SECURE SESSION MECHANISM IN CLOUD PLATFORM

MR. SATVIK DHANDHANIA AND PROF. GOVINDA K
VIT University,SCSE,Vellore,India
satvik.dhandhania2011@vit.ac.in; kgovinda@vit.ac.in

ABSTRACT: Cloud computing provides people the way to share distributed resources and services that belong to different organizations or sites. Since cloud computing shares distributed resources via the network in an open environment, it makes security problems a major burden while developing and deploying cloud computing applications. This paper proposes a model which will make use of the oPass, a user authentication protocol resistant to password stealing and password reuse attacks, to register and authenticate a system in an organization followed by deployment of a key generator algorithm sent to the client system by syncing a plugin that will keep generating keys in regular time intervals. The same algorithm will be running on the cloud server and the keys generated on the server will be compared to the client's keys at regular time intervals. All systems of the organization will have their respective algorithms and thus the cloud server will be able to identify authenticated systems. A method has been proposed to build a trusted computing environment for cloud computing system by integrating the trusted computing platform into cloud computing system along with oPass. In this model, some important security services, including authentication, authorization, confidentiality and integrity, are provided in cloud computing system.

KEYWORDS: Access control, authorization, clouds, session, security, trusted platform.

INTRODUCTION

Cloud computing has revolutionized the way companies are implementing their information systems. This computing model offers a unique structure in the way to utilize computing resources to business and individual users from a third party company as an alternative to their own computing infrastructure which turns out to be far more expensive [1, 2]. The customers are provided leverage by using cheap hardware resources for which they have to pay on the basis of usage along with the reduced requirement of managing the tasks for maintaining the cloud. Cloud computing provides users the illusion of unlimited computing resources to a user. The user can utilize computing resources of any scale regardless of the concern for the maintenance and provision of these resources [3, 4]. Thus they only tend to use the infrastructure without caring about the management and maintenance of these resources since most of it is based on virtual machines running on the cloud server. The growing popularity of cloud computing brings forth security challenges, which are particularly exacerbated due to resource sharing [26]. Cloud computing's multi-tenancy and virtualization features pose unique security and access control challenges due to sharing of physical resources between untrusted tenants, which might lead turn to an accretion of side-channel attacks[27]. Again, multi-tenancy computation can result in unapproved information flow. Heterogeneity of services in cloud computing environments demands varying degrees of granularity in access control mechanisms. Therefore, an inadequate or unreliable authorization mechanism can threateningly increase the risk of unauthorized use of cloud resources and services. In addition to preventing such attacks, a structured authorization mechanism can assist in implementing standard security measures. Such access control challenges and the complications linked with their administration ask for complex security architecture.

In the history of cloud computing there have been many data disclosures, either planned or inadvertent. This unveils the risks of privacy and confidentiality of the cloud data storage deployment. The first ever kind of the risk is the inadvertent disclosure of data which happens because of the errors in the design of the cloud computing software of the providers or due to missing out on trivial bugs. One example of unintentional disclosure of data where non-authenticated users could view the documents by Google Docs was due to a bug [5], whereas the Flicker and Facebook have also leaked the private pictures of the users due to various security flaws [6]. Security is therefore a major factor for any cloud computing infrastructure, because it is necessary to ensure that only authorized access is permitted and secure behavior is accepted.

Cloud computing data centres have a central server administration system, which manages all the operations done by that data centre. Cloud computing provides centralized storage, processing memory, and bandwidth. Due the centralization of computing resources and access through internet, it turns to be eye-catching targets for insider or outsider attackers. The cloud service provider's record of guarding data has been unsatisfactory till now. Recently, Apple's cloud server was hacked to gain access to celebrities' personal pictures. Also, Twitter made an agreement with Federal Trade commission of the United States due to its slovenly security practices which allowed the attackers to subterfuge as any authorize user of the system in 2011 [7]. In addition, several sites have met happenings of security breaches which results in the data loss of users which included email addresses and even credit card numbers [8]. In the modern era where almost each and every computer user accesses the internet to perform various day to day activities like bill payments, online shopping and email services, a breach of security would clearly undermine people promoting the use of the cloud computing as a paradigm. "Fig. 1" demonstrates the current mechanism that takes place during cloud services.

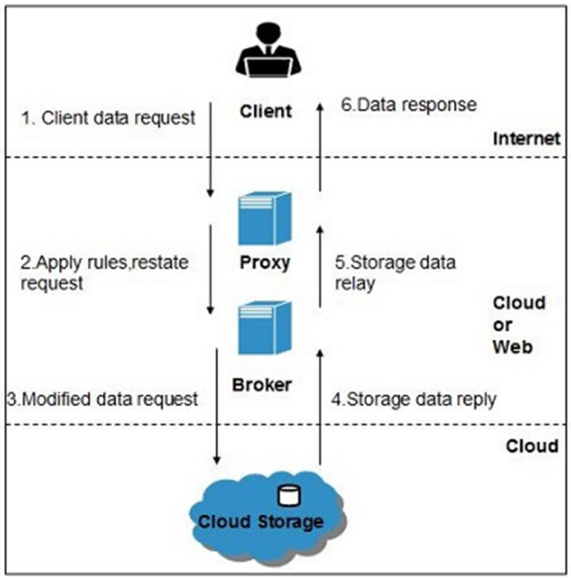

Fig. 1 Overview of the System

Again cloud data service providers encounter huge amount of pressure from government agencies around the world to reveal the private data of the users as per their need. Such as, Google Inc. complies with most of the requests it receives to give the private data of its clients [9]. Additionally, government agencies of several countries have threatened to block many companies' services if they are not given the right to monitor the user's private data [10].
Sometimes cloud service providers also indulge to divulge users private data for money making schemes and incentives from various parties whose business depend on such users private data thereby giving the users the illusion of data being private. Google and Facebook are two of the service providers which have undermined their policy and default settings of privacy in order to endorse new products and services. Thus all user behaviour is being tracked. No wonder when people open up their browsers they receive advertisements of products they might have shown interest some time back at some other site. Moreover, if a provider of cloud service still keeps its promise still the data remains at risk [11, 12]. Users have a shown strong concern over the data confidentiality, security and unauthorized access [13]. This problem turns out to be a menace in case of cloud computing as users have no knowledge about the physical location of their data nor do they have any control over the data centre. Further, the country the user's data resides in may have other security policies compared to the resident of the user which undermines cloud security to a great extent. A mischievous data service provider may also damage users' data by updating, transforming, or falsifying segments of the data.
Thus trust in the cloud computing platform is mainly dependent on the security provided by the service provider. It is an unquestionable fact that if the system is secure, then it will also be trustworthy [11, 14, 15]. This paper proposes a model which will make use of the oPass, a user authentication protocol resistant to password stealing and password reuse attacks, to register and authenticate a system [28]. Further, it would help differentiate between an authenticated system and an unauthorized system in the future.

LITERATURE REVIEW

Amongst various quality-of-service (QoS) metrics, security is one of the greatest concerns to scientists because their data may be intercepted or modified or stolen by mischievous parties during transactions. Therefore the literature survey presents the previous work on securing cloud access and found that numerous systems and methods are incorporated to secure the cloud from different perspectives. Some of the systems tend to create a trusted cloud environment by providing high level of security. Others have made use of other authentication mechanisms which made tasks cumbersome.

A trusted computing environment was proposed by researchers for cloud computing in 2010 [29, 30]. The platform provides the protection of data by implementing a strong authentication mechanism, and the access is restricted by role based access control method in cloud computing system [14]. A multi – clouds database model was presented as an alternative to single cloud environment by the authors in [16]. The purpose of this model was to safeguard the cloud system from the peril of malevolent insider threat and circumvent the failure of the whole cloud services infrastructure. A novel architecture for authenticated key exchange was proposed with the name of cloud computing background key exchange. It utilizes the internet key exchange and randomness reuse approach for key exchange [17]. Trust management is the major worry of research for most of the researchers. The model TFMC introduced a trust management model for cloud computing which is based on the fuzzy set theory [18]. The user can use this model for decision making during the selection of a specific cloud service provider to evaluate the trustworthiness of different

cloud service providers. A unique cost effective and secure model of data distribution is proposed for multi – cloud storage by the authors in [19]. The main idea behind this model is to provide a low cost mechanism of user's data distribution of on available multiple cloud storage providers. Another Storage Architecture has been presented where the private data and public data have been kept separately and the private data is encrypted and access to it requires Multi-factor access modules [31].

The single sign on (SSO) is implemented on the top layer of the cloud computing model. The rationale to this mechanism was to present the user with the best of quality of service including secure storage and availability of data [13, 20]. This method lessens the number of login and increase the security of the overall system.

Privacy preserving and public auditability has been the focus of different research work [21, 22, 23]. The authors proposed a public auditing architecture for cloud computing keeping the privacy preserved [24]. This architecture not only provides the privacy preservation but also support activity like block less verification, public auditability dynamic operation support on data. On the other hand the authors have proved that the architecture presented is insecure due to its incapability to stand against the existential forgery implementing by a known message attack [25]. Authors have discussed about data's origin and categorized the data based on who created it based on the boot ID of the client and hence processing the requests to the cloud server based on provenance of the request in [32].

It is evident from the above discussion that a lot of research has been carried out and still a lot of research work is going on to make cloud computing a secure and trusted technology for the customers. However, several works of this kind are enduring different kind of security issues. Some cannot thwart the illegitimate data access by the cloud service provider while some face the problem of insider malevolent activity. A number of mechanisms are expensive not only in terms of finances, but due to requirement of a time for data processing and the availability of data are affected. So, to avoid these disadvantages the architecture proposed in this paper will allow only the legitimate user to access and store data with confidence. The prime advantage of this scheme is that, it will not only provide security of data at rest (storage at cloud server), but will also provide integrity for various projects in which a colossal organization may be working on.

PROPOSED MECHANISM

The cloud user access mechanism presented here combines the mechanism of oPass user authentication protocol and the trusted computing platform along with a proposed mechanism to validate each users session with the cloud server. This process would provide a vantage to big organizations as a myriad number of systems are present in the organization which access and utilize the cloud resources over time. The stored data of the organization would be kept on systems on the cloud which could only be accessed by systems that pass all the authentications described in this given mechanism. The whole process can be broken down into three processes according to the systems that take part in the process: the server end process, the authenticated client process and the unauthenticated client process.

The server side process

At the cloud server, whenever a new system tries to access cloud storage or resources, the server first obtains the Organization Identification Number (OID) from the user and the MAC address of

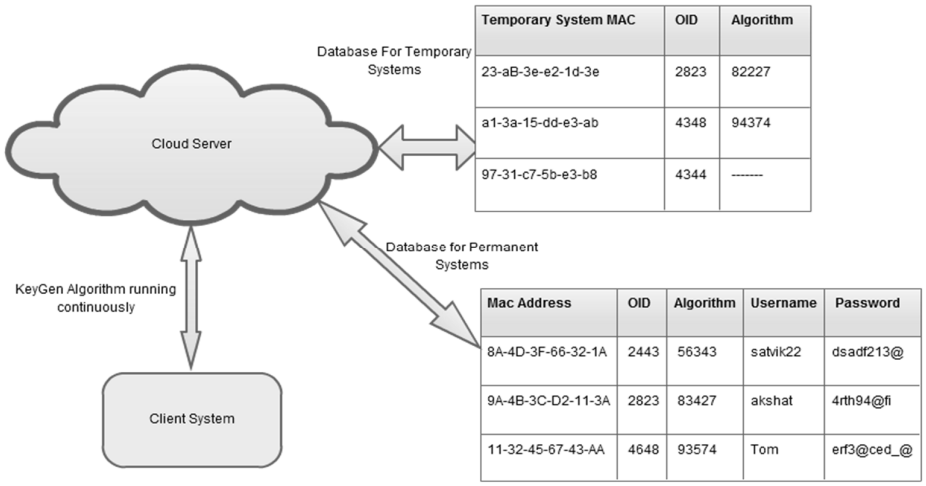

Fig. 2 Server Side Process

the client system. The OID provides the cloud server the details of the organization that is already registered with the cloud service provider to utilize resources of the cloud. Next, the cloud fetches the mobile number that has been registered by the organization for the zone in which the client system resides. Thus large organizations can have multiple OID's according to the area or zone they reside in. Next, the server connects to the telecom service provider and sends a one-time password (OTP) to the registered number. This OTP would only be received by an authorized user. This OTP allows the client to register their system on the cloud server. This OTP would only remain valid for a short interval of time in the order of a few minutes. Thus eavesdropping or using the OTP on another system is annulled unless the authorized mobile number has been compromised. After the server receives this OTP from the client system it first verifies the MAC address to check whether it is coming from the same system or some other system. If it comes from the same system, it stores the MAC address of the client system and along with it chooses a key generator algorithm to allot to the client system for session authentication at regular intervals else it ignores and annuls the OTP. This key generator algorithm selected here is chosen from a pool of key generators stored in the database by cloud service providers. This algorithm is then sent to the client system to be used every time the client tries to access the services of the cloud. This algorithm will be run on the client system as a browser plug in and will generate a new key every five minutes. The same algorithm would be running on the server and hence the same key would be generated at the server end. Thus every five minutes this key would be compared. Any discrepancy would lead to the end of the session. The only way to reset the key generator algorithm would be by getting the OTP again and then choose to sync the algorithm of the client with the server. Thus a new algorithm would be selected by the server and the same would be passed to the client plug in. This set of key generator algorithms could be further populated with time by the service providers to improve the security of the cloud service. Thus only systems whose MAC address is stored in the server database can access the resources (storage) used by the organization. This MAC address along with the key generator algorithm running at the client system validates an authorized user on the cloud server. Thus the OTP would only be used during

registering the system. Using the telecom service provider annuls the man in the middle attack of using the original password [28]. Thus only authorized systems can avail the cloud services.

If however, the user is already registered, then he can log in using the OID, username and password that has been defined for that system access only. Thus the server first verifies OID then receives the MAC address and then according to the MAC address verifies the username and password. Thus any discrepancy in any of the above details would reject the client from accessing the cloud services. "Fig. 2" demonstrates the data stored at the cloud server for authentication purposes. Maximum number of attempts allowed would be three for the username and password before being blocked. If MAC is incorrect it would ask for authentication using the registered mobile number and ask for the sent OTP through the secured telecom channel. After logging into the cloud, the plug-in would start generating keys every five minutes and the same algorithm would run on the cloud, thus there would be a secondary check every five minutes. All man in the middle attacks would be annulled as using these keys it would not be able to identify the next key. Further, any person trying to access the resources in any illegitimate way might replicate the MAC and OID but will not be able to guess the correct key to be used during the five minute period.

The authenticated client side process

An authenticated client already has it MAC address and key generator algorithm defined in the cloud database. Thus when the user logs in using the username and password for the defined MAC address, the plug-in starts generating keys that are send to the server for verification every five minutes. Thus the correct system and plug-in helps in authentication. A third person even after knowing the MAC, OID, username and password would not be able to replicate the key generator every five minutes. Thus any incongruity will result in the system being kicked out from the cloud resources access.

The unauthorized client process

Here two scenarios can turn up. (i) Authenticate an unauthorized client to be an authorized one and (ii) Temporary usage of unauthorized system to access resources.

i. Authenticate an unauthorized client to be an authorized one: "Fig. 3" gives a brief overview of the steps to authorize a system which has been described in detail as follows:

a. The system first request a permanent authentication from the server using a 3G connection by providing the MAC address and OID.
b. The server next fetches the authenticated mobile number for that OID and sends an OTP to the telecom provider to send through SMS channel to the mobile number.
c. This OTP is received by the authorized user and then used on the same system to proceed. This step provides the server with the user MAC address and OTP. Thus the server knows that the same system is trying to access the cloud authentication. No other system can use the OTP generated for any other system. Further, the OTP remains valid only for a few minutes and becomes obsolete after a few minutes.
d. The server now asks the user to create a username and password to register the session and then log in to the cloud environment using this system. The entered username and password is then stored in the server database.
e. Now after this process the cloud server acquires the MAC address and associates it to the user name and password. It also selects an algorithm from a pool of algorithms and then synchronizes the client system plugin with this algorithm. Thus, the server and client run the same algorithm to generate the same key for session verifications.

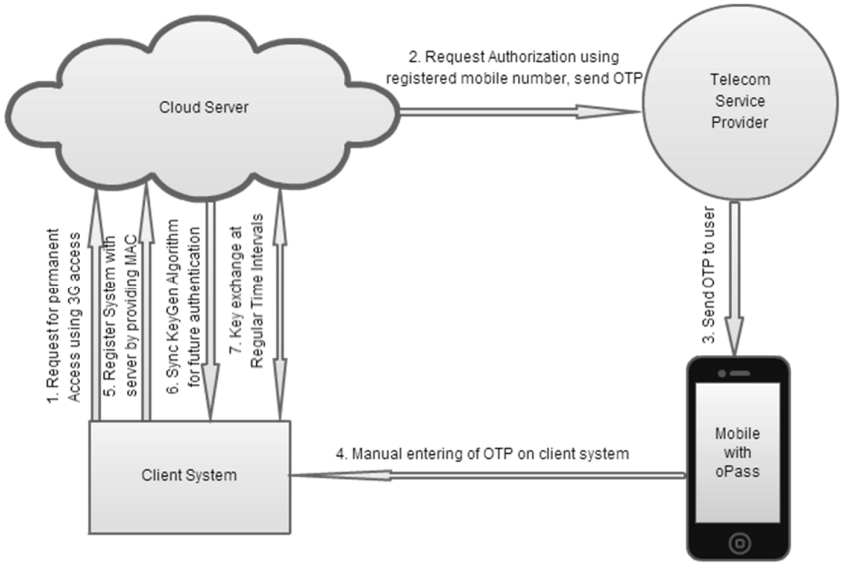

Fig. 3 Authorization Process

f. Next the server stores the Username, Password, MAC Address, Organization ID and Algorithm ID in a secure database and during each login check all the credentials and algorithms first output to allow the user cloud access.

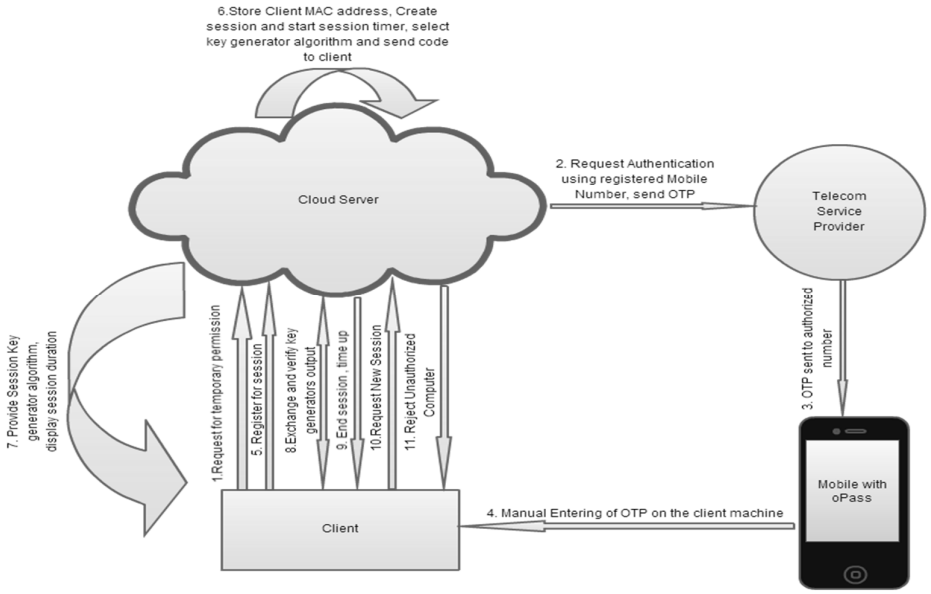

Fig. 4: Unauthorized Kiosk Process

ii. One time use through an untrusted unauthorized kiosk: "Fig. 4" describes in short the procedure of using an untrusted kiosk. Here the following steps are followed to begin operations on an untrusted kiosk:

a. The system first requests a temporary authentication from the server using a 3G connection by providing the MAC address and OID.

b. The server next fetches the authenticated mobile number for that OID and sends an OTP to the telecom provider to send through SMS channel to the mobile number.

c. This OTP is received by the authorized user and then used on the same system to proceed. This step provides the server with the user MAC address and OTP. Thus the server knows that the same system is trying to access the cloud authentication. No other system can use the OTP generated for any other system. Further, the OTP remains valid only for a few minutes and becomes obsolete after a few minutes.

d. Now, the server selects an algorithm from a pool of algorithms and then synchronizes the client system plugin with this algorithm. Thus, the server and client run the same algorithm to generate the same key for this session.

e. Next the server stores the Organization ID, MAC address and algorithm ID in a secure database and allows the user cloud access. All this data is stored in a different database which is different from the database used for storing the permanent user details.

After the session has ended or the session time will be expired the table entry field for algorithm ID will be removed from the server. Thus the next time the system tries to access the cloud, the server would be unable to find the algorithm and hence reject the request. The other system details are stored to keep track of access attempts and other location data. This data may further be used to create patterns of checking the third kiosk accesses where information like location could be used to restrict access only to a city based on the generated statistics. To use the system again the same procedure has to be repeated where if OID and MAC match the field is updated else the MAC is stored with the new OID. Thus it can also be used to check the systems that have accessed multiple organizations cloud access.

Algorithms

The algorithms that would be used for plugins are predefined algorithms and NOT random algorithms where the user who knows the function and will be able to predict all its output. Thus every time both the server and client share the key, the same key would be produced every time. In case of an unauthorized attempt, the generated key would be stored by the server and reused the next time until the authorized system connects. Thus even failed attempts would not make the keys appear incongruous.

These algorithms would be selected from a pool of algorithms where a random number would be generated on the server and using the modulus function it would be mapped to a table with indexes and algorithm ID. Thus using this ID will enable us to be able to call the appropriate algorithm in the server side implementation. Thus dynamic binding of the function would be taking place during runtime. The instance of the algorithm would be stored for each client system separately as there may be multiple clients using the same algorithm. The simple idea is to create a repository of key generators which would be randomly selected for the client system. Thus intruders would be unable to track the exact algorithm which is being used for the system as all systems may have different algorithms. Thus instead of having a complex and heavy computation algorithms running on the client system, simple low computation cost algorithms can be used on the client to generate keys. Thus intruders would be unable to detect the algorithm that has been used in the client

system. Also, the algorithm can be changed anytime by the administrator. Hence, there will always be a dilemma for an intruder. Organizations that would follow a habit of changing and re-syncing the algorithm at regular time intervals would have a much safer environment as intruders in the long run may figure out the key generator used in the system.

Handling Security Issues

i. Security Breach: In case of a security breach, the administrator of the organization can simply reset the algorithm for all the systems in that organization by going to the cloud portal using his mobile's 3G internet connection and enter his authentication that has been provided by the cloud service provider. Thus a new key generator would be provided to the users and it needs to be configured on all the client systems which would require re-registration. Further, while using untrusted computers a temporary separate key generator shall be provided to that system for the session. oPass will provide security for registration operations of users where the permanent password would be transmitted only through the secure 3G channel during configuring the cloud with the mobile number. Thus, after authenticating the mobile only one time passwords (OTP's) will be required to authorize the new systems. Thus the single sign-on would require the OID, username and password to permit a user to make use of the cloud resources and if a security issue occurs then all the systems would have to immediately deploy the new algorithm. This would require the client to request a new algorithm on the cloud server.

ii. Man in the middle attack: A user might intercept a connection to get the MAC address, username, password and OID from an authenticated system. However he will never get the key generator algorithm generator for the client system by intercepting once the system has been authorized. The only way he will get the algorithm is either by hacking into the cloud server to access the database table where all these information is stored or else by compromising the client system.

Further Advantages: Using these mechanism the servers can be locked and be limited to only a few number of clients, segregated on the basis of their OID. For example, an organizations local branch is utilizing cloud services from a particular center then using pattern generators the user's location can be traced and only allow access of cloud resources to systems in the close propinquity of the tracked locations only. Thus attacks from far off locations will also be easily blocked. Similarly, if an organization is utilizing a particular data center then that data center may also be configured to accept only request from only those clients with the same OID that has been configured during the setup phase.

CONCLUSION

With the increase of cloud computing users, the need of having a compact and secure environment is essential to the further expansion of cloud services as a model for the future. The presented mechanism for authentication and authorization would be an effective solution to the security issues faced by cloud. The major privacy and security issues are a big concern for various multinational companies and numerous other start-up's. Thus having this secure model will lure companies to adopt cloud computing as their primary architecture. This mechanism provides an additional security layer using the mobile channel. However, the major drawback to this approach is the time that would be required to authorize a large number of systems. As individual attention and time is required to complete the process, however once the systems are initialized the system tends to be far more efficient than any other system model.

REFERENCES

L. Yousef, M. Butrico and D. Da Silva, “Toward a Unified Ontology of Cloud Computing”, Grid Computing Environments Workshop, (2008), GCE '08, pp. 1 – 10.

S. Ullah and Z. Xuefeng, “Cloud Computing: a Prologue”, International Journal of Advanced Research in Computer and Communication Engineering, vol. 1, no. 1, (2012), pp. 1 – 4.

S. Ullah, Z. Xuefeng, Z. Feng and Zhao Haichun, “TCLOUD: Challenges and Best Practices for Cloud Computing”, International Journal of Engineering Research and Technology, vol. 1, no. 9, (2012), pp. 01-05.

R. L Grossman, “The Case for Cloud Computing”, IT Professional, vol. 11, no. 2, (2009), pp. 23 – 27.

J. Kincaid, “Google privacy blunder shares your docs without permission”, TechCrunch, (2009) March, http://techcrunch.com/2009/03/07/huge-google-privacy-blunder-shares-your-docs-without-permission/.

Flickr, Flickr phantom photos, (2007) February, http://www.flickr.com/help/forum/33657.

U.S. Federal Trade Commission, FTC accepts final settlement with twitter for Failure to safeguard personal information, (2011) March, http://www.ftc.gov/opa/2011/03/twitter.shtm.

E. Mills, “Hackers release credit card, other data from stratfor breach”, CNET News, (2011) December, http://news.cnet.com/8301-27080_3-57350361-245/hackers-release-credit-card-other-data-from-stratfor-breach/.

Google Inc. Transparency Report, https://www.google.com/transparencyreport/userdatarequests/countries/ ?t=table.

M. Reardon, “India threatens to shut down blackberry service”, CNET News, (2010) August, http://news.cnet.com/8301-30686_3-20012981-266.html.

S. Ullah, Z. Xuefeng and Z. Feng, “TCLOUD: Inter – Node Communication Model Based on Social Trust Framework for Cloud Computing”, Advanced Material Research , vol. 717, no. 2, (2013), pp. 688-695.

J. Vijayan, “36 state AGs blast Google’s privacy policy change” Computerworld, (2012) February, http://www.pcadvisor.co.uk/news/mobile-phone/3340102/36-state-ags-blast-googles-privacy-policy-change/.

S. Ullah, Z. Xuefeng and Z. Feng, “TCLOUD: A Multifactor Access Control Framework for Cloud Computing”, International Journal of Security and Its Applications, vol. 7, no. 2, (2013), pp. 15-26.

Z. Shena and Q. Tong, “The Security of Cloud Computing System enabled by Trusted Computing Technology”, 2nd International Conference on Signal Processing Systems, Vol-63 (2010), pp. 11-15.

S. Ullah, Z. Xuefeng and Z. Feng, "TCLOUD: A New Model of Data Storage Providing Public Verifiebility and Dynamic Data Recovery for Cloud Computing", Journal of Software Engineering and Applications, vol. 6, no. 3B, (2013), pp. 23-28.

A. M. Abdullatif, B. Soh and E. Pardede, “MCDB: Using Multi-clouds to Ensure Security in Cloud Computing”, IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC), (2011), pp. 784-791.

E. C. Liu, X. Zhang, J. Chen and C. Yang, "An Authenticated Key Exchange Scheme for Efficient Security Aware Scheduling of Scientific Applications in Cloud Computing”, IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC), (2011), pp. 372 – 379.

X. Sun, G. Chang and F. Li, "A Trust Management Model to Enhance Security of Cloud Computing Environments", International Conference on Networking and Distributed Computing, (2011), pp. 244 – 248.

Y. Singh, F. Kandah and W. Zhang, "A secured cost-effective multi-cloud storage in cloud computing", IEEE Computer Communications Workshops (INFOCOM WKSHPS), (2011), pp. 619 – 624.

R. G. Ashish and D. M. Bhavsar, "Securing user authentication using single sign-on in Cloud Computing", IEEE Nirma University International Conference on Engineering (NUiCONE), (2011), pp. 1-4.

H. Zhuo, S. Zhong and N. Yu, "A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability", IEEE transactions on Knowledge and Data Engineering, vol. 23, no. 9, (2011), pp. 1432-1437.

S. Ullah, Z. Xuefeng and Z. Feng, "TCLOUD: A Reliable Data Storage Architecture for Cloud Computing", Advanced Material Research, vol. 717, no. 2, (2013), pp. 677-687.

W. Qian, C. Wang, J. Li, K. Ren and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing", Computer Security–ESORICS (2009) Issue No.05, pp. 355-370.

W. Cong, K. Ren, W. Lou, and J. Li, "Toward publicly auditable secure cloud data storage services", IEEE Network, vol. 24, no. 4, (2010), pp. 19-24.

X. U. Chun-xiang, H. E. Xiao-hu and A. Daniel, "Cryptanalysis of auditing protocol proposed by Wang, *et al.*, for data storage security in Cloud Computing", (2012) Vol 308, pp 422-428.

H. Takabi, J.B.D. Joshi, and G.-J. Ahn, "Security and Privacy Challenges in Cloud Computing Environments," *IEEE Security & Privacy*, vol. 8, no. 6, 2010, pp. 24-31.

T. Ristenpart et al., "Hey, You, Get off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds," *Proc. 16th ACM Conf. Computer and Communications Security* (CCS 09), ACM, 2009, pp. 199-212.

Hung-Min Sun, Yao-Hsin Chen, and Yue-Hsun Lin, "oPass: A User Authentication Protocol Resistant to Password Stealing and Password Reuse Attacks" *Proc.* IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 7, NO. 2, APRIL 2012.

Zhidong Shen, Qiang Tong, "The Security of Cloud Computing System enabled by Trusted Computing Technology" Proc. *2010 2nd International Conference on Signal Processing Systems (ICSPS) Vol 2, pp 11-15*.

Zhidong Shen, Li Li, Fei Yan, Xiaoping Wu, "Cloud Computing System Based on Trusted Computing Platform", Proc. 2010 International Conference on Intelligent Computation Technology and Automation, Vol-1,pp 942-945 .

Sultan Ullah and Zheng Xuefeng, " TCLOUD: A Trusted Storage Architecture for Cloud Computing", Proc. International Journal of Advanced Science and Technology Vol.63,(2014),pp.65-72, http://dx.doi.org/10.14257/ijast.2014.63.06.

John Lyle and Andrew Martin, "Trusted Computing and Provenance: Better Together" (Oxford University Computing Laboratory)