ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 2, Issue 9, September 2017

Effective Use of ABKE Algorithm for Secure Cloud Storage Auditing

[1] R.Raj Priya1, [2] Dr.A.Muthukumaravel, [3] K.Bharathi
[1] Research Scholar, Department of Computer Science, Bharath University, Chennai.
[2] Head, Department of MCA, Bharath University, Chennai.
[3] Asst. Professor, Dept. of Computer Science, Pachaiyappa's College for men, Kanchipuram.

Abstract: ***The use of cloud computing is increased in the field of Information Technology. The data growth is tremendous in nature and so the cloud users outsource their data to the centralized system. Hence, data integrity plays a challenging task in the outsourced data. Cloud infrastructure is more powerful and reliable compared to other personal computing devices but cloud storage has both internal and external threats for data integrity. In previous works, static process of data verification model is achieved but still it lacks to eliminate the internal and external threats. In this paper, we aim to study about the dynamic support of the data verification model without compromising the fairness of the system. We design "Attribute Based Key Exchange (ABKE)", that intends to exchange the data from the client to the server in a secured way. By the issue of certificates from Certificate Authority (CA), the client can communicate and outsource the data to the server safely. This solution enables the system to work better in terms of fairness, security and privacy. Experimental designs will prove the effectiveness of the system.***

Keywords: *Public auditing, ABKE algorithm,Certificate authority,TPAr*

I. INTRODUCTION

Cloud computing is one of the fast improving technology in today's world. The practice of storing, managing and processing data that is available on networks of remote servers hosted on internet and not on own personal computer or local server. In simple words cloud computing means internet based services provided on demand where all the data and resources are stored on servers called cloud. Cloud services can be divided into three types. They are private, public and hybrid. Private services are services provided by business data centres whereas public services are provided by third party provider over internet. Hybrid services are a combination of public and private.We are proposing this paper for the public type of service.People use cloud computing since it has many advantages such as it hides the implementation and platform details, resources and all other complexities like maintaining, managing etc. All these facilities at low cost is the most great advantage.Apart from these the user can use the cloud from anywhere and anytime that is portability has kept users at ease always and cloud infrastructure a great success.Even though there are many advantages of cloud computing there are also some disadvantages or requirements that a good cloud service provider should satisfy.The important requirements out of all are security,data integrity and privacy preserving.Because data stored on load is shared by many people there are more chances for threats like loss or corruption of data due to hardware, software or sometimes human mistakes. At those situation,the cloud service provider also called as CSP may try to hide the information lost to maintain its reputation or fame. So it's necessary for a data owner to check all its data frequently. This leads to the requirement of data integrity.Next comes the security issues which is mainly of two types.Security issues faced by CSP and security issues faced the data owner. Both data owner and CSP should guarantee that each other's data are protected and secure.The third requirement is privacy preserving which shouldbe taken care when security issue is dealt with. To address these problems, data auditing is very important.Regular auditing ensures that the system checks user data correctness without affecting or increasing burden to both CSP and data owner that is by reducing overhead on CSP and data owner.

In existing system they have differentiated tag index which indicated tag computation and block index which indicated block position, and rely on an index switcher to keep a mapping between both of them. After every update operation, it allocates a new tag index for the operating block and updates the mapping between the tag indices and the block indices. Such an indirection between block indices and tag indices enforced block authentication and avoids tag re-computation of blocks after the operation position simultaneously. In this paper they only concentrate on combining data dynamics support and fair dispute arbitration into a single auditing scheme. There is a Third Party Auditor(TPA) in this system which is viewed as a delegator of the data owner and is not necessarily trusted by the CSP. To address the fairness problem they introduce a third-party arbitrator(TPAr) into the model,which is a professional institute which solve conflicts between data owner and CSP and is paid by both the data owner and the CSP.But the main disadvantage of this paper is in a cloud scenario both

owners and CSP have the motive to cheat. The CSP makesprofit by selling its storage capacity to cloud users, so he has the motive to reclaim sold storage by deleting rarely or never accessed data, and even hides data loss accidents to maintain a reputation as mentioned above because the signatures are generated by CSP and verified by TPAr. Also it requires a strong method to overcome this problem where the existing method is found weaker.
To do so in this paper we propose a public verifier or a third party called a Certificate Authority (CA) which uses ABKE algorithm and produce keys and also issues a certificate which authenticates the data owner to CSP.We also use a Third Party Arbitrator as in existing system who helps in solving disputes when fairness problem comes between data owner and cloud server.Our proposed system helps in securely auditing cloud storage.

II. PROBLEM STATEMENTS

2.1 System Model:

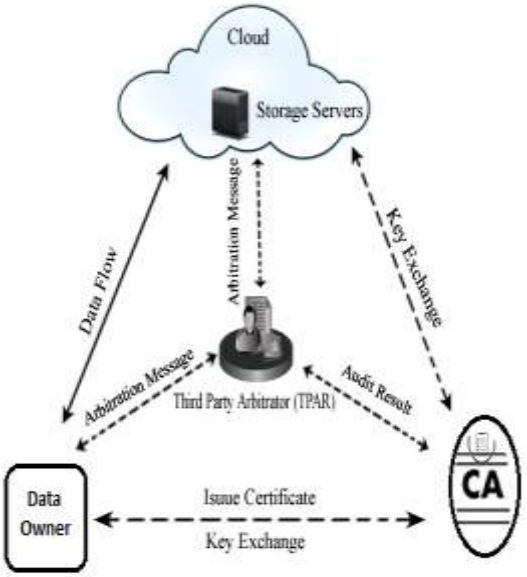

Figure1 Overall system model

The proposed system compromises of four modules and they are the data owner or client, Cloud Service Provider(CSP), Third Party Arbitrator(TPAr) and Certificate Authority (CA). The data owner is the client who stores, access or modifies data onto cloud server by uploading, downloading or by viewing files.To do so the data owner must first register and get the signature. Once signature is received it uses the signature for all encryption and decryption.The CA using ABKE algorithm issues certificate that is the signature used for encryption and decryption by the data owner once after it registers and a request is initiated .Also each time when the client initiates any request to CSP for example download or upload, then first the request is transferred to CA to check the signature and then from there only the request is transferred to CSP along with concerned shared key. CA provides keys shared between client and CSP so the message will be encrypted and decrypted for safe transmission. By key generation the CA records all necessary information for auditing and transfers the audited result to TPAr.The CSP is one who owns cloud server which has network of servers connected through internet.CSP provides all services on demand for which the client pay. CSP computes based on the policy so that the data owner can be authenticated and approves the request by providing the correct shared client key.The TPAr solves the fairness problem between the data owner and the cloud service provider. We always assume that the data owner is honest and the CSP is dishonest but there are many chances for the data owner to misbehave for making some money at times. Therefore the third party arbitrator is used to solve the disputes. Upon every transaction between client and cloud server, a metadata signature is computed and sent to TPAr individually by both. These signatures are stored along with the audited result sent by CA and used at times of dispute.

2.2 Threat Model:

i) Threats caused by CSP:

The CSP might delete few data for space,hide data loss caused by system failure, human errors, forward the data to other server, send polluted dataetc to maintain the reputation and for making profit.

ii) Threats caused by Data Owner:

We always assume that the data owner is honest and they won't misbehave but there are chances for a data owner to misbehave for making money or take revenge on CSP for previous disputes.

iii) Threats caused by others:

Other users who share same CSP, other users on internet might try to hack the shared keys for accessing private data's.

2.3 Design Objectives:

i) Dynamic data support: The auditing process should ensure or work on the dynamic data which reduces the online burden of the client.
ii)Security: The message being transferred should be safe guarded. Nor the CA or the TPAr should be able to see the data.
iii) Fairness guarantee: The disputes should be fairly cleared.

III. ATTRIBUTE BASED KEY ENCRYPTION (ABKE) ALGORITHM:

Attribute Based Encryption (ABE):
ABE is a concept designed upon public key cryptography. In basic public key cryptography the message is encrypted using the receiver's public key and sent to the receiver. Then in the Identity Based Encryption (IDE) which used one of the attribute of receiver for encryption, example the email address. Note in IDE only one identical attribute is used for encryption.Then came the advance concept that is ABE.It combines a set of attributes of receiver and encrypt messages. There are two types of ABE. Cipher Text Policy Attribute Based Encryption (CP ABE) and Key Policy Attribute Based Encryption (KP ABE).The term or rule is, someone can only be able to decrypt a cipher text if only if the person hold the matching attributes where user keys are always issued by trusted party like Certificate Authority in this paper. In this paper we use CP ABE type of ABE algorithm and below is the explanation.

In CP ABE a user's private key is related with a set of attributes and cipher text specifies an access policy over a defined universe of attributes within the system. A user will able to to decrypt a cipher text, if only if the attributes satisfy the policy of respective cipher text. Policies are definedusing conjunction, disjunction over attributes and(k,n)- Thresholdgates. That is k out n attributes have to be present.For example, let us assume that the universe of attribute is defined to be {P,Q,R,S} and user 1 receives a Key to attributes {P,Q} and user2 to attribute {S}. If cipher text is encrypted with respect to the policy (P $\wedge$ Q) v S then user 2 will be able to decrypt, while user 1 will not be able to decrypt.In the example user 1 should have both attributes P and R because the policy states P^R, but since it has P and Q, it cannot decrypt.User 1 can decrypt since it has attribute S as stated in policy that the user must have either attribute S or attribute P,R. Therefore the authorization is included into the encrypted data that is cipher text and only the user who satisfies the related policy will be able to decrypt the cipher text. So the policy is designed in such a way that only authenticated user will be able to decrypt.
Attribute Based Credential (ABC):
Attribute Based Credential algorithm is used by service provider to authenticate data owners/users y providing high security and privacy preserving. The data owner obtains a certificate issued y the CA derived upon attributes ad which is shown to service provider for authentication. ABC supports unlinkability, that is,"Multiple communications with same server by 1 or more client cannot be linked together".
Attribute Based Key Exchange (ABKE):

ABKE is an algorithm which is the combination of Attribute Based Encryption (ABE) and Attribute Based Credential (ABC) algorithm. In ABE scheme the user who decrypts will stay passive and therefore it is non-interactive. To avoid this we make use of the property unlinkability in ABC algorithm. The rule states that different users of the system cannot pass the policy combining all their attributes into single for verification that neither could have individually passed. In this paper we focus to achieve an interactive solution involving a user that holds a certificate issued by CA, claiming for its attributes and a server that consists of a policy that is computable on the certificate (that is set of attributes). The main aim for the server is to establish a shared key with the user, if and only if user's certificate satisfies the policy.

We present functionality F_{abke} for attribute-basedkey exchange supporting attribute privacy, unlinkability, and collusion resistance. The functionality is initialized with a set of attribute vectors $\{X_i\}$, where X_i corresponds to the attribute vector of client P_i. The functionality begins by waiting for a message from a server S_j that contains a circuit C representing S_j's policy. The functionality stores this information and broadcasts a notification to all parties $P_1,..,P_n$ that a policy is available. Upon receiving a response by one of the parties, say, P_i, the functionality proceeds as follows. If $C(X_i)=1$, the policy is satisfied and so the functionality forwards a key k to both P_i and S_j. If $C(X_i)=0$, then F_{abke} sends $\perp$ to both P_i and S_j.

IV. RELATED WORKS

4.1 Remote integrity checking

The authors of this paper are Y.Deswarte, J.J.Quisquater and A.Saidane. In this paper the problem dealt is checking the integrity of files stored on remote servers. Since servers are easily prone to malicious attacks by hackers and other attacks, the result of simple integrity checks which runs on the remote servers cannot be trusted. Also, downloading the files from the remote server to the verifying host is generally impractical. Finally two solutions are proposed based on challenge-response protocols in this paper.

4.2 Demonstrating data possession and uncheatable data transfer

The authors of this paper are D.L.Gazzoni Filho and P.S.L.M.Barreto.We observe that a certain RSA-based secure hash function are homomorphic encryption which means is a form of encryption that allows computations to be carried out on cipher text, thus generating an encrypted result which when decrypted matches the result of operations performed on plain text. In this paper a protocol based on hash function which prevents 'cheating' in a data transfer transaction is used, while placing a little burden on the trusted third party that oversees the protocol. This paper also explains a cryptographic protocol based on similar principles, through which a prover can demonstrate possession of an arbitrary set of data known to the verifier. The verifier isn't required to have this data at hand during the protocol execution, but rather only a small hash of it. The protocol is also provably as secure as integer factoring

4.3(POR):Proofs of retrievability for large files

The authors of this paper are A. Juels and B. S. Kaliski Jr. In this paper, we define and explore *proofs of retrievability* (PORs). A POR scheme enables an archive or back-up service i.e prover to produce a concise proof that a verifier can retrieve a target file *F*, that is, that the archive retains and reliably transmits files data sufficient for the user to recover *F* in its entirety.A POR may be viewed as a type of cryptographic proof of knowledge (POK), but specially designed to handle a *large* file *F*. In this paper POR protocols are explored, here in which the communication costs, number of memory accesses for the prover, and storage requirements of the verifier are small parameters essentially independent of the length of *F*. In addition to proposing new, practical POR constructions, we explore implementation considerations and optimizations that bear on previously explored and related schemes. A POR provide guaranteed quality-of-service that is shown that a file is retrievable within a certain time bound.

4.4 Provable data possession at untrusted stores

This paper's authors are G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song. In this paper a model for *provable data possession* (PDP) that allows a client who has stored data at a server which is not completely trusted to verify that the server possesses the original data without retrieving it. The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces I/O costs. The client maintains a constant amount of metadata to verify the proof. The challenge/response protocol transmits a small, constant amount of data, which minimizes network communication. So, this PDP model for remote data checking supports large data sets in widely-distributed storage system.

4.5 Dynamic and Publicauditing with fair arbitration

This is our existing system and the authors of this paper are Hao Jin, Hong Jiang and Ke Zhou. This paper proposes a public auditing scheme with data dynamics support and fairness arbitration of potential disputes. In this model there is an index switcher which is used to eliminate the limitation of index usage in tag computation in current schemes and achieve efficient handling of data dynamics. Also to solve the fairness problem and that no party can misbehave there is signature exchange idea to design fair arbitration protocols. So that any possible dispute can be fairly settled.

V. CONCLUSION

The aim of this paper is to prvide secure auditing and to guarantee data integrity at best. And so our system is designed using"Attribute based Key Exchange" algorithm that heps to exchange information between data owner and cloud server in a secured way.This is the better solution to solve cloud security and auditing problems in terms of fairness, linkability and privacy. The proposed scheme is efficient when compared to previous systems.

VI. REFERENCES

[1] Y. Deswarte, J.-J. Quisquater, and A. Sa¨ıdane, "Remote integrity checking," in Proc. 5th Working Conf. Integrity and Intl Control in Information Systems, 2004, pp. 1–11.
[2] D. L. Gazzoni Filho and P. S. L. M. Barreto, "Demonstrating data possession and uncheatable data transfer." IACR Cryptology ePrint Archive, Report 2006/150, 2006.
[3] A. Juels and B. S. Kaliski Jr, "Pors: Proofs of retrievability for large files," in Proc. 14th ACM Conf. Computer and Comm. Security (CCS07), 2007, pp. 584–597.
[4] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proc. 14th ACM Conf. Computer and Comm. Security (CCS07), 2007, pp. 598–609.
[5] Hao Jin, Hong Jiang, and Ke Zhou "Dynamic and Public Auditing with Fair Arbitration for Cloud Data". IEEE Transactions on cloud computing, Vol. 13, NO. 9, September 2014
[6] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Proc. 14th European Conf. Research in Computer Security (ESORICS 08), 2009, pp. 355–370.
[7] M. A. Shah, R. Swaminathan, and M. Baker, "Privacy-preserving audit and extraction of digital contents." IACR Cryptology ePrint Archive, Report 2008/186, 2008.
[8] C. Wang, K. Ren, W. Lou, and J. Li, "Toward publicly auditable secure cloud data storage services," Network, IEEE, vol. 24, no. 4, pp. 19–24, 2010.
[9] C. Erway, A. Kupc¸ ¨ u, C. Papamanthou, and R. Tamassia, "Dynamic ¨ provable data possession," in Proc. 16th ACM Conf. Computer and Comm. Security (CCS 09), 2009, pp. 213–222.
[10] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, "Dynamic audit services for integrity verification of outsourced storages in clouds," in Proc. ACM Symp. Applied Computing (SAC 11), 2011, pp. 1550–1557.
[11] Q. Zheng and S. Xu, "Fair and dynamic proofs of retrievability," in Proc. 1st ACM Conf. Data and Application Security and Privacy (CODASPY 11), 2011, pp. 237–248.
[12] N. Asokan, V. Shoup, and M. Waidner, "Optimistic fair exchange of digital signatures," in Proc. 17th Intl Conf. Theory and Applications of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT98), 1998, pp. 591–606.
[13] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in Proc. IEEE INFOCOM, 2010, pp. 1–9.
[14] C. Wang, S. S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacypreserving public auditing for secure cloud storage," IEEE Trans. Computers, vol. 62, no. 2, pp. 362–375, 2013. [15] B. Wang, B. Li, and H. Li, "Oruta: Privacy-preserving public auditing for shared data in the cloud," IEEE Trans. Cloud Computing, vol. 2, no. 1, pp. 43–56, 2014.
[16] Vladimir Kolesnikov, Hugo Kraweczk,Yehuda Lindell and Tal Rabin, "Attribute-based Key Exchange with General Policies", URL:https://eprint.iacr.org/2016/518 Presented at ACM CCS 2016.