ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

Survey on Token Based Authentication in Cloud Computing

Unnati Awasthi, Prof. Ashok Verma

Research Scholar, Dept. of Computer Science & Engg, Gyan Ganga Instt. of Tech.& Sciences, Jabalpur, India

Prof. and Head of Dept, Dept. of Computer Science & Engg, Gyan Ganga Instt. of Tech.& Sciences, Jabalpur, India

ABSTRACT: Cloud computing is a general term used to describe a new class of network based computing that takes place over the internet. Secure outsourcing of calculation to an un-trusted (cloud) administration supplier is turning out to be more critical. Immaculate cryptographic arrangements in view of completely homomorphism and undeniable encryption, as of late proposed, are promising however experience the ill effects of high dormancy. Trusted registering (TC) is another promising methodology that uses trusted programming and equipment parts on figuring stages to give helpful instruments, for example, confirmation permitting the information proprietor to check the uprightness of the cloud and its calculation. Trust Multi-occupancy and trusted figuring in view of a Trusted Platform Module (TPM) are incredible advances for unravelling the trust and security worries in the cloud personality environment. Single sign-on (SSO) and OpenID have been discharged to take care of security and protection issues for cloud personality. Single Sign-On (SSO) is a verification system in which a cloud administration purchaser should be confirmed just once while getting to different administrations from numerous administration suppliers, or while getting to various administrations from the same administration supplier.

KEYWORDS: Session, JavaMail, PKI, SSO, Authentication, Public Cloud, Server-Side-Encryption

I. INTRODUCTION

Recently, cloud computing has gained a considerable acceptance as a promising model from both business and academic communities. It is a representation for empowering pervasive, convenient, on - request arrange right to use to a mutual pool of configurable registering assets (e.g., systems, servers, stockpiling, applications, and administrations) that can be quickly provisioned and discharged with ostensible administration exertion or administration supplier's cooperation. Cloud administration suppliers (CSP's) offer cloud stages for their clients to utilize and make their web administrations, much like network access suppliers offer costumers fast broadband to get to the web.

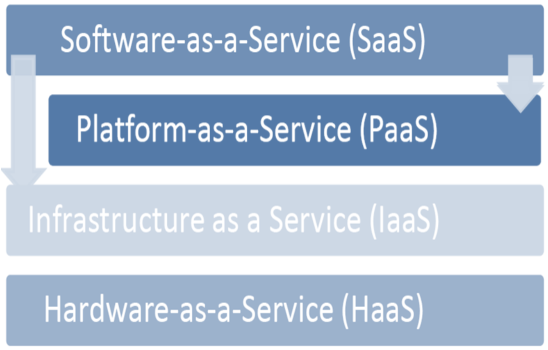
Software-as-a-Service (SaaS)
Platform-as-a-Service (PaaS)
Infrastructure as a Service (IaaS)
Hardware-as-a-Service (HaaS)

Figure 1 Cloud computing service

The client environment may be a native application or browser-based; the increasing power of the browser is available to many client devices, mobile and desktop alike. Robust capabilities in many mobile devices, the increased demand on networks, the cost of networks and the need to manage bandwidth use creates incentives, in some cases, to minimize the cloud application computing and storage footprint, and to exploit the intelligence and storage of the client device. However, the increasingly complex demands of mobile users will drive apps to demand increasing amounts of server-side computing and storage capacity.

II. AUTHENTICATION PRINCIPLES

Having set up the premise on which the protected territory is fabricated, we will now take a gander at the main issue – how to confirm clients. Security intelligence says there are three sorts of validation [9]:

- Something you know, e.g. PIN or password
- Something you have, e.g. credit card or secure ID token
- Something you are, e.g. photograph or biometrics

Sometimes the method in use is not obvious. For example, a key for a door lock would seem to be “something you have”. But a locksmith can make any key they know the shape for, so to a locksmith this is “something you know” authentication.

2.1 Passwords

In IT most verification is "something you know" - as a rule a secret key.. Passwords are prominent in light of the fact that they are moderately simple to manage and offer sensible security. Beside clients overlooking passwords, secret key verification has two principle issues. Firstly, clients regularly pick feeble passwords and re-utilize the same watchword on numerous different frameworks. Also, the secret key must be entered in full every time the client sign on – and on the off chance that it is caught that gives the aggressor complete control.

2.2 Phishing

A minor departure from the watchword catch issue is that clients can be deceived into uncovering their secret key. This has turned into a noteworthy concern as of late, with numerous "phishing" assaults being dispatched against budgetary foundations. A few reports recommend upwards of 5% of focused clients have been deceived into uncovering their subtle elements. The principle answer for phishing assaults is client training. Clients must be prepared not to give individual information in light of a spontaneous email. Sites ought to backing this by not themselves conveying such demands by email. However numerous locales do in any case send messages, for example, "your bill is prepared at this URL" - and the URL requests a client name and watchword. There might be specialized answers for this issue. One choice would be for system heads to square known phishing locales. Then again, different program bars are accessible, for instance the Mozilla Trust Bar. This shows the genuine area of a page, in a way that is (ideally) difficult to parody. Trusted areas can be arranged and will show in an unexpected way - so the client can rapidly check whether it is sheltered to enter individual information.

2.3 Single Sign-On

A rising arrangement an excess of the issues with passwords is single sign-on. As opposed to confirming independently with each site, the thought is to verify once with a focal power. Different sites believe this focal full power to give validation. The potential points of interest are numerous; including lessened organization Single sign-on can take different structures. A Windows area controller is one sample for sites and less passwords for clients to recollect. Single sign-on doesn't need to be founded on passwords, however it generally is. Be that as it may, no arrangement so far has anything 100% take-up. Sites must make procurements for clients that don't have single sign-on character. One alternative is to oblige them to make such a personality. Be that as it may, this basically constrains a specific innovation on clients and may not be well known. An option is to just offer single sign-on as an alternative, with clients as yet having the capacity to have a client name and secret word particular to the site.

2.4 SSL Client Certificates

Somewhat utilized component of SSL is the capacity to have customer authentications. This is an option "something you know" strategy. The customer can demonstrate their character by displaying an endorsement and reacting to a scrambled message. This takes care of huge numbers of the issues connected with passwords. The customer is not uncovering their private key, simply demonstrating that they know it. It is secure to utilize the same endorsement for some sites, completely© tackling the secret key re-uses issue. This additionally takes care of the secret key catch issue, including phishing. This implies the client can just sign in from their own PC not from partners or in the library and these sums to a noteworthy restriction.

III. ATTACKS ON SYSTEM

3.1 Session Fixation

The thought of "session obsession" is for the assailant to set the session ID before the client sign in. At the point when the client signs in, this session ID will be moved up to "signed in" status. Be that as it may, the assailant still knows it, and can now utilize it to access the ensured territory [10]. For URL parameters altering the session ID is simple assailant rights tempts the casualty to click a connection that contains the aggressor's picked session ID. The web server will then accommodatingly rework every one of the connections on that page to utilize the same session ID. On a basic level it ought not to be workable for an assailant to control treats on the casualty space. Be that as it may, we will in no time take a gander at different vulnerabilities that allow this. Be that as it may, HTTP validation is not powerless against this assault as there is no session ID and the secret key is required for each solicitation. There are a few ways an assailant could control another site's treat. By abusing a XSS weakness in the casualty site, the assailant can utilize JavaScript [9].

3.2 Injecting Cookies

In fact the objective posts are augmented to some degree by a component of treats. A web server can set the "area" quality on a treat, for instance to "example.com" rather than "www.example.com". So any XSS weakness in the same area as the casualty site could permit treat infusion. I chose to examine the space quality further and found weakness in some mainstream web programs. It is 200 planned that a server can set a treat for its own particular area, yet not for others [8]. Along these lines, www.example.com can set a treat for example.com yet not for victim.com. I thought about what might happen whether it attempted to set a treat for .com. Things being what they are .com Institute is not permitted but rather the confinement is not great. For illustration, it would be workable for www.attacker.ltd.uk to set a treat for .ltd.uk. This would then be sent to www.victim.ltd.uk. I called this powerlessness "cross-space treat infusion". There is likewise an issue with the utilization of SSL. A non-SSL solicitation can bring about a treat to be set, which will later be sent with SSL asks. I called this helplessness "cross-security limit treat infusion". Abuse is troublesome: the aggressor needs to DNS store harm the casualty as depicted before. The casualty is then lured to click a connection to the non-SSL casualty site, and the aggressor blocks this to set a treat. Later, when the casualty visits the SSL site, they are as yet utilizing the session ID known by the assailant [4].

3.3 Performing a Session Fixation Attack

Envision we are going to perform a session obsession assault against www.victim.ltd.uk. We take a gander at the site and see it utilizes a "sid" treat to track sessions. We record the session ID the web server has apportioned us. Presently we get another ltd.uk area; we pick attacker.ltd.uk and register this truly [5]. We set up a web server at www.attacker.ltd.uk and tempt the casualty to click a connection to this space. The web server then sets the "sid" treat, with the area ".ltd.uk". Due to deficient checking in some web programs, the treat is permitted. We additionally set a long timeout on the treat – so the client won't be allocated another session ID before we have a chance to assault account and that session ID will be set apart as signed in. We will then be allowed access to the secured range. We continue getting to the casualty web server utilizing our recorded session ID. At first we are dealt with as not signed in.

3.4 Brute Force Attacks

A straightforward sort of assault is to consequently attempt immense quantities of client names and passwords. Verification frameworks have since quite a while ago confronted this danger and taken a few countermeasures:

- Insert a delay between receiving credentials and responding success/fail.
- Lock an account after a certain level of incorrect logins is reached.

Unfortunately, neither of these countermeasures can truly be utilized on the web. On the off chance that a record is bolted after erroneous logins then this permits an assailant to effortlessly bolt individuals' records successfully a dissent of administration assault. The login deferral is not compelling in light of the fact that an assailant can endeavor numerous logins on the double, and if synchronous logins are prohibited this again opens up a potential DoS assault. An option methodology is to square IP addresses after a few fizzled logins from the same location [9]. This is convoluted by the way that numerous clients might seem, by all accounts, to be originating from the same Institute IP address, e.g. an ISP's web intermediary. For this situation an assailant can bring about DoS against clients of the same intermediary. This is not as terrible as hindering the entire Internet, but rather is still an issue. Along these lines, putting limits on login disappointments is a harmony between forestalling beast power assaults and avoiding foreswearing of administration assaults. A sensible equalization I'd recommend is to put a genuinely tight confinement on login disappointment per IP address, e.g. 10 disappointments in a 5 minute period locks IP address for 5 minutes. Put a higher confinement on disappointments per account, e.g. 1000 disappointments in a 24 hour period locks represent 24 hours. Thusly just a determined© assailant can bolt a record – yet records will be bolted sufficiently early to forestall password compromise.

IV. SERVICE ORIENTED ARCHITECTURE

Day by day as innovation is getting to be utilized as a part of basic, usage of frameworks is getting to be intricate as far as security unwavering quality and multifaceted nature in stage in light of the fact that the same number of frameworks are being coordinated more setups are getting to be decentralized [2]. Decentralization of framework accompanies one essential issue that is the means by which every framework will speak with other framework. Remote Procedure Call (RPC) conquers this issue to some degree yet it additionally has a few confinements like innovation consistency all through framework and coupling for little application or it was calm oversee however as framework develops and needs augment corporate prerequisites changes which made to work with one innovation furthermore their correspondence interfaces these issues had made requirement for innovation which work like Remote Procedure call yet no call technique ought to be uniform in framework which ever inward innovation is utilized. Here comes the idea of Service Orient Architecture to be utilized as a part of Development of frameworks and application. Administration Oriented Architecture (SOA) is essentially a configuration standard which is utilized to coordinate and execute complex framework. In this construction modeling each article is administration which is full framework in itself so to add to a major framework simply needs to coordinate distinctive Services like joining building Block. Clearly there must be some interface of collaboration with one another in these administrations [7]. Fundamental standard of SOA advancement is Reusability, Granularity, Interoperability and Modularity. By utilizing SOA if bit by bit diminish your expense of improvement an application because of simply module play of various administrations. Already Remote Procedure Call (RPC) was the setup to summon some activity on remote framework however it was likewise having its own particular restriction like stage reliance and couplings. In this entire decentralize System usage there must be focal Register which is having data about all administrations. So Service Consumer can get data about which Service will be exact for it to give want result. Taking after figure is graphical presentation of Service Oriented Architecture (SOA) [3].

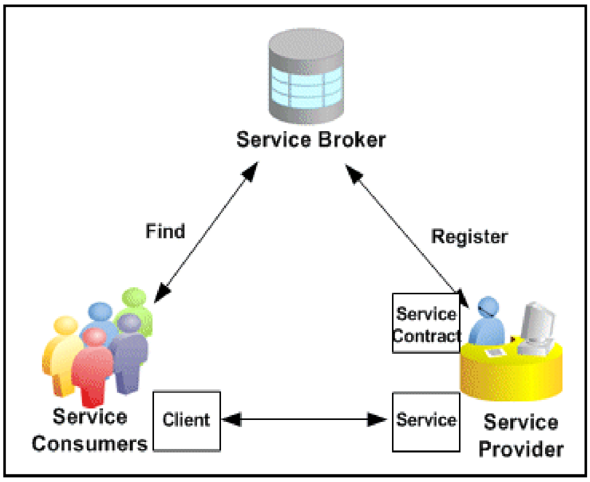

Figure 1.2 Service Oriented Architecture (SOA)

There are constantly two sorts of Services in this Architecture one is Service Provider and second is Service Consumer. Each Service Provider Register itself to Public Register which is open to each Service Consumer of System. At the point when any Service Consumer needs some data from other administration it first checks register which Service can give its longing data. After affirmation of Service Provider, Register gives back the interface quality to purchaser by which data can be assembled from any Web Service.

V. SESSIONS

The stateless way of HTTP requires associations and arrangement engineers to discover different systems for exceptionally following a guest through a web-base application. Different techniques for dealing with a guest's session have been proposed and utilized, yet the most famous strategy is using one of a kind session IDs. Tragically, in an excess of cases associations have mistakenly connected session ID administration strategies that have left their "protected" application open to mishandle and conceivable capturing [3]. This archive audits the basic suspicions and imperfections associations have made and proposes routines to make their session administration more secure and vigorous. The most widely recognized strategy for assigning so as to follow a client through a site is an interesting session ID and having this data transmitted back to the web server with each solicitation. Shockingly, ought to an aggressor figure or take this session ID data, it is an insignificant activity to commandeer and control another client's dynamic session. An imperative part of accurately overseeing state data through session IDs relates straightforwardly to verification forms. While it is conceivable to demand that a customer utilizing an associations web application give confirmation data to each "confined" page or information accommodation, it would soon get to be repetitive [8]. Along these lines session IDs are not just used to take after customers all through the web application, they are likewise used to extraordinarily recognize a confirmed client in this way in a roundabout way managing access to website substance or information.

5.1 Maintaining State

Ordinarily, the procedure of dealing with the condition of an electronic customer is using session IDs. Session IDs are utilized by the application to particularly distinguish a customer program, while foundation (server-side) procedures are utilized to relate the session ID with a level of access. In this manner, once a customer has effectively confirmed to the web application, the session ID can be utilized as a put away verification voucher so that the customer does not need to retype their login data with every page demand. Associations application designers have three routines accessible to them to both designate and get session ID data:

- Session ID information embedded in the URL, which is received by the application through HTTP GET requests when the client clicks on links embedded with a page.
- Session ID information stored within the fields of a form and submitted to the application. Typically the session ID information would be embedded within the form as a hidden field and submitted with the HTTP POST command.
- Through the use of cookies.

Every technique has certain points of interest and burdens, and one might be more proper than another. Choice of one technique over another is generally subordinate upon the sort of administration the web application is to convey and the target group. Recorded underneath is a more itemized examination of the three routines. It is critical that an associations framework engineers comprehend the confinements and security ramifications of every conveyance instrument.

5.2 The Session ID

A critical part of overseeing state inside of the web application is the "quality" of the session ID itself. As the session ID is regularly used to track a verified client through the application, associations must know that this session ID must satisfy a specific arrangement of criteria in the event that it is not to be bargained through prescient or animal power sort assaults. The two basic qualities of a decent session ID are arbitrariness and length.

i. Session ID Randomness

It is essential that the session ID is unusual and the application uses a solid strategy for creating irregular ID's. It is imperative that a cryptographically solid calculation is utilized to create an exceptional session ID for a validated client. Preferably the session ID ought to be an irregular value.

ii. Session Hijacking

As session ID's are utilized to recognize and track a web application client, any assailant who acquires this extraordinary identifier is conceivably ready to present the same data and mimic another person this class of assault is usually alluded to as Session Hijacking. Given the innate stateless nature of the HTTP/S convention. An aggressor has available to him three strategies for picking up session ID data perception, animal power and confusion of trust.

iii. Observation

As a matter of course all HTTP activity crosses the wire in a decoded, plain content, mode. Subsequently, any gadget with access to the same wire or shared system gadgets is prepared to do "sniffing" the movement and recording session ID data (also client validation data, for example, client names and passwords). Likewise, numerous edge gadgets consequently log parts of HTTP activity – specifically the URL data. A straightforward security measure to avert "sniffing" or logging of classified URL data is to utilize the encoded type of HTTP – HTTPS.

iv. Brute Force

In the event that the session ID data is produced or displayed so as to be unsurprising, it is simple for an aggressor to more than once endeavour to figure a legitimate ID. Contingent on the haphazardness and the length of the session ID, this procedure can take as meagre time as a few moments. In perfect circumstances, an assailant utilizing a household DSL line can possibly lead up to upwards of 1000 session ID surmises for every second. Subsequently it is vital to have an adequately perplexing and long session ID to guarantee that any reasonable animal constraining assault will take numerous many hours to foresee.

iv. Misdirected trust

In perfect circumstances, a customer's web program would just ever reveal secret session ID data to a solitary, trusted website. Sadly, there are various occasions when this is not the situation. For instance – the HTTP REFERER field will send the full URL, and in a few applications this URL might contain session ID data. Another mainstream strategy, using basic trust relationship defects, are HTML inserted and Cross-site Scripting (CSS or some of the time XSS) assaults. Through astute installing of HTML code or scripting components, it is conceivable to take session ID data regardless of the fact that it is held inside of the URL, POST fields and treats. Peruses requiring more data about this class of assault ought to survey a duplicate of "HTML Code Injection and Cross-site scripting".

v. Common Failings

While electronic session administration is vital for following clients and their route all through an application, the most basic use is to keep up the state data of a verified client as he does his permitted capacities. For internet saving money and retail situations, utilizing a properly solid session administration strategy is pivotal to the achievement of the association. Based upon these examinations, this segment points of interest the absolute most basic failings and suppositions that have been made.

vi. Predictable Session ID’s

The most widely recognized blemish in session ID use has dependably been consistency. As talked about before, the two reasons are an absence of haphazardness, or length, or both.

- **Sequential allocation of Session ID’s –** Each guest to the site is apportioned a session ID in consecutive request. In this manner, by watching your own session ID data, the basic routine of supplanting it with another quality a couple of emphases up or down will permit the assailant to mimic another client.
- **Session ID values are too short –** The full range of valid session ID’s could be covered during an automated attack before there is time for the session to expire.
- **Common hashing techniques –** While numerous business web administrations have worked in capacities for ascertaining hashed data, these instruments are surely understood and accessible for generation. A hashing capacity will in fact make a session ID esteem that gives off an impression of being one of a kind and incredible consideration ought to be taken to guarantee that predicable data is not utilized as a part of the era of the hash. For instance, there have been situations where the "special" hash was based upon the neighborhood framework time, and the IP location of the

uniting host. Utilizing the same hashing capacity, the assailant would have the capacity to pre-compute a substantial number of time dependant hashes for a prevalent web entry or intermediary administration (i.e. AOL), and use them to savage drive any existing session from that service.

- **Session Obfuscation –** The use of a custom method of obscuring data and using it for session management. It is never a sound idea to include client or other confidential information within a session ID. For example, some organizations have even tried encoding the user's name and password within the session ID using a shifted Unicode and hexadecimal representation of the information.

vii. Insecure Transmission

For banking and retailing applications it is urgent that all private material and session data be transmitted safely and not defenceless against perception or replay assaults. Sadly numerous business bundles have fizzled in the past to secure the trustworthiness of their session administration because of shaky transmission.

- **Use Encryption when sending session information –** As specified prior, there are a great deal of occasions whereby a clients association with the application server will be logged if not sent over an encoded channel, for example, HTTPS. This is especially imperative for applications that require high a level of privacy. On the off chance that utilizing the treat technique for overseeing session IDs, associations ought to note that the customer program will present the session ID with each demand (this incorporates pages and representation) and might even submit it to different servers inside of the same area – which could conceivably be done over a protected information channel.
- **Use different session ID's when shifting between secure and insecure application components –** As another client explores the web application as a "visitor", utilize an alternate session ID than what might be allotted in the safe part of the application. Never utilize the same session ID data in the validated and unauthenticated segments of the web application. Once more, guarantee that the session ID to be utilized as a part of the safe part of the web application is not unsurprising and in light of the past ID.
-

viii. Length of Session Validity

For secure applications all session information should be time limited and allow for client-side cancellation or server-side revocation.

- **Client Cancellation –** Many web applications neglect to take into consideration customer side cancelation, for example, "log-out". In the event that the expectation is to permit clients to collaborate with the application from anyplace, including Internet Cafes, associations should know that different clients can utilize the same machine and trawl through the "history" and reserved page data. On the off chance that the session has not been crossed out, it is an unimportant activity for the following client of the PC to "resume" the last association.
- **Session Timeout –** Again, when managing the likelihood of shared customer PCs, it is critical that there is a restricted lifetime (or time of dormancy) after which the session will naturally terminate. The expiry time ought to be kept to a base period, and is indigent upon the way of the application. In a perfect world the application ought to be fit for observing the time of inertia for every session ID and have the capacity to erase or renounce the session ID when an edge has been come to.

ix. Session Verification

The procedures for taking care of and controlling session ID data must be powerful and prepared to do effectively giving assaults focusing on the substance inside.

- **Session ID Length -** Ensure that the substance of the session ID is of the normal size and sort, and that the nature of the data is checked before preparing. Case in point, be equipped for recognizing larger than usual session ID's that might constitute a support flood sort assault. Furthermore, guarantee that the substance of the session ID does not contain unforeseen data – for instance, if the session ID will be utilized inside of the application's backend database, consideration ought to be taken that the session ID does not contain inserted information strings that might be translated as an augmentation to the "Select" SQL inquiry.
- **Source of the Session ID –** When utilizing the HTTP POST system for correspondence session data, guarantee that the application is fit for recognizing whether the session ID was conveyed to the application from the customer program through the HTTP POST technique, and not through a controlled GET ask. Changing over HTTP

POST into a GET solicitation is a typical technique for directing cross-site scripting assaults and other appropriated beast power assaults.

VI.CONCLUSION

In cloud computing where multi-tenancy, virtualization and outsourcing characteristics make it at risk of compromising security aspects and there is no physical control on data at rest or data in motion, the data can be protected by storing cryptographically and giving the key management to the authorized party. However, finding a trusted party for doing the important task in such an environment is very difficult. In order to solve the problem, the cryptography techniques need to be customized for the cloud environment. Some researchers with a combination of authentication and cryptography have tried to mitigate the abuse of any unreliable parties in the cloud. The identity based authentication and attribute-based authentication are good examples of this category. Others tried to propose a model by encryption and decryption isolation from the storage service in the cloud. Another solution that emphasizes on the key management is deploying a combination of symmetric algorithms for data and asymmetric ones for keeping the keys. One of the best solutions that many of researches are involved in is homomorphic encryption in which all functions are performed on the encrypted data. However, it is too slow in practice, and even no practical model has been seen for it. On the other hand, the client-side encryption, suggested by many researchers, mitigates the advantages of cloud. So, the first problem in cloud-computing was lack of a trade-offs between client-side and server-side encryption. The server-side encryption provides a faster encryption and decryption by utilizing the resources of the cloud but in an insecure third party. The client-side encryption provides almost more secure, but it undermines advantages of the cloud. Thus, it seems that implementing an in-house private cloud as a trusted party which offers encryption as a service can solve the problem.

REFERENCES

[1] Chia-Ming Wu et al, Ruay-Shiung Chang, Hsin-Yu [1] M. Armbrust et aI. Above the Clouds: A Berkeley View of Cloud Computing, technical report. Univ. of California, Berkeley; Feb 2009.

[2] Deepika Singh, Puran Gour, Rajeev Thakur, "User Security in Cloud Using Password Authentication", Journal of Engineering Research and Applications www.ijera.com ISSN: 2248-9622, Vol. 4, Issue 6(Version 5), June 2014, pp.39-44.

[3] Hossein Rahmani, Elankovan Sundararajan, Zulkarnain Md. Ali, Abdullah Mohd Zin," Encryption as a Service (EaaS) as a Solution for Cryptography in Cloud", The 4th International Conference on Electrical Engineering and Informatics (ICEEI 2013), 2212-0173 © 2013 The Authors. Published by Elsevier Ltd.

[4] Haibo Yang and Mary Tate, "Where are we at with Cloud Computing?: A Descriptive Literature Review" 20th Australasian Conference on Information Systems, 2-4 Dec 2009, Melbourne.

[5] Manoj V. Thomas, Anand Dhole, K. Chandrasekaran, " Single Sign-On in Cloud Federation using CloudSim" *I. J. Computer Network and Information Security,* 2015, 6, 50-58, DOI: 10.5815/ijcnis.2015.06.06.

[6] My Abdelkader Youssefi, "Securing Cloud Computing Services Using Strong User Authentication With Local Certification Authority", (IJITR) International Journal Of Innovative Technology And Research Volume No.3, Issue No.6, October - November 2015, 2493 – 2497.

[7] Satheesh K S V A Kavuri, Dr.Gangadhara Rao Kancherla and Dr.Basaveswara Rao Bobba, "Data Authentication and Integrity Verification Techniques for Trusted/Untrusted Cloud Servers", 978-1-4799-3080-7/14/$31.00_c 2014 IEEE.

[8] Manish Kumar Sharma, Rasmeet S. Bali and Arvinder Kaur, "Dyanimc Key based Authentication Scheme for Vehicular Cloud Computing", 978-1-4673-7910-6/15/$31.00_c 2015 IEEE.

[9] Tamal Kanti Chakraborty, Anil Dhami, Prakhar Bansal and Tripti Singh, "Enhanced Public Auditability & Secure Data Storage in Cloud Computing", 978-1-4673-4529-3/12/$31.00_c 2012 IEEE.

[10] Ms.Vishnupriya.S, Ms.Saranya.P and Ms.Rajasri.A, " Secure Multicloud Storage with Policy based access control and Cooperative Provable Data Possession", ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE.