Journal of Software Engineering

ISSN 1819-4311

Academic Journals Inc.

www.academicjournals.com

Journal of Software Engineering 6 (3): 49-64, 2012
ISSN 1819-4311 / DOI: 10.3923/jse.2012.49.64

Security Facilitation in Collaborative Cloud Data Storage Implementation Environment Based on Multi Agent System Architecture

Amir Mohamed Talib, Rodziah Atan, Rusli Abdullah and Masrah Azrifah Azmi Murad
Department of IT, Information System, Faculty of Computer Science and IT, University Putra Malaysia, 43400 UPM, Serdang, Selangor, Malaysia

Corresponding Author: Rodziah Atan, Department of IT, Information System, Faculty of Computer Science and IT, University Putra Malaysia, 43400 UPM, Serdang, Selangor, Malaysia

ABSTRACT

Multi Agent System (MAS) is a collection of an agents that work together to achieve a goal through communication and collaboration among each other. MASs are often distributed and agent has proactive and reactive features which are very useful. Cloud computing moves the application software and databases to the large data centers or Cloud Data Storages (CDSs), where the management of the data and services may not be fully trustworthy. Considering that the data is distributed, updated, created through different sources. This unique attribute however, poses many new security challenges which have not been well understood. To ensure the confidentiality, correctness assurance, availability and integrity of users' data in the cloud, a security framework based on MAS architecture is proposed. This prototype is named as GSecaaS (Ganawa Security as a Service), this prototype tends to use specialized autonomous agents for specific services and allows agents to interact. The proposed MAS architecture includes five types of agents: Cloud Service Provider Agent (CSPA), Cloud Data Confidentially Agent (CDConA), Cloud Data Correctness Agent (CDCorA), Cloud Data Availability Agent (CDAA) and Cloud Data Integrity Agent (CDIA). To simulate the Agents, Oracle database packages and triggers are used to implement agent functions and Oracle jobs are utilized to create Agents. Each agent is considered as an instance of the agent in the environment that can work independently and can communicate with other agents in order to fulfill its needs or fulfill the others requests. Rasch software is used to analyze the data.

Key words: Cloud computing, cloud data storage, cloud service provider, multi agent system, Rasch software

INTRODUCTION

Computer in its evolution form has been changed multiple times, as learned from its past events. However, the trend turned from bigger and more expensive, to smaller and more affordable commodity PCs and servers which are tried together to construct something called cloud computing system. Moreover, cloud computing has benefits in offering minimized capital expenditure, location and device independence, utilization and efficiency improvement, very high scalability and high computing power (Zhou *et al*., 2010).

Cloud computing can be defined as "a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and

presented as one or more unified computing resources based on Service-level Agreements (SLAs) established through negotiation between the CSP and cloud users" (Buyya and Murshed, 2002). Cloud computing environment is supplied by a distributed of Cloud Data Storages (CDSs) which are typically installed with hundreds to thousands of servers. CDS is tending to combined with cloud computing security which will provide more robust security (Zeng *et al*., 2009).

The increasing evolution of cloud computing security in terms of number of cloud users, number of cloud applications and cloud services offered by CSPs, makes them more complex and therefore, more vulnerable to various kinds of complex cloud attacks. However, they are not adapted to dynamic cloud environments, or increasing the complexity of cloud user behaviors. What is needed is a solution that is flexible and adaptable to variations and unpredictable complex evolution of cloud computing security.

Therefore, in order to provide CDS security, we apply our security framework based MAS architecture. To facilitate the security of CDS, we need to:

- Manage cloud security policies specified by CSPs
- Analyze the cloud security events which can characterize security of cloud attacks occurring in the CDS

Multi-agent System (MAS) consists of a number of agents interacting with each other through the Agent Communication Agent (ACL). Agents must be able to interact to achieve goals (Chen *et al*., 2009). Agents may exhibit different types of behaviors when interact with each others such as selfish or benevolent behavior. In CDS scenarios, selfish agents ask for help from other agents if they are overloaded and never offer help such as the agent that serving VIP cloud users for Cloud Service Provider (CSP) service never help other agents for the same service. Benevolent agents always provide help to other agents because they consider system benefit is the priority such as the agent that serving normal cloud users for CSP service are always ready to help other agents to complete their tasks (Talib *et al*., 2011a).

Data security in the cloud is hard to be determined because the functionality of the data is more critical than in the past (Rittinghouse and Ransome, 2009). However, the problem of facilitating security of CDS becomes even more challenging (Wang *et al*., 2009).

Some current security models (Wang *et al*., 2009; Bowers *et al*., 2009; Curtmola *et al*., 2008) has been developed to facilitate security of CDS but all these models produce weak security, because they work only for single cloud server. As a complementary approach, researchers have also proposed distributed cloud protocols (Schwarz *et al*., 2006; Bowers *et al*., 2009) for ensuring security in multiple cloud servers. Again, none of these distributed schemes is aware of dynamic data operations. As a result, their applicability in CDS can be drastically limited.

In a traditional on-premise cloud application deployment model, the sensitive cloud data of each enterprise continues to reside within the enterprise boundary and is subject to its physical, logical and personnel security and access control policies. However, in the CDS, the cloud enterprise data is stored outside the enterprise boundary. Consequently, the CSPs must adopt additional security checks to ensure cloud data security, cloud data fetching and prevent breaches due to security vulnerabilities in the cloud application or through malicious employees. This involves the utilization of the strong encryption techniques, data fetching and fine-grained data access control for CDS security (Mather *et al*., 2009; Rane, 2011).

Recently, the importance of ensuring CDS security has been highlighted by the following research works (Juels and Kaliski, 2007; Shacham and Waters, 2008; Ateniese *et al*., 2007, 2008). These techniques while can be useful to ensure the security without having users possessing data as well they cannot address all the security threats in CDS, since they are all focusing on single server scenario and most of them do not consider dynamic data operations. Hence, the reliability, performance and flexibility of CDS under a number of cloud users, cloud applications and distributed cloud servers should be investigated and clarified. These models, for the prediction of cloud computing security were mainly developed to interpolate/extrapolate experimental mean cloud data. A model for CDS security that under a number of cloud users, cloud applications and distributed cloud servers is therefore, needed.

Key-policy Attribute Based Encryption (K-PABE) is a public key cryptography primitive for one-to-many communications. In K-PABE, data are associated with attributes for each of which a public key component is defined. The encryption key associates with the set of attributes by encrypting it with the corresponding components of the public key. Each single user is defined as an access structure which is usually defined as an access tree over the data file attributes (Goyal *et al*., 2006).

Yu *et al*. (2010) proposed a fine-grained data access control scheme. To achieve the purpose of this scheme, they well exploiting K-PABE and associated it with the both techniques of proxy re-encryption and lazy re-encryption. Moreover, their scheme can enable the CSP to mandate most of computation overhead to powerful cloud servers. Confidentiality policy and user secret key accountability are both achieved. Formal security proofs show that their proposed scheme is secured under standard cryptographic models.

The main objective of this study is to develop a security framework based on MAS architecture prototype and a benchmark system for the evaluation of results. In order to develop our prototype, the following sub-objectives are defined: to propose a new Formula-based Cloud Data Access Control (FCDAC) considering temporal and spatial context in security policy integration, to define protocols of correctness assurance that maintain the same level of storage correctness assurance even if cloud users modify, delete, append or insert their cloud data files in the cloud, to ensure the availability of cloud data by placing each of the vectors on a different server, the original cloud data file can survive the failure of any of the servers without any cloud data loss and to introduce the practical back up cloud data "CloudZone" regularly that provide reconstruct the original cloud data by downloading the cloud data vectors from the cloud servers and whenever the cloud data corruption is detected.

MATERIALS AND METHODS

The post-survey questionnaires were answered by eighteen respondents (2 respondents from Information Security Department at MIMOS Berhad, 7 respondents from Information Security Group (ISG) at Faculty of Computer Science and Information Technology (FSKTM), UPM, 3 security experts and 3 programmers) other 3 respondents from an Open Source Cloud Computing Environment in University Putra Malaysia (UPM) in where GSecaaS is tested and validated.

The evaluation tool shall need to answer whether the tools really significant for CDS environment to be secured. Five main areas addressed as follow (Lheureux *et al*., 2006) to evaluate GSecaaS:

- **Reliability:** In which how the security solutions in GSecaaS is reliable in term of new SecureFormula, availability and integrity solution

- **Performance:** In which moving the implemented security policies of GSecaaS in cloud can actually improve the performance and the ability to scale the security solution with demand
- **Flexibility and system use:** The extent of the existing GSecaaS security policy being implemented. This includes helpful for cloud users in making decision, recording knowledge/information, communicating information
- **System quality:** defines how good the GSecaaS is in terms of its operational characteristics, such as ease of use, user friendly, response time acceptable, stable, knowledge available, clear knowledge classification, knowledge meaningful and understandable, knowledge important and helpful and easy to create cloud user profile
- **User satisfaction:** overall feelings of pleasure or displeasure regarding GSecaaS. This includes overall satisfaction, GSecaaS efficiency and GSecaaS effectiveness, overall GSecaaS satisfaction and meets knowledge needs

Several evaluation methods were considered for data analysis. However, due to ordinal Likert Scale ordered response data and some missing data, a decision to use Rasch Rating Scale model (Wright and Masters, 1982) for analysis was made, to estimate the probability of respondents' agreeableness on the high importance of the items (components). In addition to the standard reliability test, Rasch model provides additional construct validity checking to see if the hierarchy of questionnaire items' agreeableness makes sense (Baghaei, 2008; Bond and Fox, 2003).

Rasch analysis software called Winsteps (Linacre, 2005) is used for data analysis. The results of the survey are analyzed in three parts; data reliability, fitness of respondent data and questionnaire items data and determination of component groups cut-off points.

The respondents are the cloud users and CSPs who participated in evaluation of GSecaaS. Since the questionnaire items are based on ordered response scale, Rasch Rating Scale Model shall be used to estimate the probability of agreeableness on each item. Again, Winsteps application (Linacre, 2005) shall be used to perform Rasch analysis. One Winsteps sessions is administered for GSecaaS responses. Since the respondents are evaluating the GSecaaS prototypes, the Winsteps sessions shall be anchored on persons, by setting the Person means to 0 (UPMEAN = 0). The data shall be inspected for reliability and fitness, to ensure that the data could be used for further analysis.

Multi agent system architecture: In MAS architecture, we proposed five types of agents: Cloud Service Provider Agent (CSPA), Cloud Data Confidentially Agent (CDConA), Cloud Data Correctness Agent (CDCorA), Cloud Data Availability Agent (CDAA) and Cloud Data Integrity Agent (CDIA). The descriptions of these agents as follow:

Cloud service provider agent (CSPA): Is the users' intelligent interface to the system and allow the cloud users to interact with the security service environment. The CSPA provides graphical interfaces to the cloud user for interactions between the system and the cloud user. CSPA act in the system under the behavior of CSP. CSPA has the following actions: provide the security service task according to the authorized Service Level Agreements (SLAs) and the original message content sent by the CDCorA, CDConA, CDAA and CDIA. The main responsibilities of this agent are: display the security policies specified by CSP and the rest of the agents, designing user interfaces that prevent the input of invalid cloud data, receive the security reports and/or alarms from the rest of other agents to respect, translate the attack in terms of goals, monitor specific activities concerning a part of the CDS or a particular cloud user and creating security reports/ alarm systems.

Cloud data confidentially agent (CDConA): This agent facilitates the security policy of confidentially for CDS. Main responsibility of this agent is to provide a CDS by new access control rather than the existing access control lists of identification, authorization and authentication. This agent provides a CSP to define and enforce expressive and flexible access structure for each cloud user. Specifically, the access structure of each cloud user is defined as a logic formula over cloud data file attributes and is able to represent any desired cloud data file set. This new access control is called as:

- Formula-based cloud data access control (FCDAC)

This agent is also notifies CSPA in case of any fail caused of the techniques above by sending security reports and/or alarms.

Formula-based Cloud Data Access Control (FCDAC) and also named as a SecureFormula it's an access policy determined by the MAS architecture, not by the CSPs (Talib *et al*., 2011a; Talib *et al*., 2011b). It's also define as access is granted not based on the rights of the subject associated with a cloud user after authentication but based on attributes of the cloud user. In GSecaaS, CDConA provide access structure of each cloud user by defining it as a logic formula over cloud data file attribute. SecureFormula is an additional confidentiality layer used by the GSecaaS to verify that the cloud users' login page is a genuine. If you are a cloud user, you are required to register first to the system and write your valid email and enter your SecureFormula during your first login. Your SecureFormula will be sent to your email. Be ensured that, your SecureFormula is not your password. Do not set your SecureFormula to be the same as your password!

Sign in from your computer:

- Enter your Cloud User ID
- Verify that your SecureFormula image is correct
- Confirm by entering your password

Our confidentiality layer guaranteed that, even if your password is correct and your SecureFormula is incorrect, then you will not be able to login.

Cloud data correctness agent (CDCorA): This agent facilitates the security policy of correctness assurance for CDS. Main responsibility of this agent is to perform various block-level operations and generate a correctness assurance when the cloud user performs update operation, delete operation, append to modify operation or insert operation. This agent notifies CSPA in case of any fail caused of the techniques above by sending security reports and/or alarms.

Cloud data availability agent (CDAA): This agent facilitates the security policy of availability for CDS. Main responsibility of this agent is to receive and display the security issues that offer by its sub-agents of CDDPA and CDRA. CDAA facilitate two new techniques of file distribution preparation and file retrieval. This agent is also notifies CSPA in case of any fail caused of the techniques above by sending security reports and/or alarms.

Cloud data availability is to ensure that the cloud data processing resources are not made unavailable by malicious action. Our MAS architecture is able to tolerate multiple failures in cloud distributed storage systems.

Cloud data integrity agent (CDIA): This agent facilitates the security policy of integrity for CDS. It is used to enable the cloud user to reconstruct the original cloud data by downloading the cloud data vectors from the cloud servers. Main responsibility of this agent is backing up the cloud data regularly from "CloudZone" and sending security reports and/or alarms to CPSA when: human errors when cloud data is entered, errors that occur when cloud data is transmitted from one computer to another, software bugs or viruses and hardware malfunctions, such as disk crashes. Our proposed integrity layer named as "CloudZone" (Talib *et al.*, 2011c). In CloudZone, we introduce the first provably-secure and practical backup cloud data regularly that provide reconstruct the original cloud data by downloading the cloud data vectors from the cloud servers.

GSecaaS implementation: GSecaaS has been implemented (~ 30.000 lines of JAVA code) with Oracle 11 g. The implementation was based on structure-in-5 MAS architectures described above. We briefly describe the GSecaaS implementation to illustrate the role of the agents and their interaction. To simulate the agents, Oracle database packages and triggers are used to implement agent functions and Oracle jobs are utilized to create agents. Each agent is considered as an instance of the agent in the environment that can work independently and can communicate with other agents in order to fulfill its needs or fulfill the others requests. Illustrated of the implementation as:

SecureFormula (Confidentiality policy): When an on-line cloud user gets connected to GSecaaS, an instance of the Front-Store is created to display an interface that allows the new coming cloud user to register or login. Then, the Back-Store handles the information provided by the cloud user and checks its validity.

Enter your cloud user ID: Before cloud user enter, may need to register as the first time login and type the SecureFormula (Fig. 1).

This process shall read the parameters from CDConA and pass the parameter to CSPA:

Fig. 1: First time login and typing the SecureFormula

- Parameters are cloud user ID, Secure formula and password
- Build parameter Request XML
- Generate submitter serial No
- Submit to Agent_Comm_Q table with Submitter Agent=CDConA, Receiver Agent='CSPA', rule_code = 'NEW_USER', Status=NEW, Type=I (Immediate)

Verify that your SecureFormula image is correct (Fig. 2). Confirm by entering your password (Fig. 3).

In our system, we extended the (Goyal *et al.*, 2006; Yu *et al.*, 2010) works in which our proposed MAS architecture can the able to provide the cloud user key based on confidentiality policy attribute as a FCDAC as illustrated in Fig. 4.

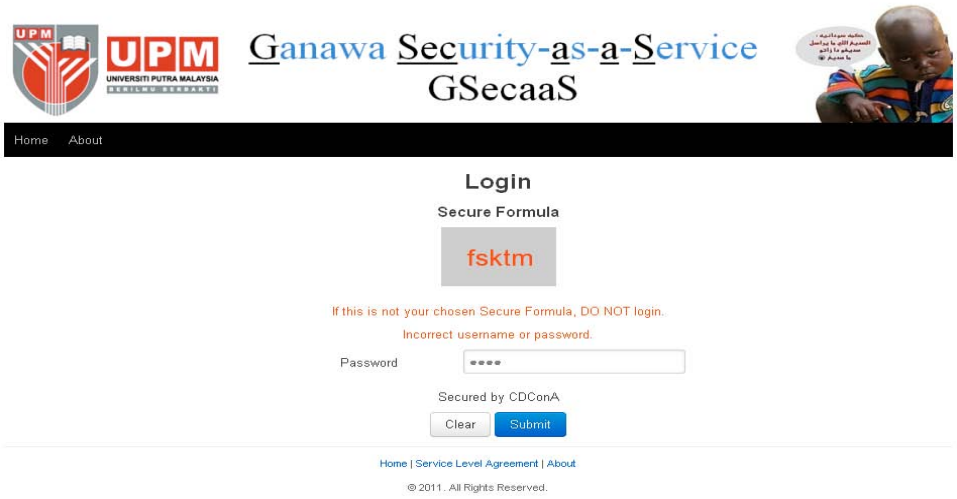

Fig. 2: SecureFormula Verification

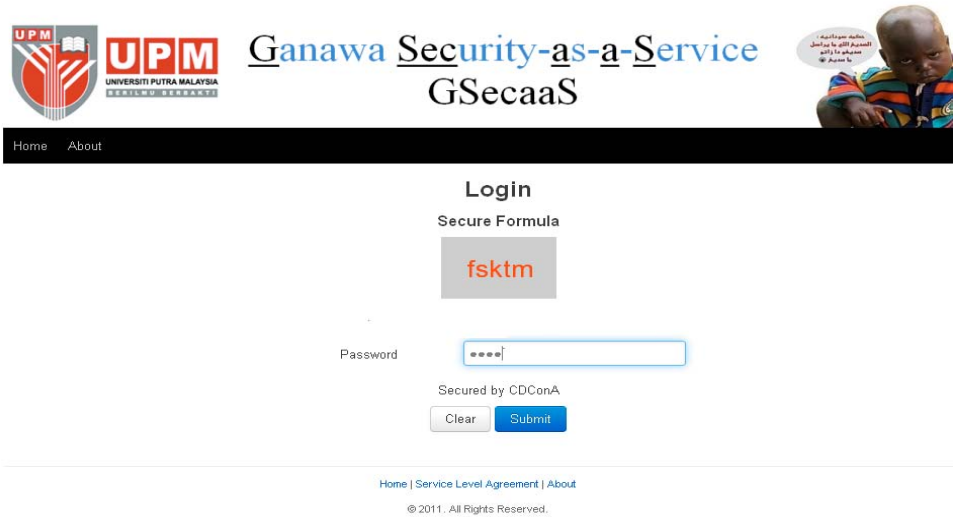

Fig. 3: Enter the password

user_id	user_first_name	user_last_name	user_password	user_secure
amir			$2a$08$jVxDYP3hnQGxGehhbymvDOUtdoi7PO38dv0qykt9XmZ...	FSKTM
tester	tester		$2a$08$X95A8n/U/7xtQSnopNpnQ.ZytZHuBBr5BuNHbvG4.OH...	Tester
tester2	NULL	NULL	$2a$08$KGCN.sPRvLuLn4Xl1jPbjO7Q5RJhXzaNUuyJwTqKSQQ...	wahaha
tester3	Tester	Rock	$2a$08$DKgIY9ZhTBOsMsPziVg/aeu9SkpkZkMD7fjYjTMr1iJ...	nothing

Fig. 4: Password Encryption and its FCDAC Illustration

Correctness assurance policy: In CDS, there are many potential scenarios where data stored in the cloud is dynamic, like electronic documents, photos or log files etc. Therefore, it is crucial to consider the dynamic case, where a cloud user may wish to perform various block-level operations of update, delete and append to modify the data.

Our proposed correctness assurance protocol is not going to be genuine if there is absent of SecureFormula. So in case of: update operation: The cloud user need to enter his/her SecureFormula plus 00, delete operation: The cloud user need to enter his/her SecureFormula plus 01, append operation: The cloud user need to enter his/her SecureFormula plus 10 and modify operation: The cloud user needs to enter his/her SecureFormula plus 11.

This process shall read the parameters from CDCorA and pass the parameter to CSPA:

- Parameters are Cloud User SecureFormula, Update function code, Append function code, Delete function code and modify function code
- Build parameter request XML
- Generate submitter serial No
- Submit to Agent_Comm_Q table with Submitter Agent=CDCorA, Receiver Agent='CSPA', rule_code = 'NEW_USER_PROFILE', Status=NEW, Type=I (Immediate)

Figure 5 shows the user interface for correctness assurance protocol.

Availability and "CloudZone" (Integrity and pack up policy): Before the cloud user pack up, the cloud user may ensure his/her data is available as illustrated in Fig. 6.

This process shall read the parameters from CDAA to ensure the availability of the cloud data and pass the parameter to CSPA:

- Parameters are Cloud User ID, SecureFormula, Password, Update function code, Append function code, delete function code and modify function code
- Build parameter request XML
- Generate submitter serial No
- Submit to Agent_Comm_Q table with Submitter Agent=CDAA, Receiver Agent='CSPA', rule_code = 'DATA_AVAILABLE', Status=NEW, Type=I (Immediate)

This process shall read the parameters from CDIA to ensure the back up of the cloud data from the "CloudZone" and pass the parameter to CSPA:

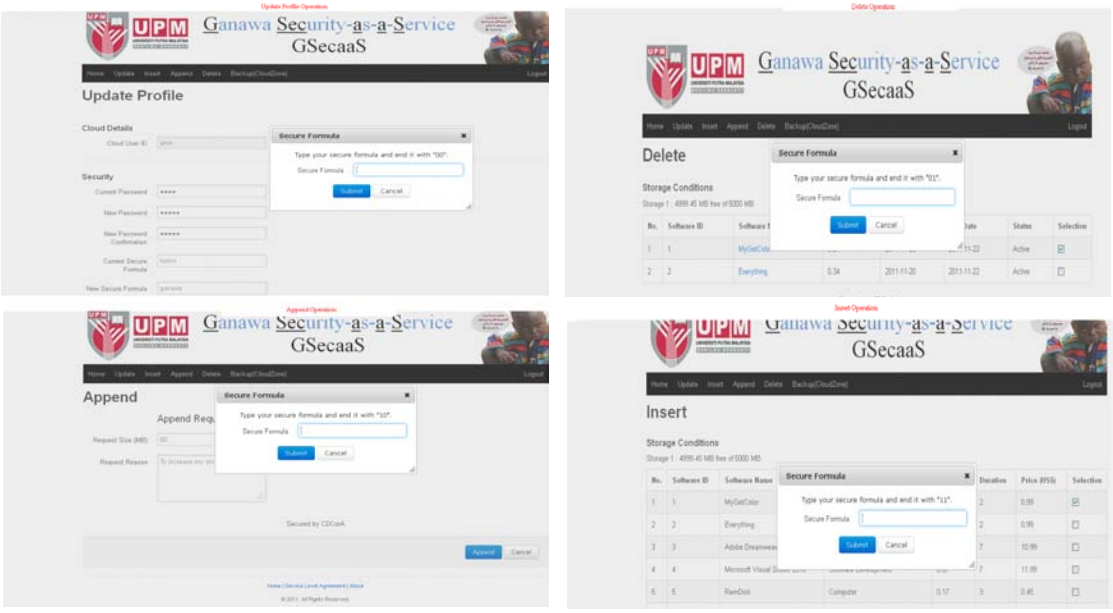

Fig. 5: Correctness assurance protocol

Fig. 6: Back up the data from "CloudZone"

- Parameters are cloud user ID, secure formula, password, update function code, append function code, delete function code and modify function code
- Build parameter request XML
- Generate submitter serial No
- Submit to Agent_Comm_Q table with Submitter Agent=CDIA, Receiver Agent='CSPA', rule_code = 'NEW_BACKUP', Status=NEW, Type=I (Immediate)

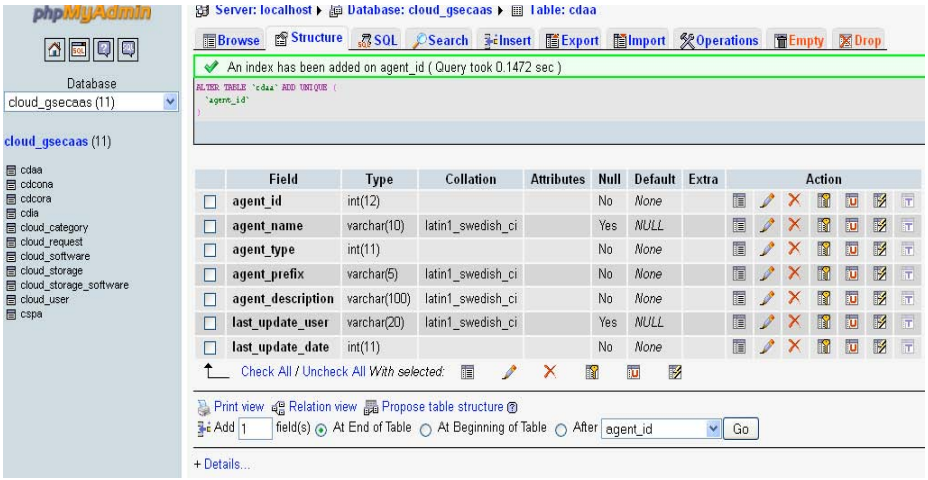

Fig. 7: CSPA stored as a table in Oracle database

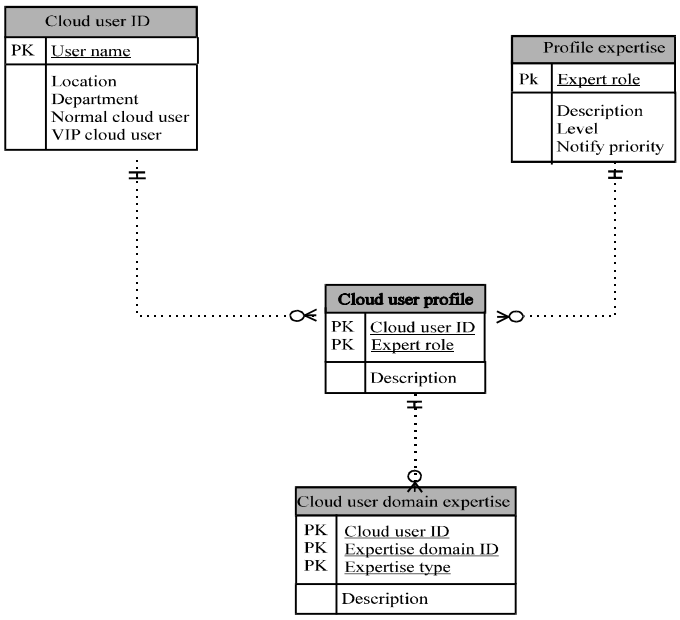

Fig. 8: Cloud user profile ERD

Data (both setups and transactions) shall be stored in an Oracle database. As much as possible, the connection between application and database shall use the native Oracle connection for faster and reliable connected sessions.

For usability, GSecaaS shall be developed for web interfaces, where cloud users from different locations can access the system using Internet web browsers. Meanwhile, for data stability, Oracle database shall be used to store information and data.

There are two types of collaborations involved in GSecaaS system. First, communications between GSecaaS and human roles shall be mainly in forms of messages and screen responses. Communications between software agents shall be carried out using a communication queue stored as a table in Oracle database as illustrated in Fig. 7.

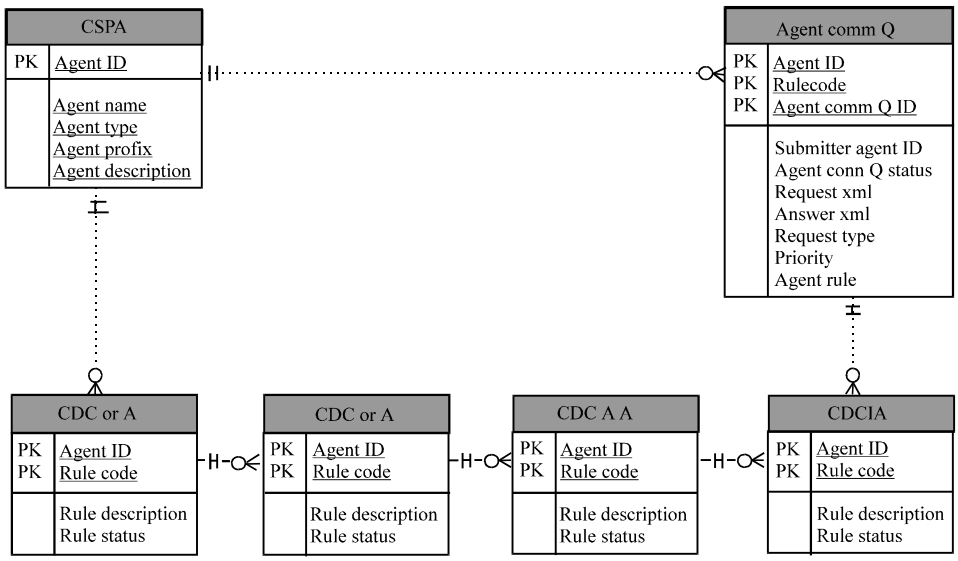

Fig. 9: Multi Agent ERD

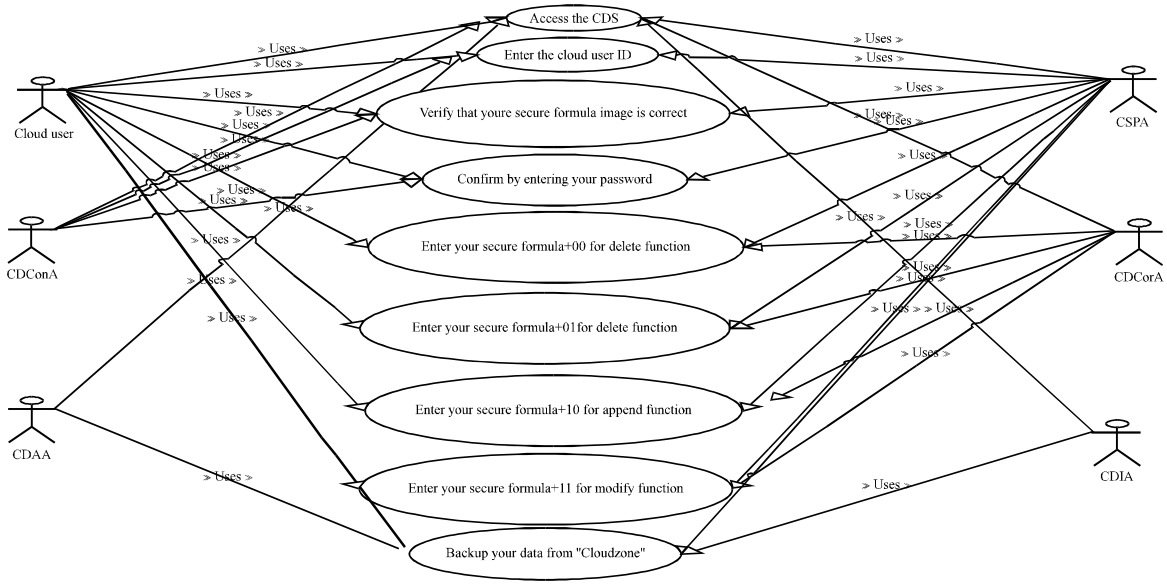

Fig. 10: GSecaaS Feature Agents Functionality

The Entity-relationship Diagrams (ERD) as illustrated for cloud user profile in Fig. 8, multi agent in Fig. 9 and GSecaaS Feature Agents Functionality in Fig. 10, respectively.

RESULTS

The GSecaaS questionnaire results are tabulated. The questionnaire items' measure for GSecaaS is listed in Table 1.

Data reliability and fitness: The summary statistics provided by Rasch, via Winsteps, are listed in Table 2-3. The summary of measured persons are centered with Mean = 0, to allow high

Table 1: GSecaaS item measures

Item	Description	Score	Measure	SE	Infit MNSQ	Infit ZSTD	Outfit MNSQ	Outfit ZSTD	Pt. Mea. Corr
RE: Reliability									
RE1	Reliability of Secure Formula policy functionality	90	1.30	1.65	1.39	1.1	1.31	1.7	1.21
RE2	Reliability of correctness assurance protocol functionality	95	1.55	1.65	1.44	1.3	2.36	1.9	1.14
RE3	Reliability of availability policy solution	78	1.27	1.72	1.25	0.7	1.11	1.4	0.88
RE4	Reliability of integrity policy "CloudZone" functionality	92	1.39	1.59	1.52	2.4	2.35	1.7	1.07
PE: Performance									
PE1	Easy to create knowledge documents	99	1.60	1.88	1.34	1.9	1.35	0.8	1.26
FL: Flexibility and system use									
FL1	Easy to make a decision	80	1.32	1.72	1.25	1.7	1.11	1.4	0.88
FL2	Recording knowledge/information	90	1.37	1.52	1.27	1.8	1.61	1.1	0.36
FL3	Communicating information	80	1.32	1.72	1.25	1.7	1.11	1.4	0.88
SQ: System quality									
SQ1	Ease of use	90	1.30	1.65	1.39	1.1	1.31	1.7	1.21
SQ2	User friendly	95	1.55	1.65	1.44	1.3	2.36	1.9	1.14
SQ3	Response time acceptable	78	1.27	1.72	1.25	0.7	1.11	1.4	0.88
SQ4	Stable	92	1.39	1.59	1.52	2.4	2.35	1.7	1.07
SQ5	Knowledge available	95	1.55	1.65	1.44	1.3	2.36	1.9	1.14
SQ6	Clear knowledge classification	99	1.60	1.88	1.34	0.9	1.35	0.8	1.26
SQ7	Knowledge meaningful and understandable	99	1.60	1.88	1.34	0.9	1.35	0.8	1.26
SQ8	Knowledge important and helpful	92	1.39	1.59	1.52	2.4	2.35	1.7	1.07
SQ9	Easy to create cloud user profile	99	1.60	1.88	1.34	1.9	1.35	0.8	1.26
US: User satisfaction									
US1	GSecaaS efficiency	92	1.39	1.59	1.52	2.4	2.35	1.7	1.07
US2	GSecaaS effectiveness	92	1.39	1.59	1.52	2.4	2.35	1.7	1.07
US3	Meet knowledge needs	95	1.55	1.65	1.44	1.3	2.36	1.9	1.14
US4	Overall GSecaaS satisfaction	99	1.60	1.88	1.34	1.9	1.35	0.8	1.26
Mean		67.1	1.77	1.69	1.95	1.2	1.01	2.1	
SD		8	1.34	119	1.37	1.0	1.75	2	

Pt. Mea. Corr = Point Measured Correlation

Table 2: GSecaaS-summary of measured persons

	Raw score	Count	Measure	Model error	Infit MNSQ	Infit ZSTD	Outfit MNSQ	Outfit ZSTD
Mean	91	29	0.00	0.47	1.01	0.00	1.00	0.00
SD	08	0	1.58	0.04	0.36	1.10	0.47	0.80
Max.	112	29	4.11	0.58	1.78	2.10	1.92	1.80
Man.	80	29	-2.44	0.41	0.55	-1.70	0.41	-1.20

Real; RMSE = 0.51, Adj.SD = 1.50, Separation = 2.95, Person reliability = 0.90, Model; RMSE = 0.47, Adj.SD = 1.51, Separation = 3.22, Person reliability = 0.91, SE of Person's mean = 0.42, Person raw score-to-measure correlation = 1.00, Cronbach's alpha (KR-20) Person raw score reliability = 0.92

Table 3: GSecaaS - Summary of measured items

					Infit		Outfit	
	Raw score	Count	Measure	Model error	MNSQ	ZSTD	MNSQ	ZSTD
Mean	47	15	0.77	0.69	0.95	0.00	1.00	0.10
SD	4	0	1.34	0.19	0.37	1.00	0.75	1.00
Max.	55	15	3.85	1.25	1.74	2.50	3.22	2.20
Min.	36	15	-2.02	0.49	0.39	-1.60	0.09	-1.50

Real; RMSE = 0.76, Adj.SD = 1.11, Separation = 1.46, Item reliability = 0.68, Model; RMSE = 0.72, Adj.SD = 1.13, Separation =1.58, Item reliability =0.71, SE of item mean = 0.25

Table 4: t-test results

Segment	Mean GSecaaS	SD GSecaaS	t-test value	df = (n-1)	t-value @95%
RE: Reliability	0.89	1.22	2.23	46	2.37
PE: Performance	0.84	1.08	2.18	25	1.91
FL: Flexibility and system use	0.87	1.18	2.21	30	2.15
SQ: System quality	0.90	1.24	2.26	55	2.77
US: User satisfaction	0.98	1.65	2.77	65	3.43

description items of GSecaaS. The summary shall also provide the reliability of data (via Person/Item reliability and Cronbach Alpha), Raw Score, Range (Max. measure – Min. measure), Mean measure, Model errors, Infit MNSQ and ZSTD and Outfit MNSQ and ZSTD. All measures are in logit scale, where the higher the logit value, the more difficult for respondents to agree on the corresponding questionnaire items.

The t-test: Five t-tests (at 95% confidence level), are conducted to discuss and explain the GSecaaS success segments (Table 4).

Person-item differential map: The Person Item Differential Map (PIDM) provides the summarized graphical distribution of items measure against persons.

DISCUSSION

The purpose of this section is to propose explanations for the results observed:

Discussion of GSecaaS item measures: Based on Table 1 above, the summary of items measure (activities) for GSecaaS is depicted in Table 5.

GSecaaS keeps track of security domain process information which stores and updates the expertise information. Due to different processes used and different agents' activities in, too much analysis could be carried out. Analysis method is therefore, required to gain a better perspective in GSecaaS activities.

Discussion of measured persons and items reliability: Based on the above summary tables (Table 2-3), the GSecaaS person reliability and item reliability are 0.90 and 0.68 which are very good and fair, respectively. The Cronbach-alpha person raw score reliability is 0.92. These indicate that GSecaaS responses are reliable for further analysis. This indicates that the data fits the Rasch rating scale model and could be used for further measurements.

Table 5: Summary of GSecaaS activities measured

Metrics/Area	GSecaaS counts
No. of initial security domain Info	143
No. of tasks	
Raised	60
Searches	20
No. of security domain info	
Raised/Updated	55
Searches	24
No. of agent advices/Accepted	
Main	30/1
Sub	15/4

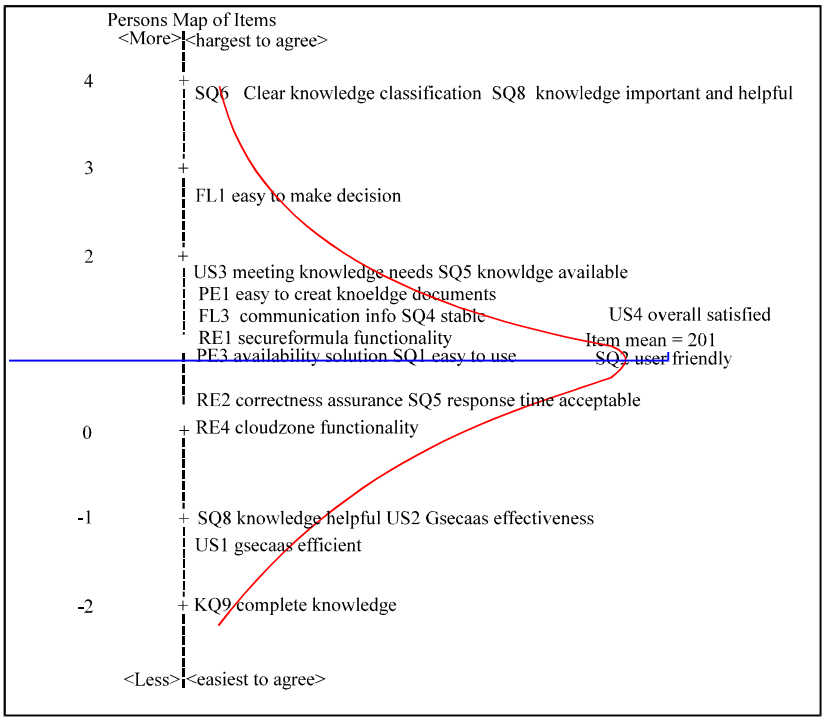

Fig. 11: GSecaaS PIDM

Discussion of t-tests: Based on the above t-test results (Table 4), GSecaaS, in overall MAS Success items, is significantly (t-test value of 2.77). Within the MAS Success segments, GSecaaS could also be concluded to be significantly -PE-performance (t-test value of 2.18), RE-reliability (t-test value of 2.23) and FL-flexibility and System Use (t-test value of 2.21) and SQ System Quality (t-test value of 2.26).

Discussion of PIDM: PIDM for GSecaaS depicted in Fig. 11. Based on the PIDM, RE2 (correctness assurance protocol functionality), SQ3 (response time acceptable), RE1 (SecureFormula functionality), SQ1 (Easy to use) and SQ5 (user friendly) are most agreeable items for GSecaaS. The item mean for GSecaaS is 2.1 logit.

CONCLUSION

To further verify the security framework, a prototype MAS called GSecaaS was developed and implemented for evaluation in an Open Source Cloud Computing Environment in University Putra Malaysia (UPM). To simulate the Agents, Oracle database packages and triggers are used to implement agent functions and Oracle jobs are utilized to create Agents. Each agent is considered as an instance of the agent in the environment that can work independently and can communicate with other agents in order to fulfill its needs or fulfill the others requests. To evaluate the prototype, five main criteria were selected to evaluate whether GSecaaS really secure the CDS. Based on the survey results, with t-test value of 2.77 (in Rasch model measured at 95% confidence level), GSecaaS, in overall, is significant. Rasch software is used to analyze the data.

REFERENCES

Ateniese, G., R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson and D. Song, 2007. Provable data possession at untrusted stores. Proceedings of the 14th ACM Conference on Computer and Communications Security, October 29, 2007, Alexandria, Virginia, USA., pp: 598-609.

Ateniese, G., R. di Pietro, L.V. Mancini and G. Tsudik, 2008. Scalable and efficient provable data possession. Proceedings of the 4th International Conference on Security and Privacy in Communication Networks, September 22-25, 2008, New York, USA., pp: 1-11.

Baghaei, P., 2008. The effects of the rhetorical organization of texts on the C-test construct: A rasch modelling study. Melbourne Pap. Lang. Testing, 13: 32-51.

Bond, T.G. and C.M. Fox, 2003. Applying the rasch model: Fundamental measurement in the human sciences. J. Educat. Measur., 40: 185-187.

Bowers, K.D., A. Juels and A. Oprea, 2009. HAIL: A high-availability and integrity layer for cloud storage. http://eprint.iacr.org/2008/489.pdf

Buyya, R. and M. Murshed, 2002. GridSim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. Concurrency Computation Practice Exp., 14: 1175-1220.

Chen, B., H.H. Cheng and J. Palen, 2009. Integrating mobile agent technology with multi-agent systems for distributed traffic detection and management systems. Transport. Res. Part C: Emerg. Technol., 17: 1-10.

Curtmola, R., O. Khan, R. Burns and G. Ateniese, 2008. MR-PDP: Multiple-replica provable data possession. Proceeding of the 31st International Conference on Distributed Computing Systems, June 17-20, 2008, Beijing, China, pp: 411-420.

Goyal, V., O. Pandey, A. Sahai and B. Waters, 2006. Attribute-based encryption for fine-grained access control of encrypted data. Proceedings of the 13th ACM Conference on Computer and Communications Security, October 30-November 03, ACM Press, Alexandria, VA, USA., pp: 89-98.

Juels, A. and J.B.S. Kaliski, 2007. PORs: Proofs of retrievability for large files. IACR Eprint Archive, http://eprint.iacr.org/2007/243

Lheureux, B.J., R.P. Desisto and M. Maoz, 2006. Evaluating software-as-a-service providers: Questions to ask potential SaaS providers. Gartner RAS Core Research Note, http://www.gartner.com/id=491288

Linacre, J.M., 2005. Measurement meaning and morality. Ph.D. Thesis, University of Sydney, Australia.

Mather, T., S. Kumaraswamy and S. Latif, 2009. Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance. 1st Edn., O'Reilly Media, Inc., USA., Pages: 338.

Rane, P., 2011. Securing SaaS applications: A cloud security perspective for application providers. Inform. Secur. Manage., 5: 301-310.

Rittinghouse, J.W. and J.F. Ransome, 2009. Cloud Computing: Implementation, Management and Security. 1st Edn., CRC Press, Boca Raton .

Schwarz, T.S.J. and E.L. Miller, 2006. Store, forget and check: Using algebraic signatures to check remotely administered storage. Proceeding of the 26th International Conference on Distributed Computing Systems, July 2006, Ayari, Khelil, pp: 12-22.

Shacham, H. and B. Waters, 2008. Compact proofs of retrievability. Adv. Cryptol., 5350: 90-107.

Talib, A.M., R. Atan, R. Abdullah and M.A.A. Murad, 2011a. Towards new data access control technique based on multi agent system architecture for cloud computing. Proceedings of the International Conference on Digital Information Processing and Communications, July 7-9, 2011, Czech Republic, Ostrava, pp: 268--279.

Talib, A.M., R. Atan, R. Abdullah and M.A.A. Murad, 2011b. Multi agent system architecture oriented prometheus methodology design to facilitate security of cloud data storage. J. Software Eng., 5: 78-90.

Talib, A.M., R. Atan, R. Abdullah and M.A.A. Murad, 2011c. Cloud zone: Towards an integrity layer of cloud data storage based on multi agent system architecture. Proceedings of the Conference on Open Systems, September 25-28, 2011, Langkawi, Malaysia, pp: 189-194.

Wang, C., Q. Wang, K. Ren and W. Lou, 2009. Ensuring data storage security in cloud computing. Proc. Int. Workshop Quality Serv., 186: 1-9.

Wright, B.D. and G.N. Masters, 1982. Rating Scale Analysis. 1st Edn., MESA Press, Chicago, USA., ISBN: 9780941938013, Pages: 206.

Yu, S., C. Wang, K. Ren and W. Lou, 2010. Achieving secure, scalable and fine-grained data access control in cloud computing. Proceeding of the 29th Conference on Information Communications, March 2010, IEEE Press Piscataway, NJ., USA., pp: 534-542.

Zeng, W., Y. Zhao, K. Ou and W. Song, 2009. Research on cloud storage architecture and key technologies. Proceeding of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, November 24-26, 2009, Seoul, Korea, pp: 1044-1048.

Zhou, M., R. Zhang, W. Xie, W. Qian and A. Zhou, 2010. Security and privacy in cloud computing: A survey. Proceedings of the 6th International Conference on Semantics Knowledge and Grid, November 1-3, 2010, Beijing, pp: 105-112.