ISSN:0975-9646

Vamsi Krishna Kondragunta et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014, 2256-2258

Retrieving Data through Two Way Security Methods In Dynamic Cloud Storage Systems

Vamsi Krishna Kondragunta , Md. Mohsin Shariff , V.Roopesh , Dr.T.V Surya Narayana

Dept. Of ECM, KLEF University
Vaddeswaram, Guntur, India

Abstract— **Storage as a service is an growing field in cloud storage and it comes only at price. Storing Data in an cloud which is not trusted can reveal the contents of that particular file. Even the File owner cannot know whether it is an trusted cloud or not .To avoid these type of ambiguous situations we propose an system of cloud storage services where we include an new actor other than owner, user and cloud. It is the trusted third party where the file owner can fully reliable on this Trusted Third Party. The implementation of cloud storage system where we can provide more security to the present cloud storage systems will be implemented in this paper. With these storage systems the security for the data which is hard to implement can be implemented easily.**

Keywords— ***cloud security, Storage Security ,file key, Trusted third party, Data Retrieval***

INTRODUCTION

Local management of huge amount of date is problematic and costly due to requirements of high storage capacity and qualified personnel. Since in the present digital world many organizations produces a large of amount of sensitive data which includes personal information, electronic health records and financial data. There may be a chance that these cloud service provider's may not be an trusted one. In these cases we cannot provide security for the file which is stored in the cloud storage systems. So in this paper we propose the implementation of new actor in the cloud storage systems. The actor is the trusted third party.

The first challenging issue here is the implementation of mutual trust between the data owner and the csp. If there is any dishonest csp ,the trusted third party can know about that csp and it can revoke that particular csp.

And the second challenging issue here is the access control, it will allow the data owner to grant or revoke access rights to the user regarding the outsourced data.

EXISTING WORK:

In this work, we referred [1] .Here the authors propose a scheme that addresses some important issues related to outsourcing the storage of data, to provide dynamisms to the data. The core principles which should consider while implementation of data outsourcing is to provide dynamic scalability. With this, we can use this data for various applications. With this kind of data ,the unauthorized user cannot access data and data can also be updated later by the user. After this kind of updating the user will receive the information of the updated data. For this implementation we need to know whether the data is rancid.

SYSTEM DESIGN AND DESIGN GOALS:

For the implementation of the proposed system , there are four actors . They are Owner ,Trusted Third Party , Cloud Service Provider and User.

The functions of the owner are namely to upload the file ,set the number of file block divisions to be made , Update the file ,Send Key to the user.

While the functions of the Trusted Third Party is to verify the file ,verify the Cloud Service Provider and to upload the file to the Cloud Service Provider.

Then comes the Cloud Service Provider. The function's of the CSP includes Upload the file into the cloud which is verified by the TTP .

The last actor is the User. The user can see the files which are updated in the cloud and he can request the owner for the file key and if the owner sends the file key, the user can download that particular file using that key.

DESIGN GOALS:

Here in this section we characterize certain design goals that are taken as pre requisites for designing the proposed work. These include aspects relating to access, privacy, storage and efficiency.

ACCESS:

A owner must register himself before his first login. Then only he can upload an file into the cloud. The user should also Registers themselves. Otherwise he cannot see the files which are uploaded into the cloud. Unauthorized users must not get access to the contents in the cloud. The details at login are verified with the one stored in server before granting access.

PRIVACY:

The user's cannot access the files which are uploaded into the cloud. They can only see the details of the file. To download the file the user should have the key which can be obtained only with the owner's permission. Thus their privacy is preserved.

STORAGE:

The file owner can upload the file and he can also modify the contents of the file. The file user once if he receive the key he can download the file and access the contents of the file. The CSP and TTP can have the key but the CSP cannot modify the contents of the file.

EFFICIENCY:

The efficiency of the proposed scheme is defined as follows: Any file owner can store and share data files with others in the group by the cloud. The user can know the contents of only particular if he gets an particular key from the owner.

PROPOSED WORK :

When the owner and user registers and logins ,this information is stored in to the log file. The files uploaded by the user will be to the cloud by the CSP with the advancement to upload the file from Trusted third party. If the CSP tries to modify the content of the data then the Trusted third party will get the details regarding modifications' tried to made by the CSP.

When the above is implemented we can improve more security with already discussed changes. When the files are uploaded into the cloud ,then these files will be encrypted and uploaded into the cloud. When the user enters the key which is received from the owner, then he can decrypt the file and download the file.

Broadcast Encryption:

Broadcast encryption algorithm[bENC] enables an broadcaster to encrypt a message for an random subset of a user's of the group. The message can be decrypted only by the user's who are present in that subset only. However, other user's outside the subset cannot decrypt the message even if they conspire to get data. Here in our proposed scheme we used this Broadcast encryption algorithm. The Broadcast encryption algorithm has mainly 3 functional algorithms. They are : SETUP, ENCRYPT, and DECRYPT.

SETUP: This algorithm takes as input the number of system users n. It defines a bilinear group G of prime order p with a generator g, a cyclic multiplicative group GT , and a bilinear map ˆe : G × G → GT . The algorithm picks a random α ∈ Zp, computes gi = g(αi) ∈ G for i = 1, 2, . . . , n,n + 2, . . . , 2n, and sets v = gγ ∈ G for γ ∈R Zp. The outputs are a public key PK =(g, g1, . . . , gn, gn+2, . . . , g2n, v) ∈ G2n+1, and n private keys {di}1≤i≤n, where di = gγi∈ G.

ENCRYPT: This algorithm takes as input a subset S ⊆ {1, 2, . . . , n}, and a public key PK. It outputs a pair (Hdr, K), where Hdr is called the header (broadcast ciphertext), and K is a message encryption key. Hdr= (C0, C1) ∈ G2, where for t ∈R Zp, C0 = gt andC1 = (v ・_j∈S gn+1−j)t. The key K = ˆe(gn+1, g)t is used to encrypt a message M (symmetric encryption) to be broadcast to the subset S.

DECRYPT: This algorithm takes as input a subset S ⊆ {1, 2, . . . , n}, a user-ID i ∈ {1, 2, . . . , n}, the private key di for user i, the header Hdr = (C0, C1), and the public key PK. If i ∈ S, the algorithm outputs the key K =ˆe(gi, C1)/ˆe(di ・_j∈Sj_=I gn+1−j+i, C0), which can be used to decrypt the encrypted version of M.

ENTITY DESCRIPTION:

Owner:

The owner has the functionalities like registering him/her self with the cloud, he can upload the file and he can update that file which he has uploaded into the cloud. When the owner receives the key request from the user then he can sent the key to the user via to the user's mail account. This mail address was collected from the database where the user's registration is done.

Trusted Third Party:

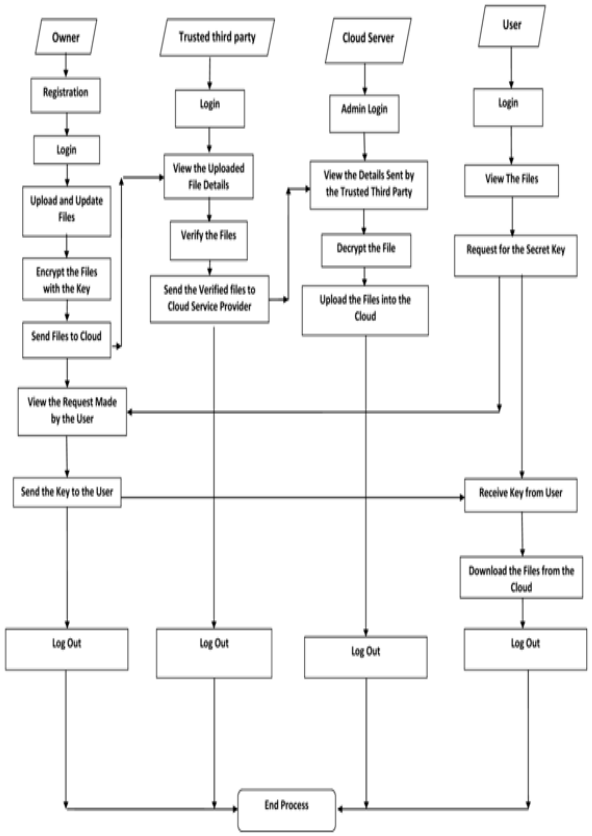
This is an important role to let the efficient functioning of the proposed system. When the owner upload's his file into the cloud, then TTP will receive the details of file upload and he verifies that file and he will upload it to the cloud. He also plays a major role in keeping track of the actions of any dishonest Cloud Service Provider (if any) exists.

Cloud Service Provider:

When the Trusted third party allows the owner's file too be uploaded into the cloud then The Cloud Service Provider will Upload the file into the cloud. The cloud Service Provider cannot make any changes to the file ,so we should select an cloud service provider to be an honest one .If any later found to be Dishonest then that cloud service provider can be revoked from the system by the Trusted Third Party.

User:

The user needs to be register him/herself into the cloud and he can then login into the system. When the owner upload an file to the cloud then the user can see the details of the file. If he is interested in the file then he can request the owner for the file key. The user will receive the key from the owner to the user's mail which he has given while registering in the system. The owner then, he can decrypt the file and download the file.



Dataflow Diagram Representing actions of entities

PERFORMANCE EVALUATION

Performance:

The cloud can be secured more with the Implementation of the Trusted Third Party. It can act as a trusted bridge between the owner and the Cloud Service Provider. The performance is increased with this implementation.

Security:

The file uploaded by the owner is encrypted with the bENC and the file is given a key which is also stored in the database. Without the key the user cannot access the file and data in that file.

Improvement:

Till now the cloud system works with owner, user and Cloud Service Provider. But whenever the file 's are stored in the cloud and those files are placed in the hands of the Cloud Service Provider. If sensitive data is stored in a dishonest cloud, then the security of the file may not be provided. The security system of cloud is improved with the use of the new actor the TTP which acts a trusted mediator among Owner, Cloud Service Provider and User.

CONCLUSION

With the growing data that need to be outsourced in to the cloud other than storing it in the remote servers have became a burden. To overcome this burden many CSP have been providing their cloud services. Among them there may be dishonest CSP. To avoid these situations With this paper's implementation that we proposed ,it will support the cloud storage system with supporting of dynamic data outsourcing , with this owner have ability of storing the data in the CSP and he can also update the Rancid Data. The User is capable of accessing the files only if he get Key from the user.

If any dishonest Cloud Service Provider is present in the system then the Trusted Third Party can revoke that CSP out of the system which is already implemented in our system.

ACKNOWLDGEMENT:

Hence, we declare that the above mentioned work is done by ourselves and it is not implemented in any of the current system.

REFERENCES

[1] Ayad Barsoum and Anwar Hasan "Enabling Dynamic Data and Indirect Mutual Trust for Cloud Computing Storage Systems" ieee transactions on parallel and distributed systems, vol. 24, no. 12, december 2013

[2] W. Wang, Z. Li, R. Owens, and B.Bhargava, "Secure and efficient access to outsourced data," in Proceedings of the 2009 ACM workshop on Cloud computing security, ser. CCSW '09. ACM, 2009, pp. 55–66.

[3] C. Erway, A. Kupu, C. Papamanthou, and R. Tamassia,"Dynami c provabl e dat a possessi on, " i n CCS'09: Proceedings of the 16th ACM Conference on Computer and Communications Security, New York, NY, USA, 2009, pp. 213–222.

[4] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik,"Scalable and efficient provable data possession," in SecureComm '08: Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks, New York, NY, USA, 2008, pp.1–10.

[5] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in CCS '07: Proceedings of the 14th ACM Conference on Computer and Communications Security, New York, NY, USA, 2007,pp. 598–609.

[6] M. J. Atallah, K. B. Frikken, and M. Blanton, "Dynamic and efficient key management for access hierarchies," in Proceedings of the 12th ACM Conference on Computer and Communications Security, ser. CCS '05. ACM, 2005, pp.190–202.

[7] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, "Plutus: Scalable secure file sharing on untrusted storage," in Proceedings of the FAST 03 Conference on File and Storage Technologies. USENIX, 2003.