International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 24 Issue 7 – APRIL 2017 (SPECIAL ISSUE)

AUDITING THE FILES IN CLOUD COMPUTING

J.Veerendeswari[#1], S.Rukshana[*2], B.Nithyalakshmi[*3] and V.Indumathy[*4]

[#] *Assistant Professor, Department of Information Technology, Rajiv Gandhi College of Engineering & Technology, Puducherry, India*

[*] *B.Tech Student, Department of Information Technology, Rajiv Gandhi College of Engineering & Technology, Puducherry*

***Abstract*— It is important to provide secure keys to share the data for developing cloud computing applications. This paper proposes a public auditing with data security scheme using self-destruct Protocol. It is not feasible to implement a full life cycle privacy security to access a sensitive shared data on cloud servers. To overcome the security problem, a key-policy attribute-based encryption with time-specified attributes (KP-TSABE) is proposed. In the KP-TSABE scheme, every cipher text is labeled with a time interval while private key is associated with a time instant. The cipher text can only be decrypted if both the time instant is in the allowed time interval. The sensitive data will be securely self-destructed after a user-specified expiration time. Further web service is invoked while the file is uploaded for security by using random algorithm selection method. It provides a difficulty for hacker to hack the file.**

***Index Terms*— Auditing, Self-destruct, Cloud Computing, random algorithm.**

I. INTRODUCTION

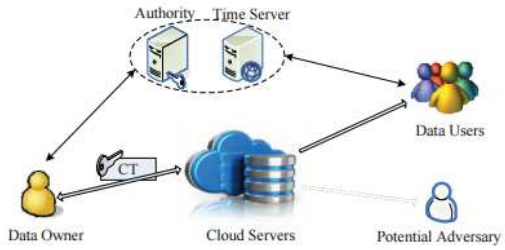
Cloud file dynamic auditing system is not available to make the data secure for the user access.

In cloud computing security is less because of larger number of users. In existing system, to achieve a security in cloud some self-destruction methods are used like Vanish, FullPP, and SSDD and so on. For a time instant key, Time Released Encryption (TRE) is used. In this system, this method does not provide a full life cycle security to the sensitive data in cloud. There is no expiring key in other self-destruction scheme which allows hacker to hack the files.

To address the fairness problem in auditing, we introduce a third-party arbitrator(TPAR) into our threat model. • which is a professional institute for conflicts arbitration and is trusted and payed by both data owners and the CSP. To overcome the security problem in cloud environment, A Key-Policy Time-Specified Attribute Based Encryption (KP-TSABE) and Web services for random multiple Encryption Algorithm is proposed. In the KP-TSABE scheme, every cipher text is labeled with a time interval while private key is associated with a time instant. The cipher text can only be decrypted if both the time instant is in the allowed time interval. The KP-TSABE is able to solve some important security problems by supporting user defined authorization period and by providing fine-grained access control during the period and it is superior to the other self-destructing scheme. While uploading file web service is invoked and select encryption algorithm randomly. Web service provides security to the file. It prevents a file from hacker to hack the file.

Data auditing schemes can enable cloud users to check the integrity of their remotely stored data without downloading them locally, which is termed as blockless verification. With auditing schemes, users can periodically interact with the CSP through auditing protocols to check the correctness of their outsourced data by verifying the integrity proof computed by the CSP, which offers stronger confidence in data security because user's own conclusion that data is intact is much more convincing than that from service providers. Generally speaking, there are several trends in the development of auditing schemes. We extend the threat model in current research to provide dispute arbitration, which is of great significance and practicality for cloud data auditing, since most existing schemes generally assume an honest data owner in their threat models. Secure self-destruction scheme A well-known method for addressing this problem is secure deletion of sensitive data after expiration when the data was used. Recently, Cachin et al. employed a policy graph to describe the relationship between attributes and the protection class and proposed a policy-based secure data deletion scheme. Reardon et al. leveraged the graph theory, Btree structure and key wrapping and proposed anovel approach to the design and analysis of secure deletion for persistent storage devices. Because of the properties of physical storage media, the above-mentioned methods are not suitable for the cloud computing environment as the deleted data can be recovered easily in the cloud servers. A data self-destructing scheme, first proposed is a promising approach which designs a Vanish system enables users to control over the lifecycle of the sensitive data. Wang et al. improved the Vanish system and proposed a secure self-destructing scheme for electronic data (SSDD). In the SSDD scheme, a data is encrypted into a cipher text, which is then associated and extracted to make it incomplete to resist against the traditional cryptanalysis and the brute-force attack. Then, both the decryption key and the extracted cipher text are distributed into a distributed hash table (DHT) network to implement self-destruction after the

update period of the DHT network. However, Wolchok et al. made a lot of experiments and confirmed that the Vanish system is vulnerable to Sybil attacks by using the Vuze DHT network [25]. So the security of the SSDD scheme is also questionable.



Architecture Working Model:

A. *File Transaction*

File Transaction is done between the owner and the user. Owner login into the server and select the user and send the file with visibility time. Time is started when the file is uploaded into the server. While uploading, key is generated and sent it to the user by message through mobile. User downloads the file by using the key before the time is expired. If user enters the invalid key for more than three times, key is deactivated automatically.

B. *Third party auditing:*

Third party auditor will audit the files and compare the files in the main server with third party server and if any file is missed matched auditor will replace the original files to main server from third party server and provides secure proof file for user access.

C. *Key Generation using Random Algorithm Selection*

In Cloud Environment Secure transaction is needed. To provide security, various random multiple encryption algorithms are implemented using web service. Web service is invoked when the file is uploaded. Advanced Encryption Standard (AES), Triple-Data Encryption Standard (Triple-DES), and RSA(Ron Rivest, Adi Shamir and Leonard Adleman)Algorithm are used as encryption algorithm. Every time file is uploaded, a different algorithm is selected to encrypt the file. Decryption key is generated based on the selecting encryption algorithm and it is sent to the user.

D. *Self-destruction of data*

Self-destructing data mainly aims at protecting the user data's privacy. All the data and their copies become destructed after a specified time, without any intervention. After user got the key, user can download the file within a specified time. User typed the key wrongly more than three times, a key is destructed. If the user needs the file, a re-request is sent to the admin. . If user missed the key more than three times, a file is deleted in server. That file is not shown in the list of files in user page. Metadata is used here to store the file temporarily. Metadata verifies the key while entering; if it is not valid a file is deleted from the metadata. Otherwise, a file is downloaded from the metadata. In KP-TSABE scheme is used for self-destruction. This scheme consists of four processes are: setup, Encryption,

E. *File Regeneration*

File Regeneration is done by admin, if the user sends a request to activate the file again. Admin accept the request and regenerate a new key by using encryption algorithm. A new key is sent to the user. A same visibility time is maintained for reactivation is given by owner. A user can give request only for three times. If it is exceeded, file is deleted from the server. After download the file, user cannot give the request again. Users can download the file before the time is expired.

II. SCREEN SHOTS

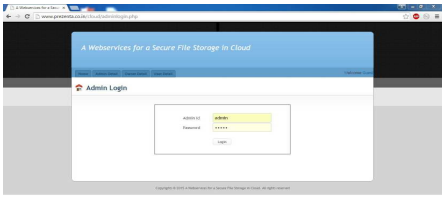


Fig 2.1 Admin login

Admin can login into the server and view the user detail and owner detail.

If user gives request to the user, admin accept the request to reactive the file again.

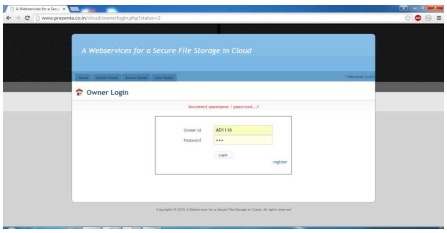


Fig 2.2 owner

Owner registers in the server and admin authorizes the owner registration, then only owner id can be registered successfully.

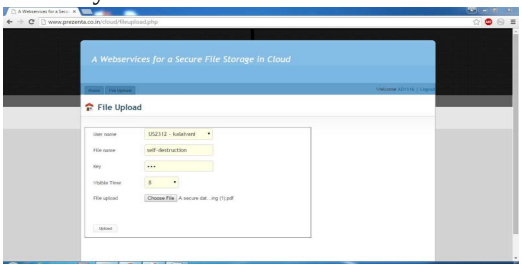


Fig 2.3 uploading file

Owner uploads the file and the file is encrypted and store into the server. A key is sent to the user by message.

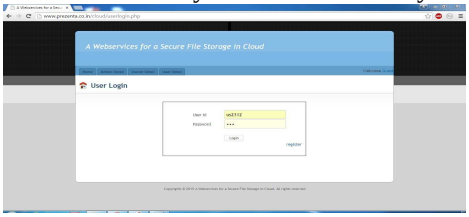


Fig 2.4 user login

User register and login into the server. User can view the list of files and can download the required file.

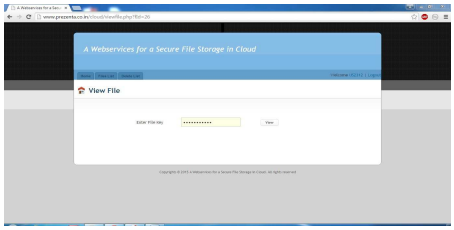


Fig 2.5 key submission

A key is sent to the user through message, when the file is uploaded into the server.

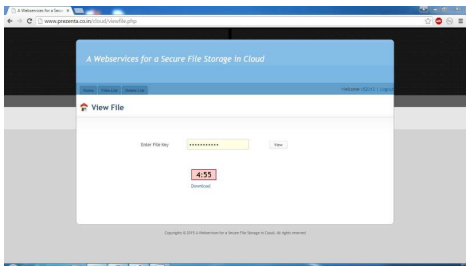


Fig 2.6 downloading the file

A file is downloaded before the timer is expired. Otherwise a file is deleted from the session. A re-request is sent to the admin.

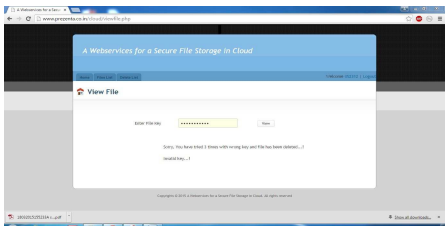


Fig 2.7 key destruction

If the Invalid key is entered more than three times, a key and the file is deleted from the server.

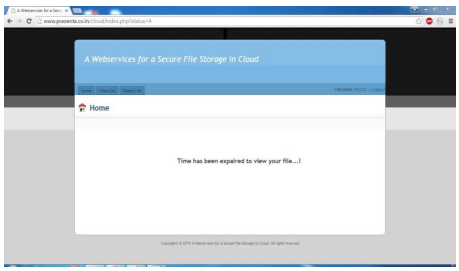


Fig 2.8 time expiration

If the timer is expired, then the file is sent to delete list.

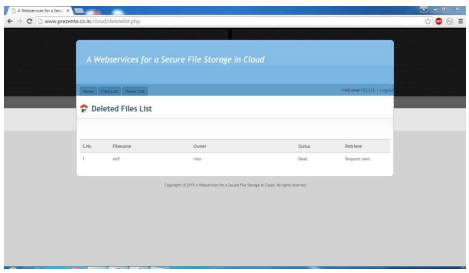
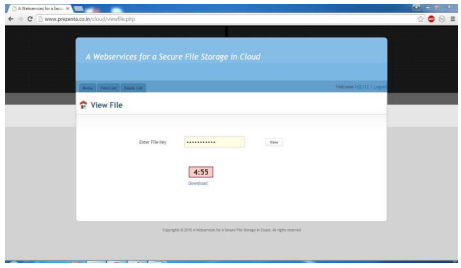


Fig 2.9 Request to admin

Again if the user sent a request, admin accept it and a key is regenerated again with same time and send to the user.



Figs 2.10 download the file

User can download the file using a new key. After file is downloaded, it will be deleted from the server.

III. FUTURE ENHANCEMENTS

Our plan to release the current self-destruct system will help to provide researchers with further valuable experience to inform future object-based storage system designs for Cloud services. In future it may be implemented in multiple server system and more security can be achieved in file transfer and database security and cloud authentication. In future, Web service system will help to provide a different approach to the researchers to provide a security in cloud environment to provide an advanced security features for data sharing. In future, it may be implemented to send the file to the multiple users at a same time. More than three algorithms in web service can be implemented in future. Also file can be destructed immediately after it is downloaded even if the key is not expired for security reasons. For security, also can propose a steganography and visual cryptography methods can be used for decrypting the key.

REFERENCES

[1] 1. W3C Working Group, “Web Services Architecture,” 2004. http://www.w3.org/TR/ws-arch/
[2] W3C Working Draft, “Soap Version 1.2 Part0: Primer,” 2001.
[3] W3C Note, “Web Services Description Language(WSDL)1.1,”2001.http://www.w3.org/TR/w sdl
[4] UDDI Spec Technical Committee Draft, “UDDI Version 3.0.2,” 2004. http://www.oasisopen.org/committees/uddi-spec/doc/spec/v3/uddiv3.0.2-20041019.htm
[5] B. Bordbar and A. Staikopoulous, “Modelling and transformation of Behavioural aspects of web services,” 3rd Workshop in Software Model Engineering (WiSME) in conjunction with UML, 2004.
[6] C. Pahl and Y. Zhu, “A Semantical Framework for the Orchestration and Choreography of Web Services,” Electronic Notes in Theoretical Computer Science, Vol. 151, No. 2, 2006, pp. 3-18. doi:10.1016/j.entcs.2005.07.033.
[7] A. N.Khana, M. L.M. Kiaha, S.U. Khanb and S. A. Madanic, “Towards Secure Mobile Cloud Computing: A Survey”, Future Generation Computer Systems, vol.29, Issues 5, July 2013.
.