both schemes, as well as the problem that user revocation cannot be instantaneous.

In [15] Junod and Karlov constructed a CP-ABE based broadcast encryption scheme that supports direct user revocation. In their scheme, each broadcast receiver’s identity is mapped to an individual attribute. The access policy consists of a set of system attributes with a set of identity attributes. Individual user revocation is achieved by updating the set of identity attributes in the access policy. This scheme is not efficient to apply to cloud storage systems as mapping each user’s identity to an attribute can make the ciphertext growing linearly. In addition, data owners should not directly control the data distribution after the data is stored in a cloud storage system.

Yu et al. [16] proposed a scheme to accomplish revocation of user access rights via attribute revocation. One of their core mechanisms is proxy re-encryption, which was first introduced in [17]. The notion of the proxy re-encryption is to use a proxy to re-encrypt a ciphertext from one secret key to another without learning the underlying plaintext. In the scheme of [16], when a user’s access right is revoked, the attribute authority generates a new re-encryption key for the system’s semi-trusted on-line proxy server. On behalf of the attribute authority, the proxy server generates and distributes new updated attribute key shares for each non-revoked user. Then the proxy server re-encrypts the existing cipher-texts with the new re-encryption key. While this scheme enables instantaneous user revocation, each revocation triggers a round of attribute key share updates and ciphertext re-encryption. This results in it being unsuitable for large data-owner-centric environments.

Jahid et al. in [18] achieved user revocation by utilizing a semi-trusted proxy to participate in the decryption process. In their proposed scheme, each user obtains an identity key in addition to their attribute key shares. The identity keys are generated by a data owner using a secret sharing scheme. The data owner also generates a proxy key for the proxy, who uses the proxy key to transfer the ciphertext in the way such that only non-revoked users with their identity keys can decrypt the data. The proxy key is regenerated whenever a user is revoked. Although the scheme achieves dynamic user revocation without attribute key regeneration, it can only revoke up to a predefined number of users. In addition, adding a new user to the system can trigger the rekeying of existing users’ identity keys, which presents a potential scalability issue in a large or highly distributed environment.

Hur and Noh [19] use attribute key encryption keys (KEKs) to address user revocation in BSW’s CP-ABE scheme [11]. Their scheme requires a data service manager (such as the storage service provider) to generate attribute KEKs and distribute the keys to users. The attributes in the access policy

of a ciphertext are re-encrypted by their KEKs before the ciphertext is sent to a requestor.

When a user is revoked, the impacted attribute KEKs are updated and redistributed. This approach brings potential management overheads and scalability issues. The attribute KEKs are generated and maintained via a global binary tree that assigns users to the leaf nodes. For a large group of users, maintaining the binary tree becomes much harder when the system needs to add or delete users. The data service manager also has to know every user's attribute set in order to generate and distribute their attribute KEKs, which can leak too much information to a semi-trusted data service manager. In addition every user needs to have two sets of keys: secret attribute key shares and attribute KEKs.

Shuaishuai Zhu, Xiaoyuan Yang and XuGuang Wu [20] work on secure and practical attribute based encryption scheme without pairings (CP-ABE-WP) under cloud computing scenarios. It presents a new practical Attribute based Encryption Scheme without Pairing (CP-ABE-WP). Based on this, a secure file sharing system (SFSS) with attribute computing support is design.

5. PROPOSED METHOD

Access structure: Let G be a multiplicative cyclic group of prime order p. Let g be a generator of G and e be a bilinear map. Let H: {0, 1}*→ G be a hash function. Let K be the threshold of the access tree to control the amount of the shared group. Zp be the Lagrange coefficient.
Setup: The setup algorithm takes no input other than the implicit security parameter. It outputs the public parameters PK and a master key MK.

Randomly choose two numbers, a1,a2 ε Zp, and compute

PK = (G, g, h=g a2, t=g a1)

MK = (a1, a2)

Generate session key (Ks).

Encryption (PK, A, M):

A encryption algorithm run by a sender. The encryption algorithm takes as input the public parameters PK, a message M, and an access structure A over the universe of attributes. The algorithm will encrypt M and produce a cipher-text CT such that only a user that possesses a set of attributes that satisfies the access structure will be able to decrypt the message. Assume that the cipher-text implicitly contains A. Outputs the cipher-text. Mathematically express as:

Firstly, starting from the root node, choose a polynomial qx with order of dx for each node x on the tree, and let dx= kx−1 to generate node specific key (Ki).

Secondly, From the root node, randomly choose a number s ∈ Zp and let qR (0)= s. The value of qR on the other dR is randomly picked.

Thirdly, from the size of file generate unique IDi.

Finally, let Y be the leaf node set of the access tree, and ET be the ciphertext embedded into the access tree T. Then ET can be computed by:

ET= (T, C'=m.ts, C=hs, IDi, Ki)

Key Generation (MK, S):

The key generation algorithm takes as input the master key MK and a set of attributes s that describe the key and the public parameters PK1. It outputs a private key SK.

The secret key can be computed by:

SK= (D'=g (a2+PK1).s)

Decryption (PK, CT, SK):

The decryption algorithm takes as input the public parameters PK, a ciphertext CT, which contains an access policy A, Generate session key(Ks), specific key (Ki), , choose unique IDi and a private key SK, which is a private key for a set S of attributes. If the set S of attributes satisfies the access structure A then the algorithm will decrypt the ciphertext and return a message M.

6. RESULT AND ANALYSIS

The projected tool is a Net beans and Cloudsim implemented product that compares the result of proposed algorithm against the existing algorithms. In order to evaluate the performance of cloud computing algorithms for attribute based encryption this paper focus on the following parameters:
- Simplicity
- Efficiency
- Robustness
- Availability
- Integration

This experiment mainly can compare the results between the two algorithms i.e. the proposed algorithm Session Based CP-ABE-WP with pairing and the existing CP-ABE-WP without pairing.
For the comparison of result the result analyst can select any one of the algorithm first and apply the algorithm to the Encryption as well as decryption, followed by it the second algorithm can be applied simultaneously. The result can be compared in context of time in milliseconds.

6.1 Result Comparison (In Tabular Form)

Table 1. Table Result Analysis for Encryption (Time Comparison in milliseconds)

S.no	Existing CP-ABE-WP	Proposed method
1.	25159476	19618528
2.	27560734	19075968
3.	22605344	13624083
4.	21450269	20052236
5.	23931376	17786106
6.	28602520	14980485
7.	23903054	19132612

8.	**11099980**	**18306828**
9.	**19762529**	**25650169**
10.	**25494908**	**28029930**
Total time (Encryption)	**229570190**	**196256945**

The table above displays some time comparison of two algorithms one is existing algorithm and second is proposed algorithm for encryption of data.

Table 2. Table Result Analysis for Encryption (Time Comparison in milliseconds)

S.no	Existing CP-ABE-WP	Proposed method
1.	**440873**	**266503**
2.	**337821**	**262750**
3.	**342257**	**266503**
4.	**331337**	**263432**
5.	**332020**	**264114**
6.	**1517806**	**261726**
7.	**328607**	**262068**
8.	**386276**	**266844**
9.	**332362**	**261043**
10.	**1686716**	**267527**
Total time (Decryption)	**6036075**	**2642510**

The table above displays some time comparison of two algorithms one is base paper algorithm and second is proposed algorithm for decryption of data.

6.2 Result Comparison (In Graph)

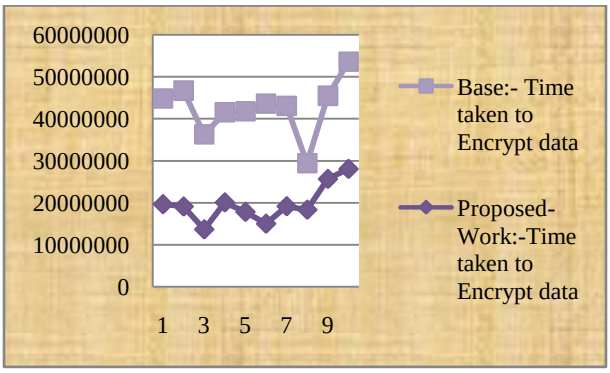

Fig 2: Comparison Graph (Decryption)

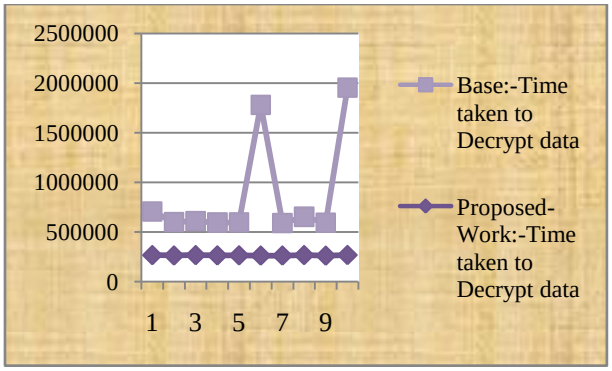

Fig 3: Comparison Graph (Decryption)

7. CONCLUSION

The proposed algorithm is a Session based CP-ABE-WP with pairing use in cloud computing an improved version of existing CP-ABE-WP without pairing. The result analysis it is quite clear that proposed system is better as compared to the existing systems. The proposed method has maintained a hierarchy of users to ensure that files are used by their appropriate users. Proposed system maintains the hierarchy of management with generation of secrete key and file attribute on the time of encryption. These generated components are required for decryption but in existing algorithm generation of these components is not possible. On the basis of these we can prove that proposed system give more efficient output with security in Hadoop system. This concept is very important in cloud computing that existing algorithm does not support.

8. REFERENCES

[1] W. Zeng, Y. Zhao, K. Ou, and W. Song, "Research on cloud storage architecture and key technologies," in2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, 2009, pp. 1044–1048.

[2] X. Jing and Z. Jian-jun, "A brief survey on the security model of cloud computing," in Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science, 2010, pp. 475 – 478.

[3] R. Chandramouli and P. Mell, "State of security readiness," in Crossroads. ACM, 2010, pp. 23 – 25.

[4] Hadoop. http://hadoop.apache.org/, 2007.

[5] S. Ghemawat, H. Gobioff, and S. Leung. The Google file system. In ACM Symposium on Operating Systems Principles, 2003.

[6] M. C. Schatz. CloudBurst: highly sensitive read mapping with MapReduce. Bioinformatics, 25(11):1363– 1369, 2009.

[7] D. Borthakur. The hadoop distributed file system: Architecture and design. http://hadoop.apache.org/, 2007.

[8] A. Sahai and B. Waters, "Fuzzy identity based encryption," in Advances in Cryptology, vol. 3494 of LNCS. Springer, 2005, pp. 457 – 473.

[9] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in 13th ACM conference on

Computer and communications security. ACM, 2006, pp. 89 – 98.

[10] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in 14th ACM conference on computer and communications security. ACM, 2007, pp. 195 – 203.

[11] J. Bethen court, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in 2007 IEEE Symposium on Security and Privacy. IEEE Computer Society, 2007, pp. 321–334.

[12] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute based encryption," in ICALP, 2008, pp. 579 – 591.

[13] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in IACR Cryptology ePrint Archive, no. 290, 2008.

[14] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters, "Secure attribute-based systems," in 13th ACM conference on computer and communications security, 2006, pp. 99 – 112.

[15] P. Junod and A. Karlov, "An efficient public-key attribute-based broadcast encryption scheme allowing arbitrary access policy in Tenth annual ACM workshop on digital rights management. ACM, 2010, pp. 13–24.

[16] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in ACM Symposium on Information, Computer and Communications Security, 2010, pp. 261 – 270.

[17] M. Blaze, G. Bleumer, and M. Strauss, "Divertable protocols and atomic proxy cryptography," in EUROCRYPT, vol. 1403 of LNCS. Springer, 1998, pp. 127–144.

[18] S. Jahid, P. Mittal, and N. Borisov, "Easier: Encryption-based access control in social networks with efficient revocation," in 6th ACM Symposium on Information Computer and Communications Security, 2011, pp. 411–415.

[19] J. Hur and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," in IEEE Transactions on Parallel and Distributed Systems, vol. 22, 2011, pp. 1214 – 1221.

[20] Shuaishuai Zhu, Xiaoyuan Yang and XuGuang Wu, "Secure Cloud File System with Attribute based Encryptio," 2013 5th International Conference on Intelligent Networking and Collaborative System 2013 IEEE.