Enhanced Identity Management System Based Secured Framework in Cloud Authorisation Environment

D.Mukesh Varma[#1], O.Gandhi[*2] , O.Aruna [#3]

[1]*M.Tech Student, Computer Science & Engineering,Narasaraopeta Engineering College, Narasaraopet,Guntur,Ap,India.*

[2,3]*Assistant Professor, Computer Science & Engineering,Narasaraopeta Engineering College, Narasaraopet,Guntur,Ap,India.*

[1] mukeshvarmachowdary@gmail.com

[2] ongolegandhi@gmail.com

[3] Arunasri52@gail.com

Abstract—Cloud computing has been a vital area of focus in recent years. The shift of IT-Market towards Cloud and its products has been a revolution. Cloud is basically an distributed network topology framed where the involved user or customer resides there data on the distributed cloud which raises many security issues to be handled, thus cloud has many weaknesses in cloud data handling and authorisation. Hence Cloud security has been an demand Area for many researchers. This paper presents an Trusted Framework for that the most popular IdM, namely OAuth, working in scope of Mobile Cloud Computing has many weaknesses in authorisation architecture. In particular, authors find two major issues in current IdM. First, if the IdM System is compromised through malicious code, it allows a hacker to get authorisation of all the protected resources hosted on a cloud. Second, all the communication links among client, cloud and IdM carries complete authorisation token, that can allow hacker, through traffic interception at any communication link, an illegitimate access of protected resources. We also suggest a solution to the reported problems, and justify our arguments with experimentation and mathematical modeling

I. INTRODUCTION

As a concept, cloud computing's primary significance lies in allowing the end user to access computation resources through the Internet, as shown in Fig. 1. Some scholars find cloud computing similar to grid computing [13], but some also find similarities to utilities such as water and electrical power and refer to it as utility computing [12]. Because the use of resources can be independently adjusted, it is also sometimes referred to as autonomic computing [15].

Dig: Cloud computing concept map

The literature contains many explanations of cloud computing [16]. After compiling scholarly definitions of cloud computing, Vaquero, Rodero-Merino, Caceres, and Lindner suggested that cloud computing could be defined as the integration of virtual resources according to user requirements, flexibly combining resources including hardware, development platforms and various applications to create services [17]. The special features of cloud computing include the storage of user data in the cloud and the lack of any need for software installation on the client side. As long as the user is able to connect to the Internet, all of the hardware resources in the cloud can be used as client-side infrastructure. Generally speaking, cloud computing applications are demand-driven, providing various services according to user requirements, and service providers charge by metered time, instances of use, or defined period.

Another emerging trend in enterprise computing is the use of smart phone devices. International Data Corporation reports on 33% increment trend on the sale of smart phones during past few years, with a prediction of 32.7% increase in 2013 [1]. Smart phone devices has been advanced greatly, in recent years, so has malicious code [2]. Although, smart phones are advancing in terms of computational power, rapidly replacing Personal Computers (PCs) as first choice of a computing device [2], Nonetheless, their major problem still is that of resource poverty. To cater with this problem, organizations have started providing access to cloud services for their users with smart phone-based clients [3][4]. The location independence and computing power of a cloud joined with the mobility of a smart phone gives the freedom of computing anything anywhere, resulting in a powerful ubiquitous computing model. This power and flexibility is bringing high popularity to what researchers call Mobile Cloud Computing (MCC) [5][6]. ABI Research estimates that MCC will gain a user-base of 240 million by the end of 2015 [7].

Being very convenient and accessible, smart phones are at a higher security risk than competing devices. This risk is mainly because of inherent nature of their application software and communication mechanism [2], as we explain further. First, tiny applications are easy to build by anyone, thus freely available, and hence contain malicious code in several instances. Second, mobile software development life

cycle does not provide any activities ensuring the security, safety and trust. Third, the constrained resources do not allow executing full antivirus software. Fourth, the inherent nature of wireless links available to eavesdropping, and the wider availability of Internet, even out of enterprise perimeter to access enterprise data leaves valuable information asset on risk. Fifth, the mobile users choose relatively simpler passwords that are easy to type with constrained input methods [5]

For all aforementioned reasons, strong authentication mechanisms for MCC are needed to protect privileged organizational data. In general, organizations deploy an IdM for greater access control, both for mobile based and PC-based client. However, our experiment, in this paper, shows that IdM based approach are not as effective for MCC as for conventional setting.

In this paper, we inspect security issues related to the use of smartphone-based clients acquiring cloud services through IdM. In particular, we discuss the problems of authorization for a protected cloud resource in two scenarios. First, when the organization's IdM is compromised through a malicious insider (i.e. malicious code) putting all the protected cloud resources on stake. Second, all the communication links among client, cloud and IdM carries complete authorization token, that can allow hacker, through traffic interception at any communication link, an illegitimate access of protected resources.

We proceed as follows. First, we discuss the background of our study in section II. In section III, we illustrate the scenario of the problem domain. We present the related work and the research methodology in section IV and V respectively. In section VI, we present our proposed solution deduced from the grounded theory and the experiments that we illustrate in section VII. We discuss limitations and future work in section VIII. Finally, we conclude the paper in last section.

II. Background

An identity management system manages the identities of individuals by ensuring their integrity throughout their lifecycle. It also maintains the associated roles, access rights, authorizations, and privileges [8]. Modern IdM's provide extended features like Single Sign-On (SSO) and federated identity management [9]. The federation of identity refers to linking the attributes of a person's identity across multiple services, or even organizations. And, an SSO refers to using one access token across multiple service and/or organizations. Popular examples of such federated identities/SSOs are Microsoft and Google accounts allowing users to use multiple services, sometime across multiple organizations. Fig. 1 illustrates the communication sequence between a user and an IdM. Figure 1 represents the basic functionality of IdM consisting of 8 steps that includes 1) user login to IdM with his username and password, 2) user request to access cloud application/date, 3) cloud ask for token, 4) user request the token from IdM, 5) IdM generates the token and send it to user and cloud, 6) user send the token, received from IdM, to cloud to finalize the process of authentication, 7) cloud compares the token received from user and IdM. On successful comparison, cloud let user access the data or application. The centralized management of identities of an organization's workers provides a solution that seems reliable, secure, and easy to deploy. Industry is adopting this mechanism on a fast pace. The deployment of IdM takes two layers, one for authentication, and another for authorization. Several options exist for deploying these layers, for example OpenID [10], SAML [11] and OAuth [12]. In our work, we embark upon authorization problems associated with the deployment of

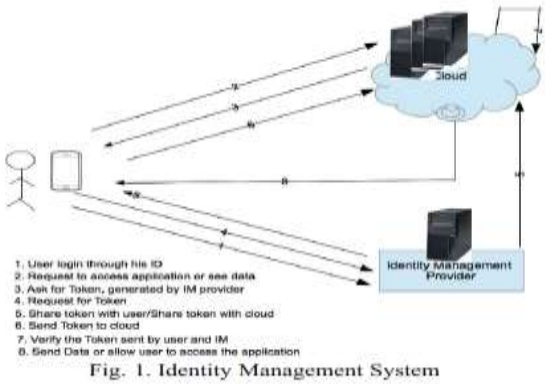

Fig. 1. Identity Management System

III. Problem Domain

We observed the limitations of IdM by looking into steps provided in Figure 1 such as what will happen if IdM is compromised. In step 4 and 5 (Fig. 1), IdMserver generates the token and send it to cloud, and if IdM is compromised then any illegitimate user can use the same token to access the cloud's services/data. This compromise could be occurred due to malicious insider or malicious code. Current IdM, in case of being compromised, put all the cloud's resources on stake.

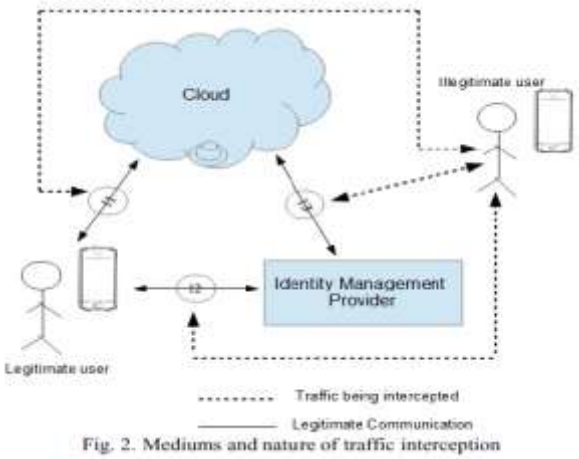

Fig. 2. Mediums and nature of traffic interception

Another problem, considered in this study, is what if an attacker intercepts the network traffic at any communication channel among IdM, cloud and user and gain unauthorized access to token that, further, can provide an unauthorized access to cloud's resources. We have three communication channels in our scenario, 1) between the mobile client and the cloud (marked with 1 in Fig. 2), 2) between the mobile client and the IdM (marked with 2 in Fig. 2), and 3) between the IdM and the cloud (marked with 3 in Fig. 2). Part of these communication links is obviously wireless, vulnerable to eavesdropping with very little effort.

If the hacker intercepts the traffic at Communication channel 1, he can access the token that is sent to cloud by user to access the data. Hacker can use this token to illegitimate access of cloud's resources. The same way, if he intercepts the traffic at communication channel 2 and 3, he will be able to

get illegitimate access to token that is being used by user to access cloud's resources

We assume, in this research, that this information over communication channel is not highly encrypted and can be decrypted with available decrypted algorithms.

For the aforementioned problems, we propose a solution in section VI. We do not work on strengthening the encryption on data link layer, nor do we suggest putting the best antivirus on OAuth server. Instead, we propose a multi-token strategy that strengthens the IdM's authorization architecture within existing structure. It reduces the probability of theft of cloud data and service when IdM is compromised or network links are eavesdropped and tokens are stolen.

IV. RELATED WORK

OAuth [12] and OpenId [10] are two similar solutions that facilitate the idea of identity management systems. The purpose and approach to manage identities are different among these solutions

In OAuth, client obtains a token (string denoting a specific scope and limited lifetime) from authorization server to access a resource, hosted on resource server. For example, end-user(resource owner) can grant printing service (client) access to her protected data, which is stored at data-storage-server (resource server) without sharing her credentials (username/password). OAuth consists of four modules (roles) that includes 1) resource owner (person/server that grant the access of a protected resource), 2) resource server (the server that hosts the protected resource), 3) client (user/application that make request to access resource on behalf of resource owner) and 4) authorization server (the server responsible to issue the token to client). Figure 3 represents the communication flow of OAuth and detail description is as follows [12]:

1.The client requests authorization from the resource owner

2. The client receives an authorization grant (credentials that represents the resource owner's authentication)

3. The client provides authorization grant to authorization server and request for access token
4. The authorization server authenticates the access token and after successful validation provides access token

5. The client request the protected resource from a resource server through by providing access token

6. The server validates the token and on successful validation, grants an access to resource.

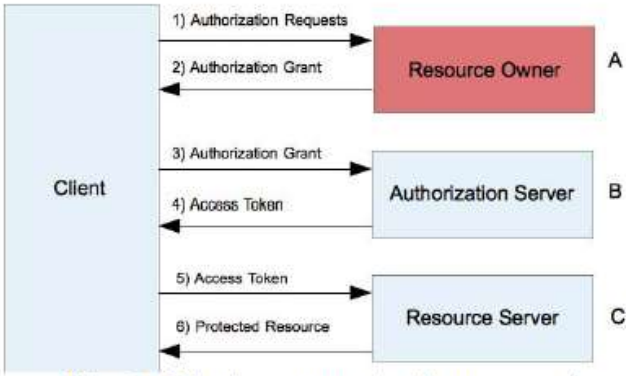

Fig. 3. OAuth communication protocol

OAuth considered, in official specification, a communication among modules (A, B and C in Figure 3) as out of scope. The specifications [12] for OAuth does not discuss the vulnerabilities and possible attacks that can be performed on the communication medium among these modules. There is no discussion on the level of damage that can be caused to system, through intercepting the traffic among these modules. It is very important to secure each medium such as (1, 2, 3, 4, 5 and 6) so no information should be intercepted. Information leakage, at any medium in OAuth, can provide sufficient access to attacker to manipulate the resource/data. The assumption, made in OAuth specification [12], such as "attacker has no access to communication between authorization server and resource owner" reduce the implication of it in a secure system such as banking etc.

In addition to this limitation, Module A is a bottleneck of the system, if it is compromised through malicious insider or malicious code, the whole system would be compromised. The client, can access the information about authorization grant, and can access the token, that further can help to access the resource owner.

In case of smart phone being stolen, OAuth does not provide any mechanism to secure the data except that it encourages users to put key lock on their mobile [12].

Our research is based on authorization of user in the IdM and this is why OpenId is out of scope. OpenId is about authentication (providing the evidence who you are) but OAuth is about authorization (granting an access of resource to third party on your own behalf). OpenId helps you login in multiple sites through single sign-on. Studies such as [13][14][15] has proven that OpenId has many security weaknesses and vulnerable to malicious code attack. These studies discussed an attack, performed by attacker on server that uses OpenId, to install malicious code. This code forwarded the user to bogus identity provider authentication page and asked for his credentials. Later, attacker through malicious code used this credentials to access the user data on original server[14][15]. Many practitioners are promoting the use of OpenId with OAuth for better security. We observed that this combination of OAuth with OpenId could be lethal to user's private data. For example, in case of authorization server being compromised, OpenId (service for single sign-on) and OAuth (authorizing the person with single sign-on)

could be an advantage to an attacker to access all resources/data of user on multiple sites.

Other than these two similar systems, there are many case studies that use IdM such as Xiao et al. in [16] mentioned current security mechanisms in mobile cloud computing as insufficient because if attacker is capable of faking/stealing user's credentials than the cloud data is on stake. Author in this study provides the algorithm to generate dynamic identities to provide secure mechanism to protect cloud data. This algorithm performs well if adequate security measurements are implemented at server level such as antivirus, network firewalling and intrusion detection systems. This algorithm has of no use, if the system is compromised, because whatever efficient key is generated through algorithm, attacker would get access to it. Leandro et al. in [9] promoted the use of Shibboleth (mechanism to control access) as access control system, in cloud computing, without the use of trusted third party e.g. IdM server. It provides strong authorization but does not provide strong authentication for example, once the user is authenticated, it does not provide a mechanism to ensure the legitimacy of the person connected with system whether a user is legitimate or an attacker.

In simple words, an illegitimate user holding valid username and password can access the cloud services without being verified. Shibboleth does not guarantee 100 % secure transaction. In order to deal with user verification, Angin et al. in [17] proposed a solution called 'active bundle scheme' for IdM with comparison of application-centric approach. This approach allows server to keep track of user in order to authenticate in such a way that does not reveal its actual identity and to protect personally identifiable information from unauthorized access. Authors in [9] discussed the similar concept except that Angin et al. do not implement or validate the solution. There are many articles such as [18][19][20][21][2] that provides IdM, with respect to PC, and modify it in order to secure user's data on cloud but we found no study, during our literature study, to implement IdM on mobile computing. We also observed that every article is modifying IdM just to protect user's identity, no one has explored it in the scenario where IdM server is compromised and network traffic is intercepted.

V. RESEARCH METHODOLOGY

Primarily, we use 2 research methods in this work. First, we conduct extensive literature survey (state-of-art). Based on the knowledge extracted from state-of-art and through empirical analysis, we trace problems in the current authorization architecture. The solutions to the problems—in the form of modified authorization architecture—are based on grounded theory established by extensive literature review of related work. To justify our solutions, we conduct experiments (stateof- practice) and do mathematical modeling in two scenarios, one for current practice in IdM and the other for suggested solutions. Finally, we compare the results of the two scenarios and show that our proposed solution provides better security.

VI. PROPOSED SOLUTION

Our experiments show that if IdM server is compromised, the attacker gets access to authorization token generated by IdM, resulting in an illegitimate access the protected resource on the cloud.

In our solution, we propose generating a distributed authorization token composed of two parts for a single resource access. First token is generated by IdM—upon producing credentials by the user—sent to the user and the cloud, as currently in practice. The second token is generated by the cloud—upon producing credentials by the user—and sent to the user. The cloud also saves this token for future use. The sequence of action is as follows.

1. The user logs in to the cloud.
2. The cloud generates a token, sends it to the user and saves it as well for future reference. The cloud also requests the user to produce the token generated by IdM.
3. The user logs in to IdM.
4. The IdM generates the token and sends it to both the user and the cloud.
5. User sends both tokens—one from cloud and the other from IdM—to the cloud to request access.
6. The cloud compares the token sent by user with the tokens saved in its database.
7. The access is granted/denied on the basis of comparison results

In this scenario, the cloud and the user possess two tokens, while the IdM server has access to a single token generated by it. It leaves a malicious insider planted into the IdM with access to insufficient information to acquire protected resources on the cloud. Fig. 4. Illustrates the communication architecture and sequence in our proposed solution.

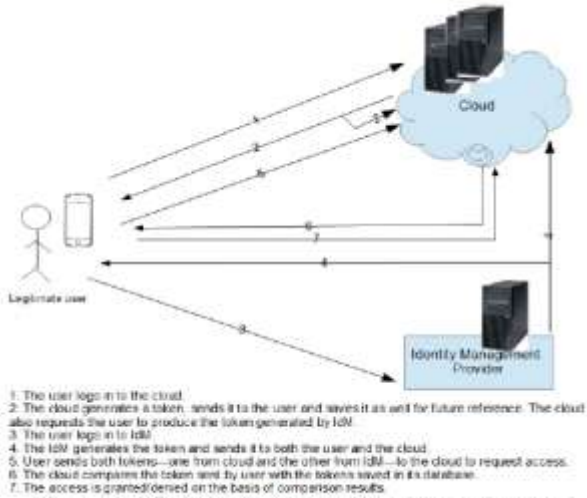

Fig. 4. Modified Identity Management System (Proposed Solution)

The client is required to login to the IdM as well as the cloud. Both servers respond with a token. Consequently, the user possesses two tokens, which are later represented to cloud to access the protected resource.

The other problem we report with the current authorization architecture is due to the insecurity of communication links, as discussed in section III. We assume that a hacker passively eavesdropping on communication links is able to read the communication, and strip off any security mechanisms applied, resulting in the recovery of the original token. She can then acquire the protected resource by presenting the token to the cloud. However, our proposed scenario limits the opportunity of a hacker to read sufficient information to acquire the protected resource.

In this scenario, if a hacker is eavesdropping on channel 2 or channel 3, as depicted in Fig. 2., she has access to only one token sent by the IdM to the user or the cloud. She is not able to access the full information to acquire the protected resource. This significantly reduces the probability of hacking the required amount of information to acquire the protected resource. We analyze this situation in the following.

Let t denotes a token and h(t) denotes the probability a token is hacked over the network, and the total probability of hacking a token is denoted by p. Also let that denotes the link between the client and the cloud, denotes the link between the user and the IdM, and denotes the link between the cloud and the IdM, then in the current IdM scenario

h(t)=h(t over)+ h(t over)+ h(t over) (1)

We also assume that the probability of hacking the token over all links is equally likely—because we do not take network security measures in account—, i.e.

h(t over)= h(t over)= h(t over)= (2)

By (1) and (2)

h(t)=p

Contrary to the above scenario, our proposed solution distributes the complete authorization token into two independent tokens. Links and carry only partial token. The complete distributed token is available only at link Thus, a hacker with access to either or cannot access the complete information to acquire the protected resource, resulting in a decreased probability of hacking the complete token. Therefore the probability for h(t) is now

h(t)=h(t over)=

This reduces the probability of hacking the token by a significant factor of 2/3. This section provides the discussion on our solution with respect to each problem, discussed in section III.

In this proposed solution we taken two most secure algorithms for encryption, decryption. That isMD-5 With RSA Algorithm. The two security approach make our framework more secure in comparison to the previous .In today's era the demand of cloud is increasing, so the security of the cloud and the user is on the top concern.

1.RSA algorithm

RSA is basically an asymmetric encryption /decryption algorithm.The RSA algorithm can be used for both public key encryption and digital signatures. Its security is based on the difficulty of factoring large integers.

Algorithm for RSA encryption

I. Person A transmits his/her public key (modulus n and exponent e) to Person B, keeping his/her private key secret.

II. When Person B wishes to send the message "M" to Person A, he first converts M to an integer such that 0 < m < n by using agreed upon reversible protocol known as a padding scheme.

III. Person B computes, with Person A's public key information, the ciphertext c corresponding to

$c \equiv m^e \pmod n$.

IV. Person B now sends message "M" in ciphertext, or c, to Person A.

Algorithm for RSA decryption

I. Person A recovers m from c by using his/her private key exponent, d, by the computation

$m \equiv c^d \pmod n$.

II. Given m, Person A can recover the original message "M" by reversing the padding scheme.

This procedure works since

$c \equiv m^e \pmod n$,
$c^d \equiv (m^e)^d \pmod n$,
$c^d \equiv m^{de} \pmod n$.

By the symmetry property of mods we have that

$m^{de} \equiv m^{de} \pmod n$.

Since $de = 1 + k\phi(n)$, we can write

$m^{de} \equiv m^{1 + k\phi(n)} \pmod n$,
$m^{de} \equiv m(m^k)^{\phi(n)} \pmod n$,
$m^{de} \equiv m \pmod n$.

From Euler's Theorem and the Chinese Remainder Theorem, we can show that this is true for all m and the original message

$c^d \equiv m \pmod n$, is obtained

2.MD5 algorithm

Message digest Algorithm5 functions also called hash functions, are used to produce digital summaries of information called message digests. Message digests (also called hashes) are commonly 128 bits to 160 bits in length and provide a digital identifier for each digital file or document.Message digest functions are mathematical functions that process information to produce a different message digest for each unique document. Identical documents have the same message digest.

- Step 1. Append Padding Bits
- Step 2. Append Length
- Step 3. Initialize MD Buffer
- Step 4. Process Message in 16-Word Blocks
- Step 5. Output

Our proposed algorithms is helpful for the today's requirement. In future we can provide several comparisons with our approach with result to show the effectiveness of our proposed framework.

VII. CONCLUSION

In the recent, third parties IdM's are introduced to manage digital identities and access control of the protected cloud

resources an organization owns. The idea is similar to outsourcing the part of a project to some third party. Such systems are becoming very popular and commonly deployed in the organization especially for MCC clients. For MCC users facing mobile device's difficult input methods, IdM's popularity depends upon the ease of use. For the organization, their popularity is due to the reason that they allow organizations to use robust digital identity management systems without having a need to deploy one such system in their premises. However, research indicates some serious flaws into their access control model, like that of stealing the authorization tokens through a malicious insider or over a network link. We have worked our ways to identify those flaws empirically and with experimentation. As a solution, we propose some modifications—supported by experimentation— to the original access control model. Primarily, we focus on distributing the authorization token generation between the IdM and the cloud. Through our experiments and analysis, we show that the possibility of hacking a token drops by a significant factor, resulting in increased security for the protected resource over the cloud.

For cloud computing to spread, users must have a high level of trust in the methods by which service providers protect their data. This study proposes A New Secured Business Framework in the Cloud Environment, emphasizing that authorization for the storage and encryption/decryption of user data must be vested with two different service providers. The privileges of Storage as Service provider include storing user data which has already been encrypted through an Encryption/Decryption Service System, but does not allow this service provider access to the Decryption Key or allow for the storage of decrypted data. Furthermore, the privileges of the Encryption/Decryption as Service provider includes management of the key required for the encryption/decryption of user data, but not the storage of decrypted or encrypted user data. In this new business model, user data in the Storage Service System is all saved encrypted. Without the decryption key, there is no way for the service provider to access the user data. Within the
Encryption/Decryption Service System there is no stored user data, thus eliminating the possibility that user data might be improperly disclosed.

REFERENCES

[1] N. Hawthorn, "Finding security in the cloud," Computer Fraud & Security, vol. 2009, issue 10, pp. 19-20, October 2009.

[2] A. Parakh and S. Kak, "Online data storage using implicit security", Information Sciences, vol. 179, issue 19, pp. 3323-3333, September 2009.

[3] A. Juels and J. Burton S. Kaliski, "PORs: Proofs of Retrievability for Large Files," *Proc. of CCS '07*, pp. 584–597, 2007.

[4] H. Shacham and B. Waters, "Compact Proofs of Retrievability," *Proc. of Asiacrypt'08*, Dec. 2008.

[5] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of Retrievability: Theory and Implementation," Cryptology ePrint Archive, Report 2008/175, 2008, http://eprint.iacr.org/.

[6] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," *Proc. Of CCS '07*, pp. 598–609, 2007.

[7] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," *Proc. of SecureComm'08*, pp. 1–10, 2008.

[8] T. S. J. Schwarz and E. L. Miller, "Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage," *Proc. of ICDCS '06*, pp. 12–12, 2006.

[9] M. Lillibridge, S. Elnikety, A. Birrell, M. Burrows, and M. Isard, "A Cooperative Internet Backup Scheme," *Proc. of the 2003 USENIX Annual Technical Conference (General Track)*, pp. 29–41, 2003.

[10] K. D. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage," Cryptology ePrint Archive, Report 2008/489, 2008, http://eprint.iacr.org/.

[11] B. R. Kandukuri, V, R. Paturi and A. Rakshit, "Cloud security issues," in Proceedings of the 2009 IEEE International Conference on Services Computing, pp. 517-520, September 2009.

[12] Hongxin Hu; Gail-Joon Ahn; Kulkarni, K. Dependable and Secure Computing, IEEE Transactions on Volume: 9, Issue: 3, 2012, pp. 318-331.

[13] M. Baker, R. Buyya, and D. Laforenza, "Grids and grid technologies for wide-area distributed computing," International Journal of Software: Practice and Experience, vol.32, pp. 1437-1466, 2002.

[14] A. Elgohary, T. S. Sobh, and M. Zaki, "Design of an enhancement for SSL/TLS protocols," Computers & Security, vol. 25, no. 4, pp. 297-306, June 2006.

[15] R. Sterritt, "Autonomic computing," Innovations in Systems and Software Engineering, vol. 1, no. 1, Springer, pp. 79-88. 2005.

[16] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility," Future Generation Computer Systems, vol. 25, issue 6, pp. 599-616, June 2008.

[17] L. M. Vaquero, L. Rodero-Merino,J. Caceres, and M. Lindner, "A break in the clouds: towards a cloud definition," ACM SIGCOMM Computer Communication Review, vol. 39, no. 1, pp. 50-55, January 2009.

[18] C. Weinhardt, A. Anandasivam, B. Blau, N. Borissov, T. Meinl, W. Michalk, and J. Stößer, "Cloud computing – a classification, business models, and research directions,"

Selected Paper from International Conference on Computing (NECICC-2k15)