

Security Framework of Cloud Data Storage Based on Multi Agent System Architecture: Semantic Literature Review

Amir Mohamed Talib

Faculty of Computer Science & IT, University Putra Malaysia

43400 UPM, Serdang, Selangor, Malaysia

Tel: 60-1-7323-3051 E-mail: ganawa53@yahoo.com

Rodziah Atan, Rusli Abdullah & Masrah Azrifah Azmi Murad

Faculty of Computer Science & IT, Information System Department

University Putra Malaysia, 43400 UPM, Serdang, Selangor, Malaysia

E-mail: (rodziah, rusli & masrah)@fsktm.upm.edu.my

Abstract

The purpose of this literature review is to provide the information about illustrating the usage of Multi-Agent System (MAS) techniques that can be beneficial in cloud computing platform to facilitate security of cloud data storage (CDS) among it. MAS are often distributed and agents have proactive and reactive features which are very useful for cloud data storage security (CDSS). The architecture of the system is formed from a set of agent's communities. This paper of literature review described on the theoretical concept and approach of a security framework as well as a MAS architecture that could be implemented in cloud platform in order to facilitate security of CDS, on how the MAS technology could be utilized in a cloud platform for serving the security that is developed by using a collaborative environment of Java Agent DEvelopment (JADE). In order to facilitate the huge amount of security, our MAS architecture offered eleven security attributes generated from four main security policies of correctness, integrity, confidentiality and availability of users' data in the cloud. This paper of literature review also describes an approach that allows us to build a security cloud platform using MAS architecture and this architecture tends to use specialized autonomous agents for specific security services and allows agents to interact to facilitate security of CDS.

Keywords: Cloud Computing, Cloud Data Storage, Cloud Service Providers, Cloud Data Storage Security, Java Agent Development and Multi-Agent System

1. Introduction

Arguably, the use of agent in security engineering is still in its infancy. To compound matters, security concerns in the MAS domain have not been emphasized in a scale compatible with its widespread academic research. More forcefully put, agent systems themselves are not typically secure.

Security is not a new issue and now it is recognized as one of the most complex problems. Due to the importance of security importance, it has been an issue in an increasingly growing network connectivity, size and implementation of new information technologies (Anderson *et al.*, 2001; McClure *et al.*, 2003). Several attempts have been made to provide security using a software agent systems approach. In these systems, the main focus was on providing a solution for specific security issues, such as authentication and authorization.

Data storage systems are expected to meet several rigorous requirements for maintaining users' data and information, including high availability, reliability, performance, replication and data consistency; but because of the conflicting nature of these requirements, no one system implements all of them together. For example, availability, scalability and data consistency can be regarded as three conflicting goals". Our MAS architecture is proposed to facilitate the correctness, confidentiality, availability, and integrity of user' CDS.

The ultimate challenge in cloud computing is data-level security, and sensitive data is the domain of the enterprise, not the CSP. Security will need to move to the data level so that enterprises can be sure their data is protected wherever it goes. For example, with data-level security, the enterprise can specify that this data is not allowed to go outside of the specific cloud server. It can also force encryption of certain types of data, and permit only specified users to access the data (John *et al.*, 2010). CDSS remains as ongoing problem for the local data center moving the data to the cloud just makes security more difficult. Many CSPs provide rudimentary security

for data stored but none seem to have integrated strong authentication and encryption services that might provide true CDSS.

Our security framework has been built by using two layers: agent layer and cloud data storage layer. Our MAS architecture has five agents User Interface Agent, User Agent, DER Agent, Data Retrieval Agent and Data Distribution Preparation Agent.

There are many cloud computing and CSPs, such as Google, IBM, Amazon, Sun Microsystems, HDS, Microsoft, Nirvanix, EMC, NetApp, HP, Symantec, etc.

There are also more and more CDS platforms, e.g., Amazon S, HDFS, Sun Network.com, CloudNAS, Data ONTAP, SkyDrive, FileStore, EMC Atoms, HP Upline, Hitachi Content Platform, GFS, KFS, Open Source Cloud Computing Environment (OSCC) in University Putra Malaysia UPM (in which our security framework could be test and validate), etc.

2. Structure of Literature Review: Scope of the Review

The literature has been drawn from a variety of sources and is not limited to research and theoretical development over the past few years, though recent works are emphasized. This paper provides a summary and review of the main features found in the literature relating to the research questions outlined in Chapter 1. In keeping with these questions, the structure of this review outline as illustrated in Figure 1(Note 1).

The first part is highlighting the literature review of CDS, CDSS and cloud application platforms. As well as give a differentiation between some of cloud platforms tools.

The second part is highlighting the literature review of MAS and its approaches, MASs designed by Prometheus Methodology and implemented by Java Agent DEvelopment (JADE) and also highlighting the procedures that MAS solving the problems and how MAS secure the CDS.

The third part highlighting the CDS policies and its attributes investigated in cloud computing environment. There is not yet a global standard specification and general architecture to CDSS. The part highlights the requirements of CDSS, propose the architecture.

3. Search Strategy

Half of the literature for this review has been found on the internet via the search engine ‘Google’ that has enabled many recent studies to be sourced. These online sources include published or unpublished papers, working papers, conference proceedings, dissertations, reviews and surveys related to main terms of our research.

4. Defining the Terms

4.1 Cloud Computing

Cloud computing is a term used to describe both a platform and type of application. A cloud computing platform dynamically provisions, configures, reconfigures, and de provisions servers as needed. Servers in the cloud can be physical machines or virtual machines. Advanced clouds typically include other computing resources such as storage area networks (SANs), network equipment, firewall and other security devices. Cloud computing also describes applications that are extended to be accessible through the Internet. These cloud applications use large data centers and powerful servers that host Web applications and Web services. Anyone with a suitable Internet connection and a standard browser can access a cloud application.

4.2 Multi-Agent System

MASs are techniques in the artificial intelligence area focusing on the system where several agents communicate with each other. In (Durfee *et al.*, 1989), MAS is defined as “a loosely coupled network of problem-solver entities that work together to find answers to problems that are beyond the individual capabilities or knowledge of each entity”.

4.3 Cloud Data Storage

CDS is composed of thousands of cloud storage devices clustered by network, distributed file systems and other storage middleware to provide cloud storage service for users. The typical structure of CDS includes storage resource pool, distributed file system, service level agreements (SLAs), and service interfaces, etc. Globally, they can be divided by physical and logical functions boundaries and relationships to provide more compatibilities and interactions. CDS is tending to combined with CDSS, which will provide more robust security.

CDS can provide cloud storage resources for all kinds of clients, and the fee can be based on CDS capacity or CDS bandwidth periodically. The data life cycle management in CDS can be based on servers' configurations, or based on the contracts between servers and clients when CDS services are initiated. CDS is also enables new application types through SOA, Web services APIs and unified service interface via virtualization over a network at low cost, and can provide anytime and anywhere access, massive data storing, sharing and collaboration via a single namespace, and policy management of storage, etc.

4.4 Cloud Data Storage System

CDSSystem is a cooperation of CDS service system with multiple devices, many application domains, and many service forms. The development of CDSSystem is benefit from the broadband network, Web 2.0, storage virtualization, storage network, application storage integrated with servers and storage devices, cluster technology, grid computing, distributed file system, content delivery network, peer-to-peer, data compression, data encryption, etc.

4.5 Cloud Data Storage Security

CDSS involves storage media physical security and data security. As general network storage, the security of CDS includes certification, authority, audit and encryption, etc. Through automatic redundant replications the data will be easy recovery once failover. The CDSS can also expand to the whole procedure of storage service, including hardware, software, data, information, network security and clients' privacy security, etc.

4.6 A security policy

A security policy is a set of rules for determining the maximum permissible access rights for a particular process to a particular segment, given the attributes of both the process and the segment.

5. Overview OF Security Framework

This section describes the security framework to facilitate CDSS upload by users in cloud computing and how we intend to apply it jointly with data sources. Figure 2 (Note 1) shows a schematic representation of security framework. The framework has been built by using two layers, more details in (Talib *et al*, 2010a; Talib *et al*, 2010b).

The functionality of those layers can be summarized as follows:

- Agent layer: This layer has one agent: the User Interface Agent. User Interface Agent acts as an effective bridge between the user and the rest of the agents.
- CDS layer: CDS have two different network entities can be identified as follows:
 - ✓ Cloud User: cloud users, who have data to be stored in the cloud and rely on the cloud for data computation, consist of both individual consumers and organizations.
 - ✓ CSP: a CSP, who has significant resources and expertise in building and managing distributed CDS servers, owns and operates live Cloud Computing systems.

6. Overview of Multi Agent System Architecture

In our MAS architecture, we proposed five types of agents User Interface Agent (UIA), User Agent (UA), DER Agent (DERA), Data Retrieval Agent (DRA) and Data Distribution Preparation Agent (DDPA). The architecture of MAS presented in Figure 3 (Note 1), more details in (Talib *et al*, 2010a; Talib *et al*, 2010b).

The rest of architecture of our MAS is described as follows:

6.1 UIA Architecture: Considered as the main and leader agent, this agent is acts as an effective bridge between the user and the rest of the agents. Such agents actively assist a cloud user in operating an interactive interface, recording the messages and data shared among agents and also serves as a data access point for other agents, as well as cloud users.

6.2 DDPA Architecture: Used to tolerate multiple failures in distributed CDS systems. In CDS, we rely on this agent to disperse the data file redundantly across a set of distributed servers. The main goal of this agent is to generate a correctness security policy to secure the CDS.

6.3 DRA Architecture: Used to enable the cloud user to reconstruct the original data by downloading the data vectors from the servers. The main goal of this agent is to generate integrity security policy to secure the CDS.

6.4 UA Architecture: Act as a customer gateway that makes features of MAS accessible to cloud users. It includes responsibility of providing cloud users with real-time information of entities residing in the MAS. User

agent also allows cloud users to control the status of loads based on priority predefined by a cloud user. The main goal of this agent is to generate both confidentiality and integrity security policies to secure the CDS.

6.5 DERA Architecture: Responsible for storing associated DER information, DER information to be stored may include DER identification number, type, local fuel availability, cost function or price at which cloud users agree to sell, as well as DER availability. The main goal of this agent is to generate availability security policy to secure the CDS.

7. Results of Literature Review

7.1 Themes, frameworks, models, architectures and approaches Design, Implementation and Simulation in CDSS

We believe that CDSS in Cloud Computing, an area full of challenges and of paramount importance, is still in its infancy now, and many research problems are yet to be identified. The most promising one we believe is a model in which public verifiability is enforced. Public verifiability, supported by (Shacham. H., & Waters., B, 2008; Ateniese *et al.*, 2007; Shah *et al.*, 2007), allows TPA to audit the CDS without demanding cloud users' time, feasibility or resources. An interesting question in this model is if we can construct a scheme to achieve both public verifiability and storage correctness assurance of dynamic data.

Although tens of researches papers have been published on related topics, security research for CDS is still in its early stage. Here, we will discuss the achievements in two research directions: secure statistic CDS and secure dynamic CDS, from which our proposed security framework benefits. In our work, we attempted to provide a complete security service solution to secure the CDS. To achieve our goal, we proposed MAS architecture for a security service system that consists of five types of agents: User Interface Agent (UIA), User Agent (UA), DER Agent (DERA), Data Retrieval Agent (DRA) and Data Distribution Preparation Agent (DDPA). UIA is acts as an effective bridge between the cloud user and the rest of the agents. Such agents actively assist a user in operating an interactive interface, recording the messages and data shared among agents and also serves as a data access point for other agents, as well as cloud users. DDPA is used to tolerate multiple failures in distributed storage systems. In CDS, we rely on this agent to disperse the cloud data file redundantly across a set of distributed servers. DRA is used to enable the cloud user to reconstruct the original data by downloading the cloud data vectors from the servers. UA is acts as a customer gateway that makes features of MAS accessible to cloud users. It includes responsibility of providing cloud users with real-time information of entities residing in the MAS. UA also allows cloud users to control the status of loads based on priority predefined by a cloud user. DERA is responsible for storing associated DER information, DER information to be stored may include DER identification number, type, local fuel availability, cost function or price at which cloud users agree to sell, as well as DER availability, more details in (Talib *et al.*, 2010a; Talib *et al.*, 2010b).

In (Wang *et al.*, 2009), they investigated the problem of data security in cloud data storage, which is essentially a distributed storage system. To ensure the correctness of users' data in cloud data storage, they proposed an effective and flexible distributed scheme with explicit dynamic data support, including block update, delete, and append. They rely on erasure-correcting code in the file distribution preparation to provide redundancy parity vectors and guarantee the data dependability. By utilizing the homomorphic token with distributed verification of erasure-coded data, their scheme achieves the integration of storage correctness insurance and data error localization, i.e., whenever data corruption has been detected during the storage correctness verification across the distributed servers, they can almost guarantee the simultaneous identification of the misbehaving server (s). Through detailed security and performance analysis, they show that their scheme is highly efficient and resilient to Byzantine failure, malicious data modification attack, and even server colluding attacks.

The computing power in a Cloud computing environments is supplied by a collection of data centers, which are typically installed with hundreds to thousands of servers (Buyya R. & Murshed, M, 2002). The authors built architecture of a typical Cloud based data center consist of four layers. At the lowest layers there exist massive physical resources (storage servers and application servers) that power the data centers. These servers are transparently managed by the higher level virtualization services and toolkits that allow sharing of their capacity among virtual instances of servers. These virtual instances are isolated from each other, which aid in achieving fault tolerant behavior and isolated security context (Smith J. E, & Nair. R., 2005)

(Juels *et al.*, 2007) described a formal "proof of retrievability" (POR) model for ensuring the remote data integrity. Their scheme combines spot-checking and error-correcting code to ensure both possession and retrievability of files on archive service systems.

In a distributed servers work (Shacham. H, & Waters. B., 2008) built on this model and constructed a random linear function based homomorphic authenticator which enables unlimited number of queries and requires less communication overhead. (Bowers *et al.*, 2008) Proposed an improved framework for POR protocols that generalizes both (Shacham. H, & Waters. B., 2008; Bowers *et al.*, 2008). Our Data Distribution Preparation Agent will do the same job of (Schwarz. T. S. J, & Miller. E. L, 2006; Shacham. H, & Waters. B., 2008; Bowers *et al.*, 2008)

(Bowers *et al.*, 2008) proposed an improved framework for POR protocols that generalizes both (Juels & Shacham's) work. Later in their subsequent work, (Bowers *et al.*, 2008) extended POR model to distributed systems. However, all these schemes are focusing on static data. The effectiveness of their schemes rests primarily on the preprocessing steps that the user conducts before outsourcing the cloud data file. Any change to the contents, even few bits, must propagate through the error-correcting code, thus introducing significant computation and communication complexity.

In a data possession work (Ateniese *et al.*, 2007) defined the "provable data possession" (PDP) model for ensuring possession of file on untrusted storages. Their scheme utilized public key based homomorphic tags for auditing the data file, thus providing public verifiability. However, their scheme requires sufficient computation overhead that can be expensive for an entire file. In their subsequent work (Ateniese *et al.*, 2008) described a PDP scheme that uses only symmetric key cryptography. This method has lower-overhead than their previous scheme and allows for block updates, deletions and appends to the stored file, which has also been supported in our work. However, their scheme focuses on single server scenario and does not address small data corruptions, leaving both the distributed scenario and data error recovery issue unexplored. Our User Agent will do the same job of (Ateniese *et al.*, 2007; Ateniese *et al.*, 2008)

In another data possession work (Curtmola *et al.*, 2008) aimed to ensure data possession of multiple replicas across the distributed storage system. They extended the PDP scheme to cover multiple replicas without encoding each replica separately, providing guarantees that multiple copies of data are actually maintained. Our DER Agent will do the same job of (Curtmola *et al.*, 2008).

In data integrity work (Filho *et al.*, 2006) proposed to verify data integrity using RSA-based hash to demonstrate uncheatable data possession in peer-to peer file sharing networks. However, their proposal requires exponentiation over the entire data file, which is clearly impractical for the server whenever the file is large. In the same work (Schwarz. T. S. J, & Miller. E. L., 2006) proposed to ensure file integrity across multiple distributed servers using erasure-coding and block-level file integrity checks. However, their scheme only considers static data files and does not explicitly studies the problem of data error localization.

(Shah *et al.*, 2006) proposed allowing a TPA to keep online storage honest by first encrypting the data then sending a number of precomputed symmetric-keyed hashes over the encrypted data to the auditor. However, their scheme only works for encrypted files and auditors must maintain long-term state. (Schwarz *et al.*, 2006) proposed to ensure file integrity across multiple distributed servers, using erasure-coding and block-level file integrity checks. However, their scheme only considers static data files and does not explicitly studies the problem of data error localization.

Cisco has developed the Secure Cloud Data Center Framework. This framework portrays the threat model of a cloud data center and the measures that one can take to mitigate security risks. Additionally, the framework shows the overarching controls, compliance, and SLA components. The key take-away in cloud data center security is that security should not be an afterthought or a building block; it should be pervasively implemented across all layers of architecture. The threat profile consists of elements such as service disruption, intrusion takeover, data leakage, data disclosure, data modification, and finally, identity theft and fraud. A cloud data center would be implemented with visibility and protection aspects across all building blocks (Bakshi, K., 2009).

Cryptography has always been an essential part of security. His challenge here is to apply typical cryptographic schemes to a Cloud Computing environment. Some of the specific weaknesses are concerned with cryptography. Since much of Cloud Computing is based on replication, it is important to maintain the distinctiveness of encryption and decryption keys. Recently, Amazon faced a challenge with this issue on their Cloud systems. (Balding, C., 2008) this problem has since been resolved, but lack of foresight in cryptography can lead to disastrous results. His application will address the challenges presented by distributed cryptography, and comprise a large focus on encryption and key distribution. (Balding, C, 2008) is also encapsulate many areas of computer science. Ideally, he will be able to adjust and adapt existing security applications to serve as at least a basis for application. Generally, security incorporates both hardware and software aspects of computer science. Specifically, the fields related to his project include, cryptography, network security, and software security.

Furthermore, he will incorporate the fields that are related to Cloud Computing, these include networking, operating systems, and virtualization.

7.2 MAS Design, Implementation and Simulation in CDSS

Currently, MAS platforms and standards organizations for MAS offer a security mechanism at a very low level. FIPA (Foundation for Intelligent Physical Agents .Available from:<<http://www.fipa.org/>>) uses an agent platform security manager (Torrellas, G.A.S. & Sheremetov, L.B., 2003) for secure communication. Security related parameters can be specified in an agent message envelope, indicating its requirement for security service, according to the FIPA ACL message standard. The agent platform security manager and the agent management system can then be used in agent platforms to maintain security via public and private key authentication or public key infrastructure. However, this provides secure agent communication at the transportation level only. The lack of security support by FIPA as well as the FIPA compliant Java Agent DEvelopment (JADE) platform has been identified in (Poggi, A, *et al.*, 2001; Poslad, S & Calisti, M., 2000 ; Farkas, C, & Huhns, M.N., 2002).

Again this add-on offers only transportation-level security. Its use of policy files is intended only for general-purpose access control to local resources such as system files, network, and so on. Permissions are defined on the basis of system-oriented actions such as: passing messages, moving among containers, and creating/killing agents. Apart from work at the platform level, some attempts have been made to address the security issues of mobile agents, which are vulnerable to threats due to their mobility (Farkas, C, & Huhns, M.N., 2002). Unfortunately, no general mechanism has been offered to cope with the control of agent access rights to resources such as critical data and services, in the context of their running on behalf of human beings or systems.

Therefore, the responsibility of enforcing security in MAS goes to the agent designers and developers who have to meet particular business-specific needs. Usually distributed over their runtime environment, agents behave and interact with each other dynamically, if not fully autonomously. Access control in such a highly dynamic environment is very difficult due to the large number of agent interactions, unforeseen agent behavior, and agent interaction process that could emerge at runtime. The agent research community and developers largely ignore the security issue given the complexity of implementing MAS-specific security requirements on top of existing functional requirements. Ignorance of security constraints would let the potential autonomous agent behavior impact negatively upon systems, such behavior not being predicted in their original design. Imposing the right control upon agent behavior, on the contrary, can provide authorized system access and resource consumption. Overall, the dynamic feature of agents leads to the fact that they are not always associated with a fixed set of data, but rather intentionally interact with each other and human beings flexibly, in the process of which they may have access to various system data and services. This brings unique challenges and a changed view of security modeling that we must bear in mind.

Security models such as the RBAC model are often concerned about permission control for human users. In MAS, some agents operate on behalf of users and others perform business functions as a result of system design. The concepts of agent rights and agent roles, resembling those of user rights and user roles are absent in the context of system resource access. This imposes difficulty on general modeling of all access subjects when access control is concerned. Both human actors and system agents should be restricted in their behavior in MAS as a result of security needs.

Apart from the access subjects, the access objects need matching notions in the design of a unique MAS security model. The agent concept in the resource access pattern raises the level of abstraction of access subjects. The usual access objects of types of files, objects, and class methods in the OO paradigm, however, do not fit well in the pattern. High level business services that can be used, managed, or composed by agents are sometimes believed at the appropriate level of abstraction of agents. Actually, building agents on top on web services has been advocated in (Petrie, C & Bussler, C., 2003) as an essential way to produce real business value from the agent paradigm. In doing so, agents fulfill system services. We can, therefore, regard agents as the agents of services through which services are accessed and made use of. But work in the direction of secure service access management via agents is rare. Since simply borrowing OO security mechanisms does not work and little work has been done in security modeling in MAS, there is a need to investigate the mainstream security access control models in order to discover a useful one and adapt/extend it as required to fit the MAS context. The alternative of designing a brand new security model applicable only to the MAS domain involves reinventing and may impose a barrier to the existing security engineering community.

(Pikoulas *et al.*, 2002) have developed a system that monitors the user's actions in real time; it also takes appropriate actions if necessary based on the predicted user's behavior. In this approach, the security is managed by a single agent that works at the user's side. Alternatively, the project in (Taraka & Jose, 2002) is based on

multi-agent systems approach and focused on providing a security framework for distributed environments. In (Taraka *et al.*, 2002) framework, the system architecture consists of several agents. Each agent handles the security monitoring at each host to inform the system administrators about an attempted intrusion or misuse. However, it is assumed that agents are able to inspect and discover unusual entries that are documented in log files without describing the mechanisms used.

(Ayesh & Bechkoum., 2000) have developed a framework using multi-agent systems for Internet security. The proposed system architecture of this approach is composed of three different agent types classified on their functionalities. The first type is responsible for intrusion detection; the second type is responsible for encryption and decryption of messages, while the third type can act as the combination of the previous two types. Although this approach has provided useful security system, it does not address some other important issues such as authentication, authorization, digital signature, and verification security services. Other approaches focused on addressing the security issues for mobile agent systems. This approach, such as (Gray *et al.*, 1998), focused on authentication to verify the agent's owners, authorization to assign access restrictions and enforcement to ensure that the agent does not violate these restrictions.

In another approach (Francisco *et al.*, 2002) have dealt with security issues in a project called DEEPSIA (Dynamic on-line Internet Purchasing System based on Intelligent Agents) that supports companies as purchasers in electronic commerce e-procurement processes. They have focused on extending the well known KQML agent communication language to incorporate security functions and proposed a new S-KQML (Secure-Knowledge Query Manipulation Language) that includes authentication, integrity and privacy.

(Lalana *et al.*, 2002) have proposed an approach to solve some of the security problems in multi-agent systems, which utilizes delegation based trust management. However, the main focus of this approach was on authentication and authorization.

In the past decade, Grids (Foster I. and Kesselman C., 1999) had evolved as the infrastructure for delivering high-performance services for computer and data-intensive scientific applications. To support research and development of new Grid components, policies, and middleware; several Grid simulators, such as GridSim (Buyya R. & Murshed, M., 2002), SimGrid (Legrand, *et al.*, 2003), and GangSim (Dumitrescu C. L & Foster., I, 2005) have been proposed. SimGrid is a generic framework for simulation of distributed applications on Grid platforms. Similarly, GangSim is a Grid simulation toolkit that provides support for modeling of Grid-based virtual organizations and resources. On the other hand, GridSim is an event-driven simulation toolkit for heterogeneous Grid resources. It supports modeling of grid entities, users, machines, and network, including network traffic.

Although the aforementioned toolkits are capable of modeling and simulating the Grid application behaviors (execution, scheduling, allocation, and monitoring) in a distributed environment consisting of multiple Grid organizations, none of these are able to support the infrastructure and application-level requirements arising from Cloud computing paradigm. In particular, there is very little or no support in existing Grid simulation toolkits for modeling of on-demand virtualization enabled resource and application management. Further, Clouds promise to deliver services on subscription-basis in a pay-as-you-go model to Cloud customers. Hence, Cloud infrastructure modeling and simulation toolkits must provide support for economic entities such as Cloud brokers and Cloud exchange for enabling real-time trading of services between customers and providers. Among the currently available simulators discussed in this paper, only GridSim offers support for economic-driven resource management and application scheduling simulation.

Another aspect related to Clouds that should be considered is that research and development in Cloud computing systems, applications and services are in their infancy. There are a number of important issues that need detailed investigation along the Cloud software stack.

Topics of interest to Cloud developers include economic strategies for provisioning of virtualized resources to incoming user's requests, scheduling of applications, resources discovery, inter-cloud negotiations, and federation of clouds. To support and accelerate the research related to Cloud computing systems, applications and services; it is important that the necessary software tools are designed and developed to aid researchers.

The High Level Architecture (HLA) is an architecture that supports distributed simulation through the reuse and interoperation of simulations (Dahmann *et al.*, 1998). In HLA, individual simulations, referred to as *federates*, participate in a *federation*, which is a set of interacting simulations. All interactions between federates occurs through the runtime infrastructure (RTI), which acts as a distributed operating system for the federation. Some simulations *publish* objects through the RTI and are responsible for updating attribute values for these objects. Other simulations *subscribe* to these objects, which allow them to read object attribute values through the RTI.

Some of the security objectives of HLA are to support federations that function at varying security levels, to support simulations operating at different security levels within a federation, to reuse simulations at different times in different federations and with different security levels, and to support confidentiality, integrity, and “need-to-know” policies (Bieber *et al.*, 1999; Elkins *et al.*, 1996; Filsinger *et al.*, 1996).

Because of the tightly-coupled nature of HLA federations, the approaches to HLA security all need to rely on the runtime infrastructure or secure gateways to implement access control and prevent unauthorized federates from obtaining information that they are not authorized to view. The security concerns in ABELS, while falling into the same general categories of authentication, access control, privacy, and integrity, are influenced by the fact that ABELS is a loosely-coupled system in which most information exchange occurs directly between individual GLAs rather than through the brokering system. In ABELS, each GLA and its user entities are considered autonomous from the ABELS cloud, making it appropriate to delegate most of the responsibilities of protecting the entities in the cloud to the individual GLAs responsible for these entities. This is the motivating principle in the design of the ABELS security framework.

(Filsinger, J & Lubbes, H. O., 1996) proposed three solutions: operating a federation with a single high security level, operating a federation with multiple single security levels and security guards and trusted agents between each security level, and finally operating a federation with multiple security domains, with each federate running on a Multi-Level Secure (MLS) host.

(Bieber, P & Siron, P., 1999) took the approach of building a secure sub-layer for the RTI that uses security labels to enforce restrictions preventing some federates from subscribing to restricted objects in the federation.

(Elkins *et al.*, 1996) proposed an option for a secure HLA/RTI interface built on the network layer using the IPSec protocol, in conjunction with a Public Key Infrastructure (PKI).

7.3 Cloud Data Security Policies and its Attributes Investigates in Cloud Computing Environment

Most of the reported research in the literature on security characterization has dealt with the security of computing and information systems from a qualitative point of view. A system is assigned a given security level with respect to the presence or absence of certain functional characteristics and the use of certain development techniques.

(Littlewood *et al.*, 1993) discuss the commonalities between reliability and security from the perspective of evaluating measures of operational security of software systems. (Swiler *et al.*, 1998 ; Jha *et al.*, 2002) use a slightly data structure which they refer to as the *attack graph* to model the security vulnerabilities of a system and their exploitation by an attacker.

A survey conducted by IDC (International Data Corporation) suggests that cloud services are still in the early adoption phase. There is a long list of issues cloud service providers need to address. The survey has rated security as the most prominent concern (Gens, F., 2008).

(Cachin *et al.*, 2009) in their survey, gives insight into the well known cryptographic tools for providing integrity and consistency for data stored in clouds. The security solutions explored and discussed by them are keeping a local copy of the data, use of hash tree, protocols such as Proofs of Retrievability (POR), and Proofs of Data Possessions (PDP), Digital Signatures etc. These solutions still require a testing on some live data to validate their suitability and ease of use. A whitepaper by AWS (Amazon Web Services) discusses physical security, backups, and certifications in their context. Similarly, other providers such as Google, Microsoft etc. have discussed the security issues in cloud computing (Google App Engine., 2008; Microsoft Live Mesh., 2008).

(Brodin, J., 2008) have identified seven prominent risks that customers must assess in order to utilize cloud computing infrastructure. In addition to these seven issues, they have also identified several other major issues that must be addressed by the cloud service providers. These issues include data storage, server security, privileged user access, and data portability. They also present virtualization specific security issues in detail.

(Manchala, D.W., 2000) has built a trust model in a distributed computing paradigm. To the best of their knowledge, none of the work so far, gives a direction to address the security challenges, specifically in cloud environments. Despite the fact that, there are solutions to address the prominent security issues, a mechanism to measure the security risk from the perspective of a service user is strongly needed. Trust models have been studied in distributed information systems (John, H., 2009; Manchala, D.W., 2000). Adopting some of the ideas of trust modeling, our work identifies a key set of trust variables and a resulting trust matrix, based on security issues in cloud computing.

8. Conclusions

The literature reviewed a number of salient themes, frameworks, models, architectures and approaches important for this study. From this review we believe that, in order to facilitate security of CDS based on MAS architecture is very complex and occurs in multiple levels.

To the knowledge of author, no previous work has been carried out in associating of MAS security techniques and CDS. Although, MAS in security is not a new issue and now it is recognized as one of the most complex problems and due to the usage of agent in security engineering is still in its infancy.

References

- Anderson R. (2001). In: Security engineering: a guide to building dependable distributed systems. New York: John Wiley & Sons Inc; 2001.
- Ateniese. G, Burns. R, Curtmola. R, Herring. J, Kissner. L, Peterson. Z, & Song. D, (2007). "Provable Data Possession at Untrusted Stores," *Proc. Of CCS '07*, pp. 598–609
- Ateniese. G, Pietro. R. D, Mancini. L. V, & Tsudik. G. (2008). "Scalable and Efficient Provable Data Possession," *Proc. of SecureComm '08*, pp. 1–10.
- Ayesh A, Bechkoum K. (2000). "Framework of multi-agents internet security system". Appl Inform (AI'2000).
- Bakshi, K. (2009) "Cisco Cloud Computing - Data Center Strategy, Architecture and Solutions". Point of View White Paper for U.S. Public Sector, 1st Edition
- Balding, C. (2008). "Is Your Amazon Machine Image Vulnerable to SSH Spoofing Attacks?" Cloud Security. Available at: <http://cloudsecurity.org/2008/07/14/is-your-amazon-machine-image-vulnerable-to-sshspoofing-attacks/>.
- Bieber. P and Siron. P. (1999). "Design and implementation of a distributed interactive simulation security architecture". In *Proceedings of the 3rd IEEE International Workshop on Distributed Interactive Simulation and Real-Time Applications*, pp 113–119
- Brodkin, J. (2008). Seven Cloud Computing Security Risks, available at: <http://www.gartner.com/DisplayDocument?id=685308>. (Accessed on May 2010)
- Bowers. K. D, Juels. A, & Oprea. A. (2008). "HAIL: A High-Availability and Integrity Layer for Cloud Storage," Cryptology ePrint Archive, Report 2008/489, <http://eprint.iacr.org/>.
- Buyya R. and Murshed, M. (2002). "GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing", The Journal of Concurrency and Computation: Practice and Experience (CCPE), Vol 14, Issue 13-15, Wiley Press.
- Cachin, C., Keider, I., Shraer, A. (2009). "Trusting the Cloud". IBM Research, Zurich Research laboratory
- Curtmola. R, Khan. O, Burns. R, & Ateniese. G. (2008). "MR-PDP: Multiple- Replica Provable Data Possession," *Proc. of ICDCS '08*, pp. 411–420.
- Dahmann, J. S, Fujimoto, R. M and Weatherly. R. M. (1998). "The DoD High Level Architecture: An update". In *Proceedings of the 1998 Winter Simulation Conference*, pp 797–804
- Dumitrescu C. L and Foster. I. (2005). "GangSim: a simulator for grid scheduling studies". Proceedings of the IEEE International Symposium on Cluster Computing and the Grid
- Durfee, E.H., Lesser, V.R. and Corkill, D.D. (1989). "Trends in Cooperative Distributed Problem Solving". In: *IEEE Transactions on Knowledge and Data Engineering*, March 1989, KDE-1(1), pages 63-83.
- Elkins, A, Wilson, J. W and Gracanin. D. (1996). "Security issues in High Level Architecture based distributed simulation". In *Proceedings of the 2001 Winter Simulation Conference*, pp 818–826
- Farkas, C, Huhns, M.N. (2002). "Making agents secure on the semantic web", IEEE Internet Computing (6) (2002)76–79.
- Filho. D. L. G, & Barreto. P. S. L. M. (2006). "Demonstrating Data Possession and Uncheatable Data Transfer," Cryptology ePrint Archive, Report 2006/150, <http://eprint.iacr.org/>.
- Filsinger, J and Lubbes, H. O. (1996). "System security approach for the High Level Architecture (HLA)". In *Proceedings of the 14th Workshop on Standards for Interoperability of Distributed Simulation (winter)*

- Francisco M, Edson M, Joao P, Pedro S, Adolfo G. (2002). "Dealing with security within DEEPSIA Project". In the Proceedings of the WSEAS International Conference on Information Security, Hardware/Software Code sign, E-Commerce and Computer Networks 2002;2431–9.
- Foster I. and Kesselman C. (1999). "The Grid: Blueprint for a New Computing Infrastructure". Morgan Kaufmann.
- Foundation for Intelligent Physical Agents .Available from:<[http:// www.fipa.org/](http://www.fipa.org/)>. (Accessed on April 2010)
- Gens, F. (2008). "IT Cloud Services User Survey", part 2: Top Benefits and Challenges
- Google App Engine. (2008). available at: <http://appengine.google.com>. (Accessed on April 2010)
- Gray RS, Kotz D, Cybenko G, Rus D. In: Vigna G, editor. (1998). "D'Agents: security in a multiple-language, mobile agent system". Mobile Agents and Security; 1998.
- Java Agent DEvelopment Framework, Available from:<<http://jade.tilab.com/>>. (Accessed on April 2010)
- Jha, S. Sheyner, O. and Wing. J. (2002). "Minimization and reliability analysis of attack graphs". Technical report, CMU Tech. Report, CMU-CS-2-109.
- John, H. (2009). "Security Guidance for Critical Areas of Focus in Cloud Computing", <http://www.cloudsecurityalliance.org/guidance/> (Accessed 2 July 2010)
- John, W. Rittinghouse Jame, F. & Ransome. (2010). "Cloud Computing Implementation, Management and Security", CRC Press, p. 153. (Chapter 6)
- Juels. A. Burton. J & Kaliski. S. (2007). "PORs: Proofs of Retrievability for Large Files," *Proc. of CCS '07*, pp. 584–597.
- Lalana K, Tim F, & Anupam J. (2002). "Developing secure agent systems using delegation based trust management". In Proceedings of Security of Mobile Multi-Agent Systems Workshop (AAMAS 2002)
- Littlewood, B. Brocklehurst, S. Fenton, N. Mellor, P. Page, S. & Wright. D. (1993). "Towards operational measures of computer security". *Journal of Computer Security*, 2:211.229
- Legrand, A. Marchal, L. & Casanova. H. (2003). "Scheduling distributed applications: the SimGrid simulation framework". Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid
- Manchala, D.W. (2000). E-Commerce Trust Matrix and Models
- McClure S, Scambray J, Kurtz G. (2003). In: "Hacking exposed: network security secrets and solutions". McGraw-Hill Osborne Media; 2003.
- Microsoft Live Mesh. (2008). available at: <http://www.mesh.com>. (Accessed on April 2010)
- Pikoulas J, Buchanan W, Mannion M, Triantafyllopoulos K. (2002). "An intelligent agent security intrusion system". IEEE International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS 2002), Lund, Sweden 2002
- Poggi, A. Rimassa, G. Tomaiuolo, M. (2001). "Multi-user and security support for multi-agent systems", in: Proceedings of WOA, 2001.
- Poslad, S. Calisti, M. (2000). "Towards improved trust and security in FIPA agent platforms", in: Proceedings of the Autonomous Agents, 2000.
- Schwarz. T. S. J, & Miller. E. L. (2006). "Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage," *Proc. of ICDCS '06*, pp. 12–12.
- Shacham. H, & Waters. B. (2008). "Compact Proofs of Retrievability," *Proc. of Asiacrypt '08*.
- Shah. M. A, Baker. M, Mogul. J. C, & Swaminathan. R. (2007). "Auditing to Keep Online Storage Services Honest," *Proc. 11th USENIX Workshop on Hot Topics in Operating Systems (HOTOS '07)*, pp. 1–6.
- Smith, J. E, and Nair, R. (2005). "Virtual Machines: Versatile platforms for systems and processes". Morgan Kauffmann.
- Swiler, L. Phillips, C. and Gaylor. T. (1998). "A graph-based network vulnerability analysis system". Technical report, SANDIA Report, SAND97-3010/1
- Talib, A.M. Atan, R. Abdullah, R. and Murad, M.A.A. (2010a). "A Framework of Multi-Agent System to Facilitate Security of Cloud Data Storage" *Proc. Annual International Conference on Cloud Computing and Virtualization CCV, Global Science & Technology Forum GSTF*, pp. 241.

Talib, A. M., Atan, R., Abdullah, R. & Murad, M. A. A. (2010b). Formulate a Security Layer of Cloud Data Storage Framework Based on Multi Agent System Architecture. *GSTF International Journal on Computing*, ISSN: 2010-2283, Vol. 1, No. 1, 2010.

Taraka DP, & Jose' MV. (2002). "Multi-agent network security system using FIPA-OS". In: In the Proceedings of the IEEE SoutheastCon.

Torrellas, G.A.S & Sheremetov, L.B. (2003). "An authentication protocol for agent platform security manager", in: Proceeding of the Emerging Technologies and Factory Automation, 2003, pp.623–628.

Wang, C. Wang, Q. Ren, K. and Lou, W. (2009). "Ensuring data storage security in cloud computing,"

Note 1:

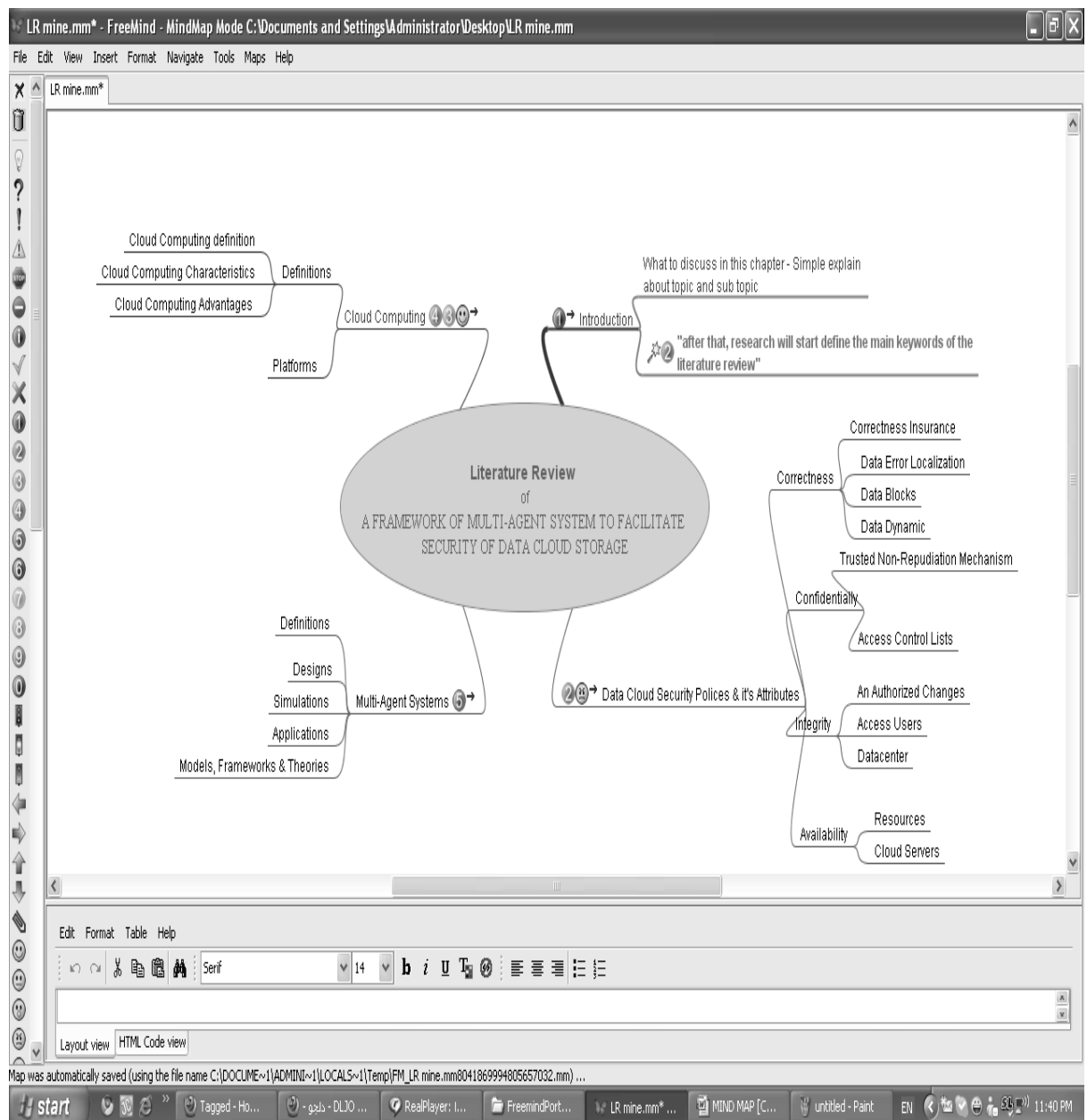


Figure 1. Structure of literature review

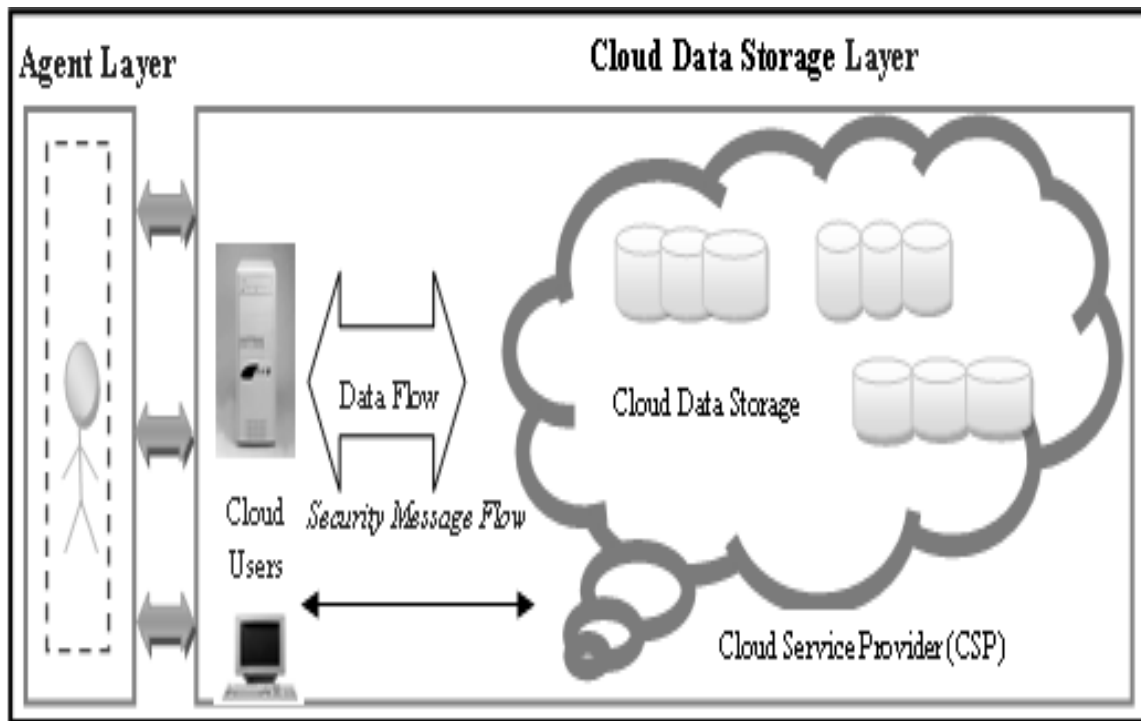


Figure 2. Security Framework

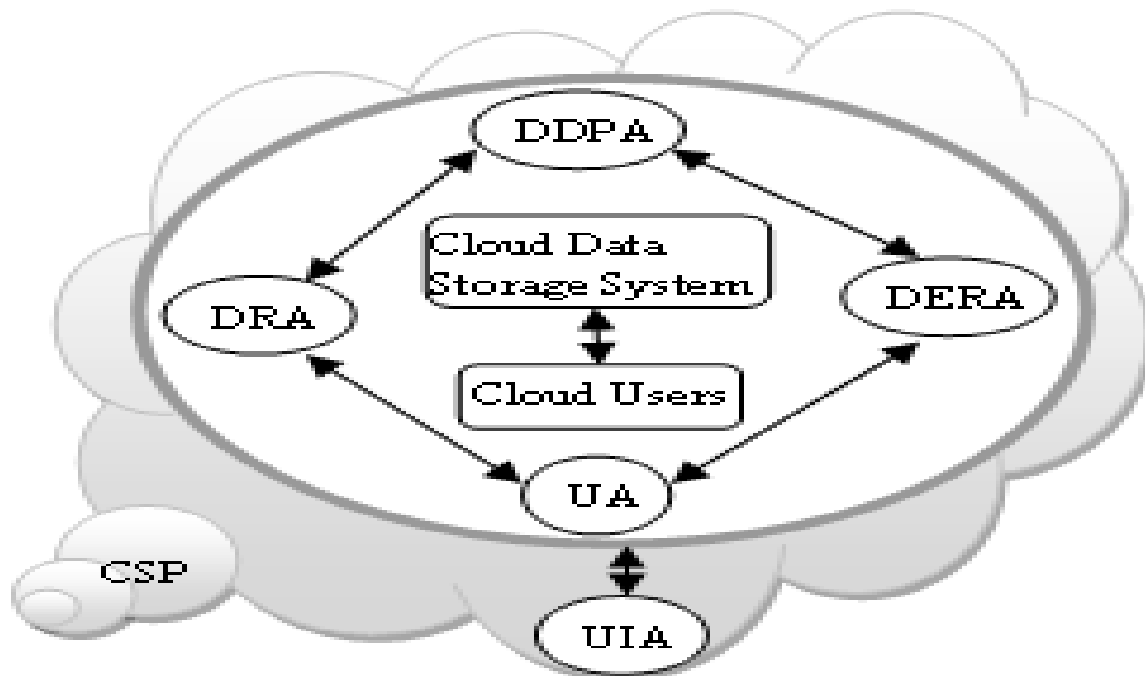


Figure 3. Proposed MAS Architecture