International Journal of Engineering Research-Online
A Peer Reviewed International Journal
Articles available online http://www.ijoer.in

Vol.3., Issue.2, 2015

RESEARCH ARTICLE

ISSN: 2321-7758

EFFICIENT AUTHENTICATION USING MERKLE HASH ALGORITHM IN JELASTIC SERVER

M.KANI[1], M.SAKTHIVEL[2]
[1]Master of computer Application, Vel tech High Tech Engg College, Chennai
[2]Assistant Professor, Vel tech High Tech Engg College, Chennai

Article Received: 30/03/2015 | Article Revised on:10/04/2015 | Article Accepted on:14/04/2015

ABSTRACT

A new approach of Fingerprint authentication using Merkle Hash Tree is proposed for jelastic server. A decentralized access control is proposed to support anonymous authentication for data stored in cloud environment. In this system authenticity of each and every individual is verified by cloud server before storing the data in cloud. The authenticity is ensured by server without the knowledge of user identity. The proposed system also has the added feature of access control which only valid users to access and decrypt the stored information in data centers. Decentralized approach prevents replay attacks and enable user to create, modify and read the data in the cloud data center. Revocation is also introduced to abort the attacks in centralized data center. Decentralized access control is robust in distributed data's, which is highly secured and authorized compared to existing centralized schemas. The communication, storage, manipulation, retrieval, security are highly defined in decentralized access control schema.The user has to submit the adjacent and sibling shares of fingerprint template for authentication purpose. The signature is generated in the cloud service provider and thus verified with the stored signature in the cloud . The misuse of sensitive data can be avoided and this provides an effective and efficient user remote authentication with the cloud.

Keywords: biometric modality, Finger print, Merkle Hash Tree, jelastic server, Hasing,

I INTRODUCTION

Cloud computing have lots of attention for both academic and industrial research world. In cloud network internet is used to store both in source and out source. In the open network everyone have rights to share the common network and there is no secure in the sharing the information. 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. 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. The validity of the user who stores the data is also verified. 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. Apart from the technical solutions to ensure security and privacy, there is also a need for law enforcement industrial worlds. 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. Apart from the technical solutions to ensure security and privacy, there is also a need for law enforcement. Recently, Wang et al. [2] addressed secure and dependable cloud storage. Cloud servers prone to Byzantine failure, where a storage server can fail in arbitrary ways [2]. The cloud is also prone to data modification and server colluding attacks. In server colluding attack, the adversary can compromise storage servers, so that it can modify data files as long as they are internally consistent. To provide secure data storage, the data needs to be encrypted. However, the data is often modified and this dynamic property needs to be taken into account while designing efficient secure storage techniques.

II PROBLEM DESCRIPTION

Existing work on access control in cloud are centralized in nature. Some scheme uses a symmetric key approach and does not support authentication. Some do not support authentication as well. Earlier work by Zhao et al. provides privacy preserving authenticated access control in cloud. However, the authors take a centralized approach where a single key distribution center (KDC) distributes secret keys and attributes to all users. Unfortunately, a single KDC is not only a single point of failure but difficult to maintain because of the large number of users that are supported in a cloud environment. However, the authors take a centralized approach where a single key distribution center (KDC) distributes secret keys and attributes to all users. Unfortunately, a single KDC is not only a single point of failure but difficult to maintain because of the large number of users that are supported in a cloud environment.

III OUR CONTIBUTION

We, therefore, emphasize that clouds should take a decentralized approach while distributing secret keys and attributes to users. It is also quite natural for clouds to have many KDCs in different locations in the world. We A new approach of Fingerprint authentication using Merkle Hash Tree is proposed for jelastic server. In the proposed scheme, the cloud verifies the authenticity of the ser without knowing the user's identity before storing data. Our scheme 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. Moreover, our authentication and access control scheme is decentralized and robust, unlike other access control schemes designed for clouds which are centralized. The Fingerprint template is split it into eight shares using image processing technique in the client side. The splitted eight shares are given as inputs to merkle hash tree where in each share has to undergo hashing function and hence root signature is generated. The signature is generated and stored in the Jelastic cloud server.(Jelastic cloud is a public cloud which is used to access the file stored in the jelastic cloud database with valid mailid and password).The user has to submit the adjacent and sibling shares of fingerprint template for authentication purpose. The signature is generated in the cloud service provider and thus verified with the stored signature in the cloud. The misuse of sensitive data can be avoided and this provides an effective and efficient user remote authentication with the cloud.

KEY FEATURES INCLUDING

a).User Enrollment: Users have an initial level Registration Process at the web end. The users provide their own personal information for this process. The server in turn stores the information in its database.

b).Trustee and Key Distribution Center: Users receive a token 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 token. There are multiple KDCs (here 1), which can be scattered. Users on presenting the token to KDC receive keys for encryption/decryption and signing. SK are secret keys given for decryption, Kx are keys for signing.

c).File Access Control:After the key was received by the User, the message MSG is encrypted under the access policies. The access policies decide who can access the data stored in the cloud. The cipher text 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.

IV SYSTEM OVERVIEW

To overcome the problem of existing system, we proposed a fingerprint authentication scheme using the concept of Merkle Hash Tree. The data owner stores the file in an encrypted form in the cloud server. The cloud user has to register with the owner along with the root signature. The Fingerprint template is split it into eight shares using image processing technique in the client side. The splitted eight shares are given as inputs to merkle hash tree where in each share has to undergo hashing function and hence root signature is generated. The signature is generated and stored in the Jelastic cloud server.(Jelastic cloud is a public cloud which is used to access the file stored in the jelastic cloud database with valid mailid and password).The user has to submit the adjacent and sibling shares of fingerprint template for authentication purpose. The signature is generated in the cloud service provider and thus verified with the stored signature in the cloud. The misuse of sensitive data can be avoided and this provides an effective and efficient user remote authentication with the cloud.

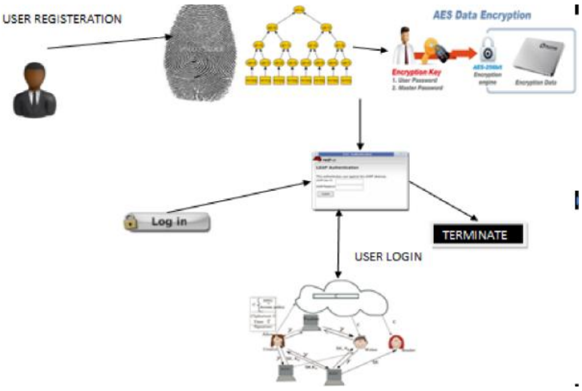


Fig. 1: System Architecture of cloud data

1.DESIGN AND IMPLEMENTATION

Implementation involves following constraints.

Constraints as Informal Text

Operational Restrictions

Constraints Integrated in Existing Model

Constraints as a Separate Concept

Constraints Implied by the Model

Design Involves

Determination of the Involved Classes

Determination of the Involved Objects

Determination of the Involved Actions

Determination of the Require Clauses

Global actions and Constraint Realization

2.**CONSTRAINTS IN IMPLEMENTATION**

A hierarchical structuring of relations may result in more classes and a more complicated structure to implement. Therefore it is advisable to transform the hierarchical relation structure to a simpler structure such as a classical flat one. It is rather straightforward to transform the developed hierarchical model into a bipartite, flat model, consisting of classes on the one hand and flat relations on the other. Flat relations are preferred at the design level for reasons of simplicity and implementation ease. There is no identity or functionality associated with a flat relation. A flat relation corresponds with the relation concept of entity-relationship modeling and many object oriented methods.

V.SECURITY ANALYSIS

a).Data Privacy: In this work, we consider only search privacy; because of privacy of the documents can be ensured by the encryption algorithm. That is, we focus on the confidentiality of the search request and the index T. Using the trapdoor technology, the attacker directly to get the plaintext is impossible from the cipher text. So we mostly concern the confidentiality of the index T.

b).Verifiable Search ability:

we assume k steps are performed by the server. If "No" returned, we would know that the first k 1 characters are matched while Tw [k] is mismatched, which could be described by a k bit binary sequence b = (1;:::;1;0); if "Yes" is returned, b = (1;:::;1;1). In our scheme, firstly, we check the number of proof whether is equal to the sending trapdoors. If not, we can say the server is not making a full search. If pass, we exploit a random sampling method to check. For each of Proof to be tested, As [8], Similarly, starting from the last (or k-th) step, if "Yes", verify checks the integrity of the Concatenation of the document identifiers by Computing a keyed hash of it and comparing with the received one. In fact, the completeness of the search outcome is examine adhere. If the server returns a fraction of the search outcome, the user can find the server is not honest. Then we test that whether the trapdoor equals the received symbol string of the proof. After that, j decreased by one. If "No", the above step is skipped. Next, verify validates the correctness of the search outcome by decrypting the first part of Tj qj[r1] = Enc(Tj;qj[r0]; parent (Tj;qj)[r1]) to get (x; y) and testing whether: (1) Tw[j] equals x (2) Tw[j 1] equals y. To cheat the search results, the server need to forge the proof. There are two possible case:

(1) the server honestly search for a fraction of trapdoors and forge the proof or other trapdoors, at worst, the server do not any search;

(2) the server forge the proof according to the received trapdoor. For the case (1), the server must can generate a valid r1 with a different memˆ =6 mem, consider the adversary do not know the sk, it can be seen a random oracle.

In case (2), the adversary may use the r1 of another node, note that each node has a global unique r1, which will be rejected by verify. In addition, the argument above can be applied recursively to the (j1)the step in verify and so on.

V1.PERFORMANCE ANALYSIS

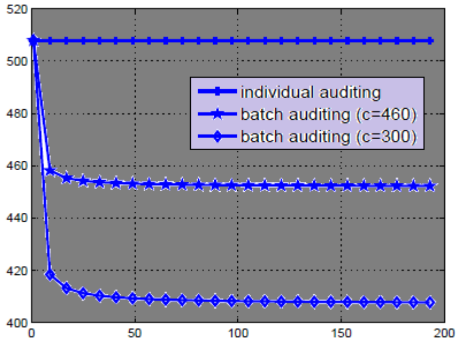
We conducted a thorough experimental evaluation of the proposed techniques on real data set: the recent ten years 'IEEEINFOCOM publications. The data set includes about 2,600 publications. We extract the words in the paper titles to construct the core keyword set in our experiment. The total number of keywords is3, 262 and their average word length is7.44. Our experiment is conducted on a Linux machine with an Intel Core 2processor running at 1.86GHz and 2G DDR2- 800 memory. The performance of our scheme is evaluated regarding the time cost of fuzzy set construction, the time and storage cost of index construction, the search time of the listing approach and the symbol-based tire-traverse approach.

VII.ALGORITHM AND EXPERIMENTAL RESULTS

To data's in the cloud center is maintained in secured manner by the use of crypt algorithms. The information is shared between the authorized persons by the generation of one time secure key. The files are viewed by authorized person in the cloud data center. Refreshed key is generated by the decentralized access. key distribution center act as the global manager for sharing the data in the cloud data center .once the key is received encryption is done by the crypt mechanism and the original data is obtained by the sever side. The median ghost has lower capability to guess the original data by the implementation of decentralized data.

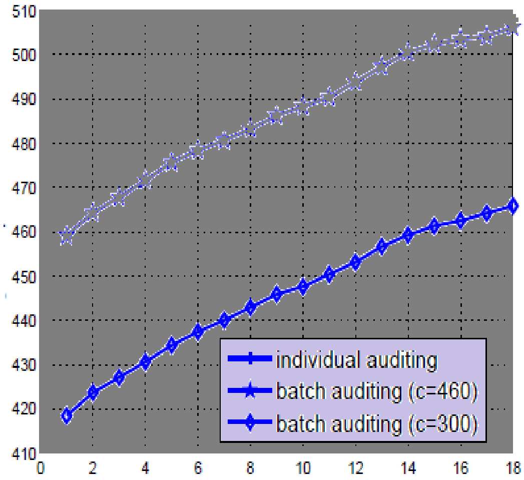
```
Merkle Hash algorithm

1. check file tag t, verify its
   authenticity, and end if fail;
2. Generate a secert key Sk={(i,_i)}
   challenge request
3. Compute μ' = Sk
4. Randomly pick r ← Zp, and compute
   R = e(u, v)r and  = h(R);
5. Compute μ = r + μ' mod p ;
6. Compute  = h(R), and then
   verify {μ, _,R} via
7. Generate the server side key
8.block the crypt message
9. generate the original message on
   both side
```

The algorithm describes the sharing of secured key in the common data center.



File shared in data center

The result shows the common sharing of files without aunthencity in the cloud network. The intermediation involvs key in the unsecured area. The authentication is effectively shared and secured in the common data center in the decentralized modem. The access of data is obtained in the following data sets.



Files shared after key generation

The result set describes the sharing of key in the secured data center.

VIII.CONCLUSION

In this project A new approach of remote user fingerprint authentication scheme using the concept of Merkle Hash Tree has been proposed. The data owner stores the file in an encrypted form in the cloud server. The cloud user has to register with the owner along with the root signature. In the client side, the Fingerprint template is split it into eight shares using image processing technique. The splitted eight shares are given as inputs to merkle hash tree wherein each share has to undergo hashing function and hence root signature is generated. The signature is generated and stored in the cloud server. The user has to submit the adjacent and sibling shares of fingerprint template for authentication purpose. The signature is generated in the cloud service provider and thus verified with the stored signature in the cloud. The misuse of sensitive data can be avoided and this provides an effective and efficient user remote authentication with the cloud.

Future Work:

Here we use the finger print image for the authentication process. In future we will use iris or face recognition.

REFERENCES

[1] International Conference on Dependable, Autonomic and Secure Computing (DASC 2009), Chengdu, China, December 12–14, 2009.

[2] M. Jakobsson, K. Sako, R. Impagliazzo, Designated verifier proofs and their applications, in: International Conference on the Theory and Application ofCryptographic Techniques (EUROCRYPT 1996), Zaragoza, Spain, May 12–16, 1996.

[3] A. Juels, B. Kaliski Jr., PORs: proofs of retrievability for large files, in: Proceedings of the 14th ACM Conference on Computer and CommunicationsSecurity (CCS'07), Alexandria, Virginia, USA, October 28–31, 2007.

[4] B. Kang, C. Boyd, E. Dawson, A novel identity-based strong designated verifier signature scheme, Journal of Systems and Software 82 (2) (2009) 270–273.

[5] G. Karame, M. Strasser, S. Capkun, Secure remote execution of sequential computations, in: 11th International Conference on Information and Communications Security (ICICS'09), Beijing, China, December 14–17, 2009.

[6] A. Marinos, G. Briscoe, Community cloud computing, in: Proceedings of Cloud Computing: First International Conference (CloudCom 2009), Beijing,China, December 1–4, 2009.

[7] R. Merkle, Protocols for public key cryptosystems, in: IEEE Symposium on

Security and Privacy, Oakland, California, USA, April, 1980.

[8] F. Monrose, P. Wyckoff, A. Rubin, Distributed execution with remote audit, in: Proceedings of the Network and Distributed Systems Security Symposium (NDSS), San Diego, California, USA, 1999.

[9] P. Patel, A. Ranabahu, A. Sheth, Service level agreement in cloud computing, in: Cloud Workshops at OOPSLA09, Orlando, Florida, USA, October 25–29,2009.

[10] S. Pearson, Y. Shen, M. Mowbray, A privacy manager for cloud computing, in: First International Conference (CloudCom 2009), Beijing, China,December 1–4, 2009.

[11] A. Sadeghi, T. Schneider, M. Winandy, Token-based cloud computing: Secure outsourcing of data and arbitrary computations with lower latency, in:Trust and Trustworthy Computing, Berlin, Germany, June 21–23, 2010.

[12] H. Takabi, J. Joshi, G. Ahn, Security and privacy challenges in cloud computing environments, IEEE Security & Privacy 8 (6) (2010) 24–31.

[13] C. Wang, K. Ren, J. Wang, Secure and practical outsourcing of linear programming in cloud computing, in: 30th IEEE Conference on Computer Communications (INFOCOM 2011), Shanghai, China, April 11–15, 2011.

[14] C. Wang, Q. Wang, K. Ren, W. Lou, Privacy-preserving public auditing for data storage security in cloud computing, in: 29th IEEE Conference on Computer Communications (INFOCOM'10), San Diego, California, USA, March 14–19, 2010