A Study on Security Protocols and Mechanisms in Cloud Computing

Chandramouli Reddy
Research Scholar
Jain University
Bangalore, India

Dr. Suchithra R
Jain Global Campus,
Jain University
Bangalore, India

***Abstract*—Cloud computing paradigm is a service oriented system that delivers services to the customer at low cost. Cloud computing needs to address three main security issues: confidentiality, integrity and availability. In this paper, we propose user identity management protocol for cloud computing customers and cloud service providers. This protocol will authenticate and authorize customers/providers in other to achieve global security networks. The protocol will be developed to achieve the set global security objectives in cloud computing environments. Confidentiality, integrity and availability are the key challenges of web services' or utility providers. System vulnerability is critical to the cloud computing facilities; the proposed protocol will address this as part of measures to secure data at all levels. The protocol will protect customers/cloud service providers' infrastructure by preventing unauthorized users to gain access to the service/facility. This paper explains about the security protocols and mechanisms that exist in cloud computing.**

Keywords—Component; Cloud Computing, Confidentiality, Integrity, Encryption, Identity Management, Authentication, Predicate Encryption, Hashing, Repudiation, Auditing.

INTRODUCTION

This paper addresses the security and privacy-related challenges in cloud computing. There are numerous security issues for cloud computing as it encompasses many technologies including networks, databases, operating systems, virtualization, resource scheduling, transaction management, load balancing, concurrency control and memory management. Therefore, security issues for many of these systems and technologies are applicable to cloud computing. For example, the network that interconnects the systems in a cloud has to be secure. Furthermore, virtualization paradigm in cloud computing leads to several security concerns. For example, mapping the virtual machines to the physical machines has to be carried out securely. Data security involves encrypting the data as well as ensuring that appropriate policies are enforced for data sharing. In addition, resource allocation and memory management algorithms have to be secure.

NETWORK SECURITY ATTRIBUTES OF CLOUD COMPUTING

Authentication: is a process by which one entity verifies the identity of another entity. This can be a person or program. The authentication process can be done in three ways; something that user knows such as password or login name, something user has such as personal identification number (PIN) and something user is such as finger print. There is could be a message authentication system that determines the source of message. Another authentication process can be machine-to-machine, which can be client, server and/or mutual authentications. Client authentication involves the server verifying the client's identity, server authentication involves the client verifying the server's identity and mutual authentication involves the client and server verifying each other's identity. In the context of UCS, a web server can be authenticated, so that user can deal with a real website, and not an imitating (disguising) web server. Transport layer socket (TLS) can be used for this process [4].

Authorization: is the process that ensures that a person has the right to access certain resources. Users can not be allowed to access any resources without knowing the attributes of such users. Users can have access rights to resources; but the authority to do something is not within their reach. For example, a user can use ATM card to withdraw money from the ATM machine. Having been authenticated, he cannot withdraw beyond a recommended maximum irrespective of any amount he has in his bank account. Cloud computing uses these access control and authorization to regulate resources usage and minimize fraudulent practices [1].

Auditing: is a process of collecting information about user attempting access to a particular resource, or per-forms actions. The log in system must be able to record all actions performed on that resource. In case there is any problem, the log file can be checked to trace such a user out.

Confidentiality (*privacy*): is an act that keeps private or sensitive information from being disclosed to unauthorized individuals, entities or processes. In cloud computing environment, it is important to maintain transactions' secrecy, because e-payment instruments like visa are involved.

Integrity: is the ability to protect data from being al-tered or destroyed by unauthorized persons or processes during the course of transmission. This is important in cloud computing environment, because the mobile de-vices use air medium and for this reason, data must be well protected.

Availability: is the unhindered accessibility of a service. An online service is available if a user or program can gain access to the pages, data, or services provided by the site when they are needed. This is critical to UCS. Unavailability of a web site may hinder the on-going transactions and it may

lead to loss of money and customers. Technologies such as load balancing hardware and software are aimed at ensuring availability [16].

Non-repudiation: is the ability to limit parties from refuting that a legitimate transaction took place. Since cloud computing transactions involve money, it is important that the customer commits himself by endorsing his/her signature [16, 17]. It is obvious that these attributes may be difficult to achieve, we therefore proposed a "User Identity Management Protocol (U-IDMP). In this case the emphasis is on user attributes provided by the enterprises for cloud service providers to verify such a user.

SECURITY PROTOCOLS

1.User Identity management protocol for cloud computing (U-IDMP)

The development of user identity management protocol intends to answer some questions being asked by stake-holders and developers. Some these questions are on authorization, authentication, encryption, key management identity provisioning etc. We attempt these questions in our architecture and the U-IDM [5] protocol developed. Our approach in solving these challenges that raised some questions is to develop a "user identity management protocol that will involve stage by stage transactions' verification of the customers. Authentication, authorization and accounting (AAA) are considered in developing this protocol. These 3A are crucial to the implementation of any protocol in cloud computing

Environments. Figure 1 describes our U-IDM protocol. The stakeholders work together to achieve successive trans-actions by monitoring the security of their infrastructural networks. The stakeholders are cloud service providers, registry, metering, billing, and customers. Bank and ISPs and others are not left out.

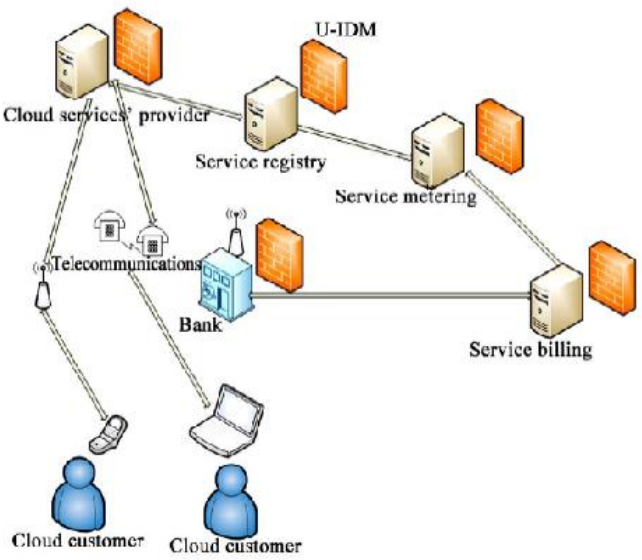

Figure 1. User Identity Management Protocol architecture

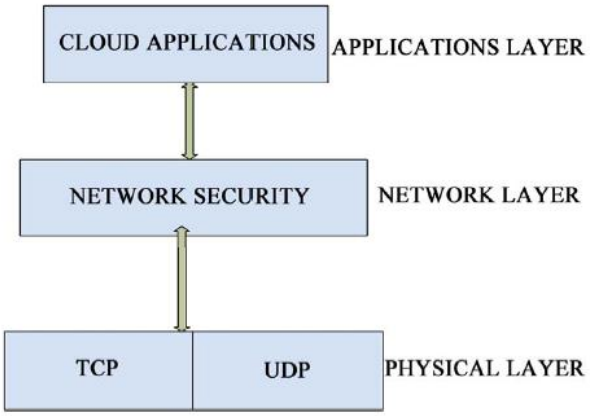

Figure 2 : User Identity Management Protocols

Figure 2 explains the layers developed to support this protocol. The physically layer has provision for time sensitive/insensitive applications by providing TCP/UDP transmissions options. The network layer takes care of different network connections that may be used by the customers i.e. GSM operators and Internet service providers. At this layer, a common protocol is adopted thereby allow effective and good quality of services to the customers. The network operators provide appropriate security measures. This is to prevent system vulnerability, threats and attacks. Here verification take place, hardware and software are authenticated before it is be-ing transmitted to the application layer. Also, integration of other security software is used at this layer. The use DIDS/NIPS, anti-virus and possibly Firewalls at this level is crucial to the successful of services' delivery and quality of service in the cloud computing environments.

II. Predicate Encryption

Predicate encryption is a new encryption paradigm which gives a master secret key owner fine-grained control over access to encrypted data. The master secret key owner can generate secret key tokens corresponding to predicates. An encryption of data x can be evaluated using a secret token corresponding to a predicate f; the user learns whether the data satisfies the predicate, i.e., whether f(x) = 1. Predicate encryption provides a function to search encrypted data and fine-grained access control. That makes a new direction to solve traditional problems. The enhanced functionality and flexibility provided by PE systems are very attractive for many practical applications: network audit logs, sharing of medical records, un-trusted remote storage and so on. More applied research is needed to build predicate encryption into real-world systems. Since PE mechanism originated in theoretical research, considering its high complexity, it is unable to be widely used in the industry. As a result of this, many fascinating open problems remain. An efficient and flexible mechanism PE plays an important role in promoting the popularity of cloud storage. Predicate encryption provides a function to search encrypted data and fine-grained access control. That makes a new direction to solve traditional problems. The enhanced functionality and flexibility provided by PE systems are very attractive for

many practical applications: network audit logs [6], sharing of medical records [7], un-trusted remote storage [8] and so on.

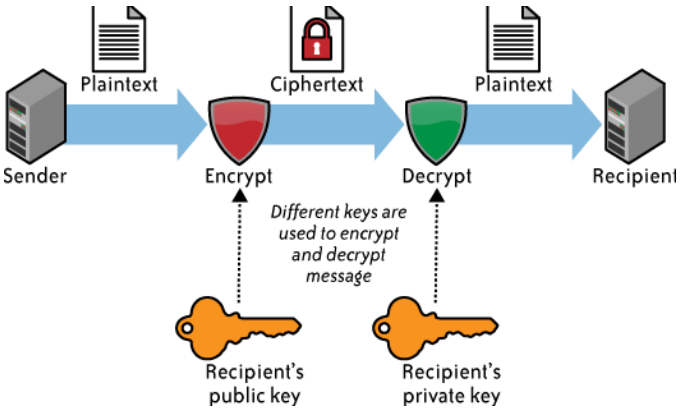

Figure 3: Predicate Encryption

III. Functional Reencyption

Proxy re encryption allows a proxy to convert a cipher text computed under Alice's public key into one that can be opened by Bob's secret key. There are many useful applications of this primitive. For instance, Alice might wish to temporarily forward encrypted email to he r colleague Bob, without giving him her secret key. In this case, Alice the delegator could designate a proxy to re-encrypt her incoming mail into a format that Bob the delegate can decrypt using his own secret key. Clearly, Alice could provide her secret key to the proxy but this requires an impracticable level of trust in the proxy. The primary advantage of PRE scheme is that they are unidirectional (i.e., Alice can delegate to Bob without Bob having to delegate to her) and do not require delegators to reveal all of their secret key to anyone –or even interact with the delegate in order to allow a proxy to re -encrypt their cipher texts .In this schemes, only an inadequate amount of trust is placed in the proxy. For example, it is not able to decrypt the cipher texts it re-encrypts and we prove our schemes secure even when the proxy publishes all the re-encryption information it knows. This enables a number of applications that would not be sensible if the proxy needed to be fully trusted.

IV. Share Authority Based Privacy preserving Authentication Protocol (SAPA)

Shared authority based privacy-preserving authentication protocol (SAPA) is a protocol to address above privacy issue for cloud storage. In the SAPA, 1) shared access authority is achieved by anonymous access request matching mechanism with security and privacy considerations (e.g., authentication, data anonymity, user privacy, and forward security); 2) attribute based access control is adopted to realize that the user can only access its own data fields; 3) proxy re-encryption is applied to provide data sharing among the multiple users. Meanwhile, universal compos ability (UC) model is established to prove that the SAPA theoretically has the design correctness. It indicates that the proposed protocol is attractive for multi-user collaborative cloud applications.

Owner Registration: In this module an owner has to upload its files in a cloud server, he/she should register first. Then only he/she can be able to do it. For that he needs to fill the details in the registration form. These details are maintained in a database.

Owner Login: In this module, any of the above mentioned people have to login, they should login by giving their email id and password.

User Registration: In this module if a user wants to access the data which is stored in a cloud, he/she should register their details first. These details are maintained in a Database.

User Login: If the user is an authorized user, he/she can download the file by using file id which has been stored by data owner when it was uploading.

Access Control: Owner can permit access or deny access for accessing the data. So users can able to access his/her account by the corresponding data owner. If owner does not allow, user can't able to get the data.

Encryption & Decryption: Here we are using this aes_encrypt & aes_decrypt for encryption and decryption. The file we have uploaded which has to be in encrypted form and decrypt it.

File Upload: In this module Owner uploads the file (along with Meta data) into database, with the help of this metadata and its contents; the end user has to download the file. The uploaded file was in encrypted form, only registered user can decrypt it.

File Download: The Authorized users can download the file From cloud database.

Cloud Service Provider Registration: In this module, if a cloud service provider (maintainer of cloud) wants to do some cloud offer, they should register first.

Cloud Service Provider Login: After Cloud provider gets logged in, He/ She can see Cloud provider can view the files uploaded by their clients. Also upload this file into separate Cloud Database.

TTP (Trusted Third Party) Login: In this module TTP has monitors the data owners file by verifying the data owner's file and stored the file in a database .Also TTP checks the CSP(Cloud service provider),and find out whether the CSP is authorized one or not.

V. Dynamic auditing protocol for data storage in cloud computing

Traditionally, owners can check the data integrity based on two-party storage auditing protocols. In cloud storage system, however, it is inappropriate to let either side of cloud service providers or owners conduct such auditing, because none of them could be guaranteed to provide unbiased auditing result. In this situation, third party auditing is a natural choice for the storage auditing in cloud computing. A third party auditor

(auditor) that has expertise and capabilities can do a more efficient work and convince both cloud service providers and owners. For the third party auditing in cloud storage systems, there are several important requirements which have been proposed in some previous works. The auditing protocol should have the following properties: 1) Confidentiality. The auditing protocol should keep owner's data confidential against the auditor. 2) Dynamic Auditing [15]. The auditing protocol should support the dynamic updates of the data in the cloud. 3) Batch Auditing. The auditing protocol should also be able to support the batch auditing for multiple owners and multiple clouds. We consider an auditing system for cloud storage as shown in Fig.4, which involves data owners (owner), the cloud server (server) and the third party auditor (auditor).The owners create the data and host their data in the cloud. The cloud server stores the owners' data and provides the data access to users (data consumers). The auditor is a trusted third party that has expertise and capabilities to provide data storage auditing service for both the owners and servers. The auditor can be a trusted organization managed by the government, which can provide unbiased auditing result for both data owners and cloud servers [16].

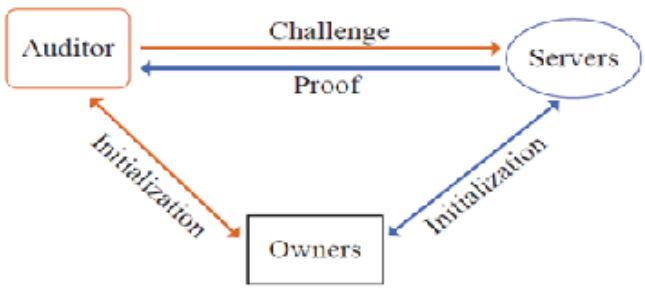

Figure 4: System model of the Data Storage Auditing

VI. *Fully Homomorphic encryption*

Homomorphic Encryption [11] systems are used to perform operations on encrypted data without knowing the private key (without decryption), the client is the only holder of the secret key. When we decrypt the result of any operation, it is the same as if we had carried out the calculation on the raw data. Definition: An encryption is Homomorphic, if: from Enc (a) and Enc (b) it is possible to compute Enc (f (a, b)), where f can be: +, ×, ⊕and without using the private key. Among the Homomorphic encryption we distinguish, according to the operations that allows to assess on raw data, the additive Homomorphic encryption (only additions of the raw data) is the Pailler [9] and Goldwasser-Micalli cryptosystems, and the multiplicative Homomorphic encryption (only products on raw data) is the RSA and El Gamal cryptosystems.

In 1978 Ronald Rivest, Leonard Adleman and Michael Dertouzos suggested for the first time the concept of Homomorphic encryption. Since then, little progress has been made for 30 years. The encryption system of Shafi Goldwasser and Silvio Micali was proposed in 1982 was a provable security encryption scheme which reached a remarkable level of safety, it was an additive Homomorphic encryption, but it can encrypt only a single bit. In the same concept in 1999 Pascal Paillier was also proposed a provable security encryption system that was also an additive Homomorphic [10] Encryption. Few years later, in 2005, Dan Boneh, Eu-Jin Goh and Kobi Nissim invented a system of Provable security encryption, with which we can perform an unlimited number of additions but only one multiplication.

VII. *Proofs of data possession (PDP)*

Provable data possession (PDP) [17] that allows a client that has stored data at an untrusted server to verify that the server possesses the original data without retrieving it. The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces I/O costs. The client maintains a constant amount of metadata to verify the proof. The challenge/response protocol transmits a small, constant amount of data, which minimizes network communication. Thus, the PDP model for remote data checking supports large data sets in widely-distributed storage systems.

A PDP protocol (Fig. 5) checks that an outsourced storage site retains a file, which consists of a collection of n blocks. The client C (data owner) pre-processes the file, generating a piece of metadata that is stored locally, transmits the file to the server S, and may delete its local copy. The server stores the file and responds to challenges issued by the client. Storage at the server is in Ω (n) and storage at the client is in O (1), conforming to our notion of an outsourced storage relationship. As part of pre-processing, the client may alter the file to be stored at the server. The client may expand the file or include additional metadata to be store d at the server. Before deleting its local copy of the file, the client may execute a data possession challenge to make sure the server has successfully stored the file. Clients may encrypt a file prior to out-sourcing the storage. For our purposes, encryption is an orthogonal issue; the "file" may consist of encrypted data and our metadata does not include encryption keys. At a later time, the client issues a challenge to the server to establish that the server has retained the file. The client requests that the server compute a function of the stored file, which it sends back to the client. Using its local metadata, the client verifies the response [18].

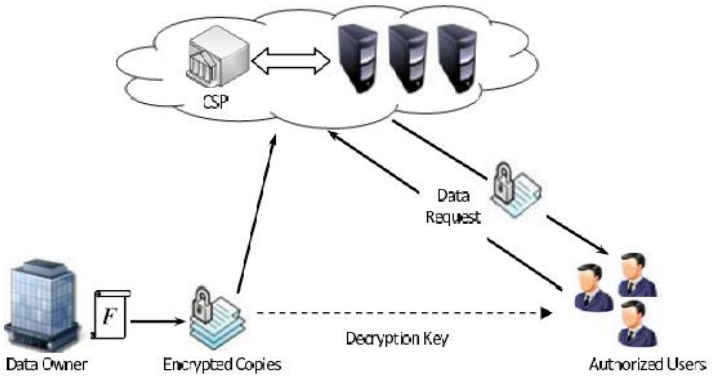

Figure 5: Protocol for provable data possession

VIII. Private Information retrieval (PIR)

A Private Information Retrieval (PIR) [19] protocol allows a database user, or client, to obtain Information from a data-base in a manner that it prevents the database from knowing which data has been accessed or retrieved. The PIR problem is stated as follows: Let the database be modeled as 'n' bit string and user wants to retrieve the ith bit of string 'n', so that 'i' remains unknown to the database. The PIR protocols are divided into two main classes. Computationally Symmetric Private Information Retrieval (cSPIR) [20]: The receiver retrieves an element by him from sender's database, so that the sender has no knowledge about which element was transferred. Computationally-private information retrieval (CPIR): The client retrieves an element from server's 'n' element database of 'l' bit strings, so that server is not aware of which element was retrieved.

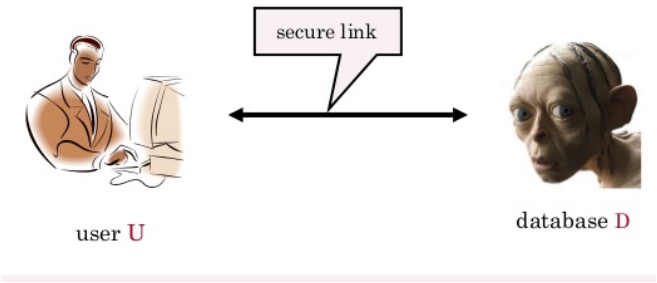

Figure 6: Private Information retrieval

SECURITY MECHANISMS

I. Encryption

Data, by default, is coded in a readable format is known as plaintext. When transmitted over a network, plaintext is vulnerable to unauthorized and potentially malicious access. The encryption mechanism is a digital coding system dedicated to preserving the confidentiality and integrity of data. It is used for encoding plaintext data into a protected and unreadable format. Encryption technology commonly relies on a standardized algorithm called a cipher to transform original plaintext data into encrypted data, referred to as cipher text. Access to cipher text does not divulge the original plaintext data, apart from some forms of metadata, the data is paired with a string of characters called an encryption key, a secret message that is established by and shared among authorized parties. The encryption key is used to decrypt the cipher text back into its original plaintext format.

a) Symmetric Encryption

Symmetric encryption uses the same key for both encryption and decryption, both of which are performed by authorized parties that use the one shared key. Also known as secret key cryptography, messages that are encrypted with a specific key can be decrypted by only the same key. Parties that rightfully decrypt the data are provided with evidence that the original encryption was performed by parties the rightfully possess the key. A basic authentication check is always performed, because only authorized parties that own the key can create messages. This Maintains and verifies data confidentiality.

Symmetric encryption does not have the characteristics of non-repudiation, since determining exactly which party performed the message encryption or decryption is not possible if more than one party is in possession of the key.

b) Asymmetric encryption

Asymmetric encryption relies on the use of two different keys, namely a private key and a public key. With asymmetric encryption (which is also referred to a public key cryptography), the private key is known only to its owner while the public key is commonly decrypted with the corresponding public key. Conversely, a document that was encrypted with a public key can be decrypted only using its private key counterpart. As a result of two different keys being used instead of just the one, asymmetric encryption is almost always computationally slower that symmetric encryption.

The level of security that is achieved is dictated by whether a private key or public key was used to encrypt the plaintext data. As every asymmetrically encrypted message has its own private-public key pair, messages that were encrypted with a private key can be correctly decrypted by any party with the corresponding public key. This method of encryption does not offer any confidentiality protection. However, any party that has the public key can generate the cipher text, meaning this method provides neither message integrity nor authenticity protection due to the communal nature of the public key.

II. Hashing

The hashing [24] mechanism is used when a one-way, non-reversible form of data protection is required. Once hashing has been applied to a message, it is locked and no key is provided for the message to be unlocked. A common application of this mechanism is the storage of passwords. Hashing technology can be used to derive a hashing code or message digest from a message, which is often of a fixed length and smaller than the original message. The message sender can then utilize the hashing mechanism to attach the message digest to the message. The recipient applies the same hash function to the message to verify that the produced message digest is identical to the one that accompanied the message. Any alteration to the original data results in an entirely different message digest and clearly indicates that tampering has occurred [25].

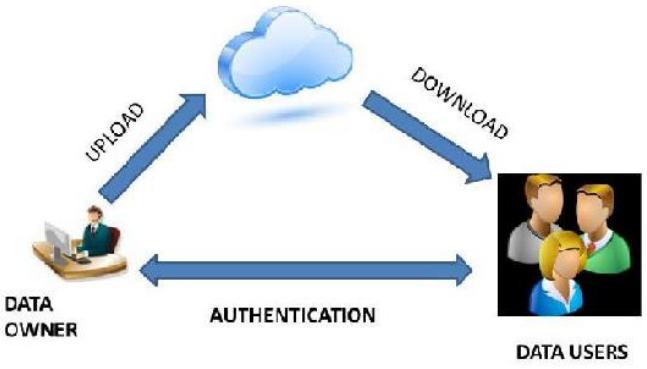

Figure 7: Hashing Mechanism

III. Digital Signature

The digital-signature [12] mechanism is means of providing data authenticity and integrity through authentication and non-repudiation. A message is assigned a digital signature prior to transmission, which is then rendered invalid if the message experiences any subsequent, unauthorized modifications. A digital signature provides evidence that the message received is the same as the one created by its rightful sender. Both hashing and asymmetrical encryption are involved in the creation of a digital signature, which essentially exists as a message digest that was encrypted by a private key and appended to the original message. The recipient verifies the signature validity and uses the corresponding public key to decrypt the digital signature, which produces the message digest. Identical results from the two different processes indicate that the message maintained its integrity. The digital signature mechanism help mitigate the malicious intermediary, insufficient authorization, and overlapping trust boundaries security threats [13].

IV. Public key Infrastructure

A common approach for managing the issuance of asymmetric keys is based on the public key infrastructure (PKI) mechanism, which exists as a system of protocols, data formats, rules, and practices that enable larger-scale systems to securely use public key cryptography. This system is used to associate public keys with their corresponding key owners (known as public key identification) while enabling the verification of key validity. PKIs rely on the use of digital certificates, which are digitally signed data structures such as validity periods. Digital certificates are usually digitally signed by a third party certificate authority (CA).

Other methods of generating digital signatures can be employed, even though the majority of digital certificates are issues by only a handful of trusted CAs like VeriSign and Comodo. Larger organizations, such as Microsoft, can act as their own CA and issue certificates to their clients and the public, since even individual users can generate certificates as long as they have the appropriate software tools. The PKI mechanism is primarily used to counter the insufficient authorization threats

V. Identity and access Management (IAM)

Identity management [21] deals with the identification such as a system in a country. IDM systems are the policies that define which devices are used or which ones are allowed on the network. IDM in cloud has to manage and control virtual devices, service identities, control points etc. IDM has become an important part of cloud these days. Now cloud providers need to control usernames, passwords and other information that is used to identify, authenticate and authorize the users for various applications. Examples: Policy definition, reporting, alerts and alarms. Unauthorized users try to log in the alarm and the alarm goes on. Some systems offer dictionary integration support for both wired and wireless systems.

The Identity and access management (IAM) [22] mechanism encompasses the components and policies necessary to control and track user identities and access privileges for IT resources, environments, and systems. Specifically, IAM mechanisms exist as systems comprised of four main components:

Authentication – Username and password combinations remain the most common forms of user authentication credentials managed by the IAM system, which also can support digital signatures, digital certificates. Biometric hardware (finger print readers), specialized software (such as voice analysis programs), and locking user accounts to registered IP or MAC addresses.

Authorization- The authorization component defines the correct granularity for access controls and oversees the relationships between identities, access control rights, and IT resource availability.

User Management– Related to the administrative capabilities of the system, the user management program is responsible for creating new user identities and access groups, resetting passwords, defining password policies, and managing privileges.

Credential Management – The credential management system establishes identities and access control rules for defined user accounts, which mitigates the threat of insufficient authorization. The IAM [23] mechanism is primarily used to counter the insufficient, denial of service, and overlapping trust boundaries threats.

VI. Single Sign –On (SSO)

Propagating the authentication and authorization information for a cloud service consumer across multiple cloud services can be a challenge, especially if numerous cloud services or cloud based IT resources need to be invoked as part of the same overall runtime activity. The single sign on Mechanism enables one cloud service consumer to be authenticated by a security broker, which established a security context that is persisted while the cloud service consumer would need to re-authenticate itself with every subsequent request.

The SSO [14] mechanism essentially enables mutually independent cloud services and IT resources to generate and circulate runtime authentication and authorization credentials. The credentials initially provided by the cloud service consumer remain valid for the duration of a session, while its security context information is shared. The SSO mechanism's security broker is especially useful when a cloud service consumer needs to access cloud services residing on different clouds.

CONCLUSION

Cloud computing is clearly one of the most enticing technology areas of the current times due, at least in part to its cost-efficiency and flexibility. However, despite the surge in activity and interest, there are significant, persistent concerns about cloud computing that are impeding the momentum and will eventually compromise the vision of cloud computing as a new IT procurement model. Despite the trumpeted business and technical advantages of cloud computing, many potential cloud users have yet to join the cloud, and those major corporations that are cloud users are for the most part putting only their less sensitive data in the cloud. Lack of control is transparency in the cloud implementation – somewhat contrary to the original promise of cloud computing in which cloud implementation is not relevant. Transparency is needed for regulatory reasons and to ease concern over the potential for data breaches. For the enhancement of technology, and hence healthy growth of global economy, it is extremely important to iron out any issues that can cause road-blocks in this new paradigm of computing. In developing solutions to cloud computing security issues it may be helpful to identify the problems and approaches in terms of Loss of control, Lack of trust, Multi-tenancy problem.

REFERENCES

[1] G. Treu, F. Fuchs and C. Dargatz, "Implicit Authorization for Social Location Disclosure," Journal of software,vol.3, No . 1, 2008, pp. 18-26.

[2] M. E. Whiteman and H. J. Mattord, "Principles of Information Security," 2nd Edition, Thomson course Technology, Massachusetts, 2005.

[3] A Cognitive Agents Based P. Venkataram and B. S. Babu, "An Authentication Sche- me for Ubiquitous Commerce: Approach," Proceedings of IEEE Workshops on Network Operations and Management Symposium Workshops, Salvador da Bahia, 7-11 April 2008, pp. 248-256.

[4] pp. 45- A. Gopalakrishnan, "Cloud Computing Identity Manage-ment," *SETLabs Briefings*, Vol. 7, No. 7, 2009, 54.

[5] A Cognitive Agents Based P. Venkataram and B. S. Babu, "An Authentication Sche- me for Ubiquitous Commerce:Approach," Proceedings of IEEE Workshops on Network Operations and Management Symposium Workshops, Salvador da Bahia, 7-11 April 2008, pp. 248-256.

[6] D. Boneh, A. Sahai, and B. Waters, "Functional encryption: Definitions and challenges," in *Proc. 8th Conference on Theory of Cryptography*, Providence, 2011, pp. 253-273.

[7] S. Agrawal, D. M Freeman, and V. Vaikuntanathan, "Functional encryption for inner product predicates from learning with errors," in *Proc. 17th International Conference on the Theory and Application of Cryptology and Information Security*, Seoul, 2011, pp. 21-40.

[8] E. Shi, J. Bethencourt, T. H. H. Chan, D. Song, and A. Perrig, "Multi-dimensional range query over encrypted data," in *Proc. IEEE Symposium on Security and Privacy*, Berkeley, 2007, pp. 350-364.

[9] Pascal Paillier, "Public-key cryptosystems based on composite degree residuosity classes", In 18th Annual Eurocrypt Conference (EUROCRYPT'99), Prague, Czech Republic, volume 1592, 1999.

[10] Craig Gentry, "A Fully Homomorphic Encryption Scheme", 2009.

[11] Sigrun Goluch, "The development of homomorphic cryptography From RSA to Gentry's privacy homomorphism", http://dmg.tuwien.ac.at/drmota/DA_Sigrun%20Goluch_FINAL.pdf, 2011.

[12] Uma Somani, Kanika Lakhani and Manish Mundra "Implementing Digital Signature with RSA Encryption Algorithms to Enhance the Data Security of Cloud in Cloud computing" IEEE 2010

[13] Shobha Rajak, Ashok Verma "Secure Data Storage in the Cloud using Digital Signature Mechanism" IJARCET June 2012.

[14] Jan De Clercq, "Single Sign-on Architectures", Proceedings of Infrastructure Security: International Conference, InfraSec 2002, Bristol, UK, pg 40-58, October 1-3, 2002.

[15] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," IEEE Trans. Parallel Distributed Systems, vol. 22, no. 5, pp. 847-859, May 2011.

[16] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," IACR Cryptology ePrint Archive, vol. 2008, p. 114, 2008.

[17] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song. Provable data possession at untrusted stores. Cryptology ePrint archive, May 2007. Report 2007/202.

[18] M. Bellare and A. Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In Proc. Of CRYPTO '04, Lecture Notes in Computer Science, pages 273–289. Springer, 2004.

[19] C. Devet, I. Goldberg, and N. Heninger. Optimally Robust Private Information Retrieval. In 21st USENIX Security Symposium, 2012.

[20] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In 36th Annual IEEE Symposium on Foundations of Computer Science (FOCS'95), pages 41 {50,oct 1995.

[21] Safiriyu Eludiora, Olatunde Abiona, Ayodeji Oluwatope, Adeniran Oluwaranti, Clement Onime and Lawrence Kehinde," A User Identity Management Protocol for Cloud Computing Paradigm", Published Online, March 2011.

[22] Ardi Benusi, "An Identity Management Survey on Cloud Computing", Int. Journal of Computing and Optimization, Vol. 1, 2014, no. 2, 63-71, Albania.

[23] Anu Gopalakrishanan, "Cloud Computing Identity Management", SETLabs Breifings Vol. 7, No. 7, 2009.

[24] Tsudik, G. "Message Authentication with One-Way Hash Functions", Proceedings, INFOCOM'92, May 1992.

[25] Pardeep Kumar, Vivek Kumar Sehgal, Durg Singh Chauhan, P. K. Gupta, Manoj Diwakar, "Effective Ways of Secure, Private and Trusted Cloud Computing", IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 2, 2011.