Enhanced Framework for Verifying User Authorization and Data Correctness Using Token Management System in the Cloud

Balamurugan B
Assistant Professor (S G) SITE
VIT University
Vellore, India.

P.Venkata Krishna
Professor SCSE
VIT University
Vellore, India.

Nirmala Devi M
PG Scholar
VIT University
Vellore, India.

Meenakshi R
PG Scholar
VIT University
Vellore, India.

Abinaya V
PG Scholar
VIT University
Vellore, India.

***Abstract*— Cloud computing is an application and set of services given through the internet. However it is an emerging technology for shared infrastructure but it lacks with an access rights and security mechanism. As it lacks security issues for the cloud users our system focuses only on the security provided through the token management system. It is based on the internet where computing is done through the virtual shared servers for providing infrastructure, software, platform and security as a services. In which security plays an important role in the cloud service. Hence, this security has been given with three types of services such as mutual authentication, directory services, token granting for the resources. Since, existing token issuing mechanism does not provide scalability to large data sets and also increases memory overhead between the client and the server. Hence, our proposed work focuses on providing tokens to the users, which addresses the problem of scalability and memory overhead. The proposed framework of token management system monitors the entire operations of the cloud and there by managing the entire cloud infrastructure. Our model comes under the new category of cloud model known as "Security as a Service". This paper provides the security framework as an architectural model to verify user authorization and data correctness of the resource stored thereby provides guarantee to the data owner for their resource stored into the cloud. This framework also describes about the storage of token in a secured manner and it also facilitates search and usage of tokens for auditing purpose and supervision of the users.**

Keywords— Token Management System, Access control

I. INTRODUCTION

Cloud is not a physical thing. A network of servers with each server having a different function is cloud. To "deliver a service" or to run applications, some servers use computing power. It provides a platform that provides service based on the demands and also provides service to the users ubiquitously. The issues that are addressed in the cloud are managed by means of security controls. And Deterrent control, Preventative Controls, Corrective Controls, Detective Controls are some of the cloud security control architectures. This type of service falls under the new category of Security as a service modeling architecture in the cloud. And this type of service models focuses more on providing security to the resources and the cloud data. This type of service scheme is also compatible with other types of service models in order to allow all type of users to access the cloud efficiently. The usage of cloud provides more opportunities and challenges to users of the cloud with its services and security mechanisms against all type of attacks to the resource. In order to guarantee the security for the resources in the cloud, data owner are provided with some identity for their resource hence they can verify or audit their resource events and usage by providing the valid identity provided for that resource. In such cases tokens are generated and issued to all the users of the cloud data to determine the status of the resource to its data owner. And also this token represents an identity of the user getting into the cloud. This type of token methodology also eases the verification process for the resource and also involves recovery process to modify the corrupted file with its original version efficiently by validating the token generated for the resource provided by the data owner. And, also token provided to the user and the resource helps in the process of auditing by the cloud service provider hence increases the performance and reliability factor in processing the resource in the cloud by the user.

Fig.1. explains System Every accesses of the user are tracked and assigned with a token for all the events processed. Token Management System (TMS) acts as an intermediate between the users and the resource in the cloud hence validate the user with their token in order to allow access to the resource. By proposing this type of scheme, both the users and

978-1-4799-2397-7/14/$31.00 ©2014 IEEE

the resources are validated and guarantee the security to both the users and resources in the cloud.

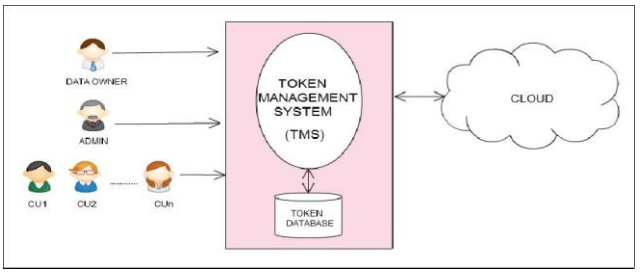


Fig.1. Overview of Token Management System

II. RELATED WORK

A. PRELIMINARIES OF TMS

The cloud computing generally includes various storage servers which are used by the cloud service provider (CSP), Cloud users (CU), Cloud data owners (DO), and several other application and infrastructure level people. There are many entities in the cloud that are being used by many people. TMS is placed as an intermediate between the cloud data and the users. TMS captures the events made by the cloud users. For every action by the user, token will be generated and stored in the token database. This is used later to audit and analyze the user behavior. If a user CU1 logs into the system then, TMS captures this action as a process or an event (P1). Now, a token is generated and stored into the database. Then, if he/she does some more actions after login then those events are captured as more processes (P2….Pn) and stored in the token database. Once he logged out from the cloud, then the end token is generated automatically and stored. A search step is also included, in case, if the auditor needs complete event list of the cloud users. This is done by searching the token database for a particular user's token and analysis is made on the generated tokens to the user.

In Khaled et al. [1], they proposed the scheme for access control for RDF data by providing tokens to the cloud users. This mechanism involves issuing tokens to the users based on their needs and authorization level. However, this methodology involves serious limitation in terms of scalability [5] when there is a large data-set associated with the cloud. Ahmed et al. [2], involves providing trusted ticket to the registered users generated by the data owner of the resource. This scheme involves encrypting the data with secret key by the data owner to ensure the correctness [6] of the data. Here the trust ticket factor has been designed with encrypted expression such as resource's data owner, user id, access rights and time expiration of trust ticket. Only registered users are provided with the secret key [7] and the trusted ticket for accessing the resource. Here, ticket plays as the mediator between CSP and the users through data owner. However, it encounters a problem in which if users share the secret key with CSP and by which, malicious attack can be performed on the resource. In Hao et al. [3], involves providing digital tickets to the user based on their smart card identity [8]. It is proved that proper mutual authentication is needed between users and cloud servers before all other services that are provided by cloud to the users. Hence, this technique reduces the server overhead by not allowing other users to share the ticket [9], thereby assurance of identity is provided for both the server and the client. However, this scheme can resist to some attack, if the ticket is lost by the user and it cannot be recovered or prevented from attacking the resource in the cloud. In Pippal et al. [4], proposed a scheme which provides more security by overcoming the limitation proposed in Hao et al.'s scheme [3]. Authentication is performed mutually to provide security against attacks. This technique guarantees security [10] to the user by allowing them to modify the password securely [11] without CSP assistance. This scheme holds good for the memory overhead issue with the help of time bound ticket based mutual authentication.

III. LIMITATIONS

The limitations found in the existing solution are: In [1], Khaled et al., it is found that the proposed scheme scales poorly to large data sets and it seems to be serious limitation as, cloud involves various data sets associated with it and it is overcome by providing more scalability through token management system. Ahmed et al.,[2] in which the proposed work encounters a problem if user share the secret key with CSP and by which, malicious attack can be performed on the resource. In [3], Hao et al., it is found that the proposed scheme can resist to some attack, if the ticket is lost by the user and it cannot be recovered or prevented from attacking the resource in the cloud

A. Summarization of Limitation

The mutational based token issuing [4] has limitation such as error recovery problem and in some critical situations it does not provide appropriate prevention to malicious attacks. Thus increases the memory overhead and this reduces the performance and the efficiency, correctness of the model (token generation). The model is adaptable only for the smaller datasets. Inconsistency occurs in case of large data sets. And, also no proper model is yet described or built to prevent the malicious attacks in critical situations.

IV. PROPOSED WORK

The limitations above are needed to be avoided in our real time scenarios. Fig 2. Explains the overall the proposed framework proves that maintaining token management system holds well for large data sets and for the critical situation to

prevent the malicious attacks. And it also reduces the memory overhead and error recovery problem.

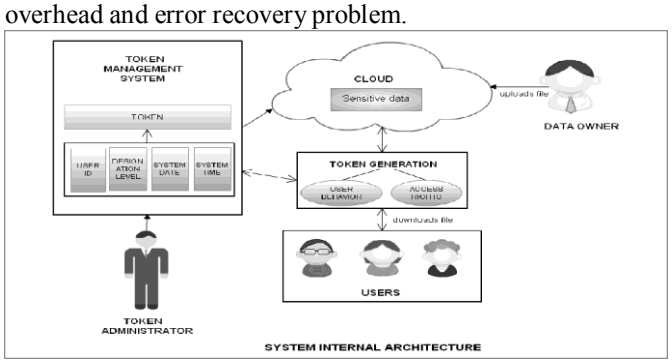


Fig.2. Proposed System Architecture

A. Token Generation

Tokens (t) are computed based on the factors such as user name, designation level of user, and the events users performed in the cloud. Here, tokens are generated for every actions of the user in the cloud. These are captured as events and an event list is maintained. Hence, issued tokens are used to validate the user with all the events processed in the cloud. Tokens are generated with two characters of upload and download identities. These are made as XX when the user logs in into the system. And these two digits are turned into value if the user uploads or downloads any resource from the cloud. And also tokens include date and time of events that has been processed in the cloud by the user. Hence, the limitation found in Khaled et al. [1], are solved by generating token based on the user behavior and their performance over the resource in

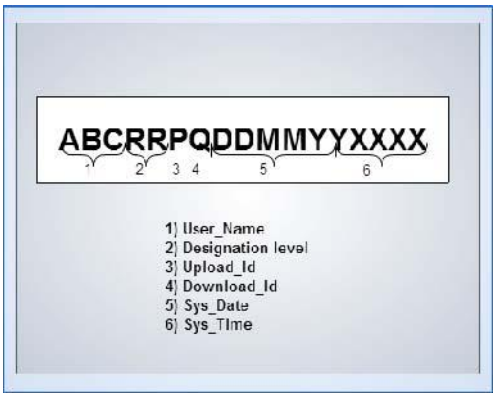


Fig.3. Token Format

Table 1:DESCRIPTION OF TOKEN

TYPE	DESCRIPTION
User_Name	The first three characters of token represents the name of the user logged into the cloud.
Designation level	The next two characters of token indicates designation level of the user.
Upload_Id	Sixth character of issued token indicates whether user has uploaded any resource into the cloud
Download_Id	Seventh character of token represents the value if the user has downloaded any resource. And, it holds the value as R,W,A based on the users mode of access.
Sys_Date	The next six characters of token indicates date at which the user has been logged in.
Sys_Time	The next six characters of token indicates time at which the user has been logged in and other actions.

B. Token Computation

UN=Substring of Username.

DL= Designation Level of the user.

UP_FLAG=Flag to represent uploading file.

DOWN_FLAG=Flag to represent downloading file.

SYS_DATE=Current system date.

SYS_TIME=Current system time.

t=Total no. of processing time.

Token, t=

$$\sum_{i=1}^{t}((UN).(DL).(UP_{FLAG}).(DOWN_{FLAG}).(SYS_{DATE}).(SYS_{TIME}))$$

t

C. Detection of authorised users

Tokens are used to detect the user's behavior by validating and measuring the performance of the user in the cloud. If, users tends to be unauthorized by their behavior they are deleted or de allocated from using the resources of the cloud. Hence, access control is provided in terms of token validation scheme, which determines that the user with his/her generated token are valid. And, this token management system is also used in the process of auditing the cloud user and the resources by all the computations that the user performed in the cloud.

D. Algorithm

Token t= null.

User (ABC) logs in.

Token t is generated (upload and download fields will be XX)

Now, t= ABCTLXX1012132203

If user uploads a file

Token of upload field=1 is generated (Download field will be XX)

Now, t= ABCTL1X1012132215

Else if, user uploads another file

Now, t= ABCTL2X1012132220

Else if user downloads a file

Then, t= ABCTLX11012132222

If user does processes events

Then, Token‘t’ is created with date and timings (the action done is described in text).

E. Error Recovery

If any resource in the cloud has been corrupted with an unauthorized access, then tokens issued to all the users who accesses the resource prior to its corruption are provided to the data owner. Thereby, user, who corrupts the resource, can be verified easily and efficiently. And, also corrupted file can be recovered with its originated file and replaced with the corrupted one and its send to the data owner of the resource. Hence, the limitation found in Hao et al. [3], are solved by allowing recovery process over the corrupted file using the token issued to the user.

1. The framework generally describes the entire process done by the user performing different process on cloud under the organization level or infrastructure level.
2. The person needs to login into his or her own system and perform certain events that he/she needs to perform and logs out after the process has been completed.
3. The process done by the person may be uploading a file, downloading a file or updating the files present in the cloud.
4. The process is followed by the TMS, which generates the token and the token database stores the tokens generated for every process of a particular user.

V. VALIDATION

Validation process involves checking whether the generated token for the user describes all the actions or events that the user performed in the cloud precisely. And, also token is validated to determine its accurateness in detecting the user behavior by the cloud service provider.
If User “ABC” logs into the cloud, then

String token= ABCTLXX1012132203

String Uname= token.get(uname());

String dlevel= token.get(dlevel(Uname));

String Up_id= token.get(up_id(Uname_event));

String Down_id= token.get(down_id(Uname_event));

Updated token is stored into the token database and new token is generated based on the user accesses on the cloud.

String gen_token=((Uname).(dlevel).(Up_id).(Down_id).(Sys_now))

If token is equal to gen_token

Then token is valid.

Else

Generated token is invalid.

The figure 4 shows the experimental results done over the cloud without token and the results are not of satisfactory level as it consumes superfluous time .on the other hand Fig .5 shows a better search time as the proposed method used token based management for cloud services.

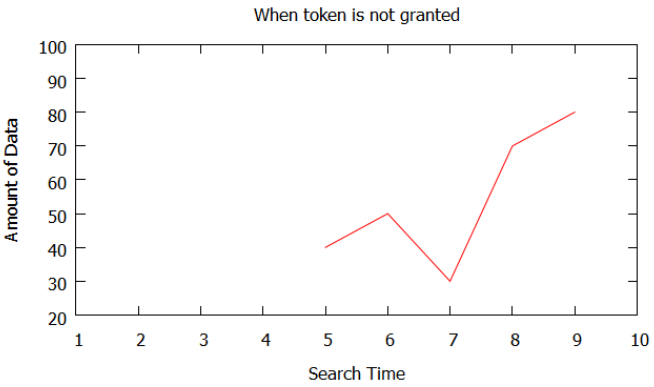


Fig.4 Token less Cloud access Management

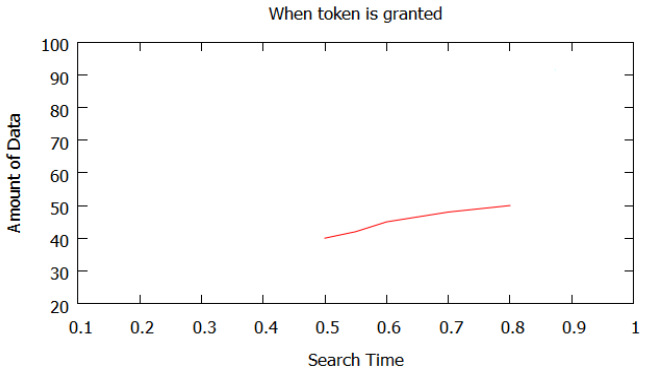


Fig.5 Token based Cloud access management

VI. CONCLUSION AND FUTURE WORK

Our work represents providing tokens to the specific users of the system by modeling a “TOKEN MANAGEMENT SYSTEM”. Here we generate a token based upon the algorithm which depicts the generation of token based on the user attributes such as user id, system date and time etc. The token generation, thus reduces the memory overhead and restricts unauthorized access even in the critical situations. This technique is also used in auditing processes and analyzing the user behavior. Also, token issuing methodology

gives assurance for both the server and the client identities and prevents the attack against the resources very efficiently by detecting user behavior before allocating the resource to the users. And, this method is validated and the result confirms the correctness and efficiency of the proposed framework. This proposed work prevents the unauthorized users from accessing the resource, but, it doesn't able to prevent the authorized user from accessing the resource unsustainably and cannot identifies the server position through which unauthorized access to the resource has been performed. Hence, our future work focuses on solving the limitation of this work. Thus this paper concludes by promising the cloud user and data owner to have more security for their resources and also reduces the overhead between client and the server.

REFERENCES

[1] A.Khaled, M.F.Husain," "A Token-Based Access Control System for RDF Data in the Clouds", Department of Computer Science, University of Texas at Dallas.

[2] M.Ahmed, Y.Xiang," Trust Ticket Deployment: A Notion of a Data Owner's Trust in Cloud Computing", School of Information Technology, Deakin University Burwood, VIC 3125, Australia.

[3] Hao, Z., Zhong, S. and Yu, N. (2011). "A time-bound ticket-based mutual authentication scheme for cloud computing". International Journal of Comput-ers, Communications and Control, 6(2), pp. 227–235.

[4] R.S.Pippal, Jaidhar C.D, S.Tapaswi," Enhanced Time-Bound Ticket-Based Mutual Authentication Scheme for Cloud Computing", Informatica 37 (2013) 149–156.

[5] H. Choi, J. Son, Y. Cho, M. K. Sung, and Y. D. Chung. "SPIDER: a system for scalable, parallel / distributed evaluation of large-scale RDF data". In Proc. 18th ACM Conference on Information and Knowledge Management (CIKM), pages 2087–2088, Hong Kong, China, November 2009.

[6] M. Ahmed, Y. Xiang, and S. Ali, "Above the Trust and Security in Cloud Computing: A Notion Towards Innovation," in Trust Com 2010: 2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, Hong Kong, 2010, pp. 723-730.

[7] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing," IEEE INFOCOM 2010, 2010.

[8] T. Messerges, E. Dabbish, and R. Sloan, "Examining smart-card security under the threat of power analysis attacks," IEEE Transactions on Computers, vol. 51, no. 5, pp. 541–552, 2002.

[9] I.-C. Lin and C.-C. Chang, "A countable and time-bound password-based user authentica-tion scheme for the applications of electronic commerce," Information Sciences, vol. 179, no. 9, pp. 1269 – 1277, 2009.

[10] Shen, Z. and Tong, Q. (2010). "The security of cloud computing system enabled by trusted computing technology". In Proceedings of 2nd International Conference on Signal Processing Systems, Wuhan, China, pp. 11–14.

[11] Chang, C.C. and Wu, T.C. (1991). "Remote password authentication with smart cards. IEE Proceedings E:Computers and Digital Techniques", 138, pp. 165–168.

[12] M.Hussain, H.Abdulsalam," SECaaS: Security as a Service for Cloud-based Applications". College of Computer and Information Technology, Taif University, Saudi Arabia.

[13] P.Syam kumar ,R.subramanian and D.Thamizh Selvam ":Ensuring Data storage in Cloud Computing using Sobol Sequence Parallel Distributed and Grid Computing 2010 1st International Conference.

[14] Tribhuwan,M.R,Bhuyar,V.A and Pirzade.S "Ensuring Data Storage Security in Cloud Computing through Two-Way Handshake Based on Token Managemnt".ARTcom,2010 International Conference.