TAMPER PROTECTION FOR DYNAMIC SERVICE LEVEL AGREEMENT IN INTELLIGENT AGENT BASED MOBILE CLOUD FRAMEWORK

K.S. Arvind[1*], R. Manimegalai[2], K.S. Suganya[3]

[1]Research Scholar, Anna University, Chennai, INDIA
[2]Principal, Park College of Engineering and Technology, Coimbatore, INDIA
[3]Department of Information Technology, Bannari Amman Institute of Technology, Erode, INDIA

ISSN 0976-3104

ABSTRACT

In utility computing, customer requests varying services from software to infrastructure as and when needed. As a result, there are many mobile cloud providers provide this services on demand basis as per current market needs. To ensure compliance and security, these providers establish a contract popularly known as Service Level Agreement (SLA) which states the services and management aspects. Such SLA is usually managed by third party entities and in some case by intelligent autonomic agents. But owing to the growing needs and demands of customers, dynamic SLA was offered by the service providers. A dynamic SLA is a redefined SLA whenever new services are demanded by the customer after SLA negotiation. Consequently these dynamic SLA need tamper protection from unauthorized users to prevent unnecessary changes which may lead to SLA violation consequently losing the trust of customers. This paper proposes a data integrity protection for the dynamic SLA using visual cryptography and semi-homomorphic encryption in intelligent agent based service level architecture. As a result the autonomous intelligent agents, customers and service providers can modify the SLA without decrypting the encrypted SLA and Sitekey authentication thereby achieving tamper protection. Furthermore this also ensures the multiparty involvement in dynamic SLA management in intelligent agent based service level architecture.

KEY WORDS
Service Level Agreement (SLA); tamper protection; dynamic SLA; data integrity; homomorphic encryption

Received: 1 April 2017
Accepted: 30 May 2017
Published: 10 July 2017

***Corresponding Author**
Email: erodearvind@gmail.com
Tel.: +91-9944323386

INTRODUCTION

Mobile cloud computing has become an integral part of both small and large organizations day to day business life which in turn resulted in its explosive growth. According to a recent survey conducted by Forbes, nearly 75 percent of the business users are using mobile cloud platform in one way or another. To achieve compliance and trust between mobile cloud customers and service providers, a contractual agreement was established popularly known as Service Level Agreement (SLA).A SLA is a statement of obligations and expectations between a customer and service provider stating the services, target, restrictions, optional services and penalties and scope of availability. SLA management usually consists of three phases namely creation, operation and removal phase. In creation phase, service provider is discovered and SLA is defined and agreement is established between the service provider and customer.

During operation phase, SLA is monitored for service satisfaction and violation and in case of SLA violation, termination of SLA is carried out. If SLA is violated, then penalties are enforced during removal phase. A Static approach to SLA management is followed in almost all scenarios of mobile cloud computing. But static SLA often tends to wastage of resources of both user and provider.

In recent years, services provided by mobile cloud providers are inter dependent on each other i.e. some services relies on completion of the previous services task and resources are also reallocated based on completion of the previous services task. Hence there is increasing need for modifying SLA after creation phase continuing through operation phase. This gave rise to a solution that is dynamic SLA.A dynamic SLA is a renegotiated SLA which contains new terms and services after a proper SLA negotiation. These dynamic SLA thereby make an intelligent decision in making use of the resources effortlessly and efficiently.

Often this SLA management is entrusted with third party entities. Their main work is to monitor whether the Service Level Objectives stated in the SLA are met or violated and reported properly to the customer. But often there are issues of trust. To overcome this advantage, use of autonomous intelligent agent to manage SLA was proposed here.

But in this intelligent agent based architecture, SLA terms are modified and updated both by customers and service providers as needed giving rising to the question of protection against tampering. Henceforth in this paper, we propose a security framework model which follows homomorphic encryption to encrypt the SLA. Then the encrypted SLA is modified without decrypting the SLA while modifying the terms and only after successful renegotiations the modified terms are enforced in the original encrypted SLA after verifying the authenticity, authorization and access permission of the modifier making it more secure. Subsequently data protection of dynamic SLA is also achieved.

This paper is organized as follows. Section 2 deals with related works researched for this proposal and section 3 describes the modified SLA management for dynamic SLA with looping. Section 4 explains in brief about the homomorphic encryption and their mechanism is explained with a scenario for proper understanding. Then we propose the use of modified Dynamic SLA in IAIS architecture with tamper protection using homomorphic encryption in section 5. In section 5.5 and 5.6, we will discuss about the performance evaluation and security analysis of or proposed methodology. Finally we conclude in Section 6.

RELATED WORKS

The inclusion of SLA management in utility or cloud computing was supported by GRID and telecommunication systems for the purpose of standardizing the process between cloud consumers and cloud service providers [1].Inter cloud, a form of mobile cloud also followed the cloud framework for SLA managements which has gained importance due to Semi-Markov Decision Process (SMDP).[2]

SLA management is important for cloud service providers as they act as a trust material between cloud consumer and cloud service provider. Ruben Trapero et al [3] proposed Security Service Level Agreement secSLA which adopted various security control frameworks such as CCM [4] and RATAX [5].The various phases of secSLA was implemented in SPECS [6].The implemented SPECS project aimed at developing an framework for security SLA which supported the entire lifecycle of secSLA.

Although the various proposed framework for ensuring security have addresses many issues, there is an issue of trust between mobile CSP and users. To prevent this we are proposing the dynamic SLA management which encompasses seven phases which will be discussed below.

To ensure authentication and confidentiality in this framework, we proposed the use of Visual cryptography technique [7] to create a SITEKEY which will be explained in the consecutive sections. Also to ensure the tamper protection of SLA document we make use of semi-Homomorphic encryption [8] technique and propose a novel semi-homomorphic algorithm which was discussed in section 5.4

DYNAMIC SLA MANAGEMENT

As discusses previously, Service Level Agreement is a critical component which ensures correct services are rendered and correct QoS parameters are established to meet the Service Level Objective. To enforce dynamic SLA management by the intelligent agents, we propose a change in the SLA management phases where we include a renegotiation phase in addition to the already existing three phases namely creation, operation, renegotiation and removal phases which is depicted in the [Fig. 1].

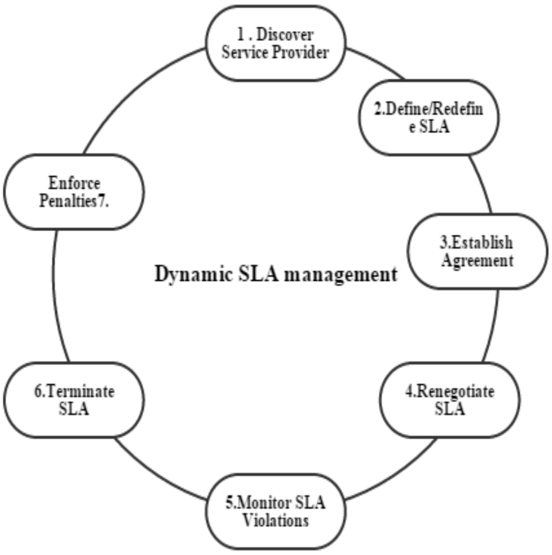

Fig. 1: Dynamic SLA management

The method for dynamically modifying the SLA involves the following steps

1. Discover Service provider based on the need of service by referring preferred service provider
2. Define/Redefine SLA
3. Mutual Agreement on the terms of SLA is established.
4. Renegotiate SLA terms to include new services and go to step 2
5. SLA violation is monitored
6. SLA is terminated
7. Penalties are enforced

Step 1, 2 and 3 involves the creation phase with step 4 forming the renegotiation phase and step 5 and 6 constitutes the operation phase and finally step 7 encompass the removal phase of modified dynamic SLA management. The main difference between the static SLA management and dynamic SLA management is the inclusion of renegotiation phase.

HOMOMORPHIC ENCRYPTION

Homomorphic encryption is a special kind of encryption which allows computation to be performed on the encrypted data which yields the same result as the computation performed on the original data. In simple words, it allows the multi users of an encrypted entity to modify the entity without disrupting the original entity. This scheme was first proposed by Gentry in [9]. As an application, let's say for banking, a user details will be stored in an encrypted form in an untrusted server and if the user wants to query and make minor modification to the details without the knowledge of decryption key, the homomorphic encryption is the suitable mechanism. This method was first developed from RSA which has a multiplicative homomorphic encryption but it was known then as privacy homomorphism. This was later developed into fully homomorphic encryption, partial homomorphic encryption and semi homomorphic encryption with applications varying from multi-party computation, secret sharing, electronic voting etc.

To support our proposal, we will be using semi homomorphic encryption scheme which supports multi-party computation. Because in our intelligent agent based SLA management, both the service provides and customers will modify the SLA terms after renegotiation. We will be making use of semi homomorphic encryption .We make use of a relaxed homomorphic encryption scheme which can recover the plaintext as long as the computed function cannot increase the size of the input too much.

PROPOSED FRAMEWORK

This paper proposes use of dynamic SLA in Intelligent Agent based architecture as an extension to the scheme proposed in [10, 11].This proposed framework employs intelligent autonomous dynamic SLA management with tamper protection for SLA. Instead of entrusting a third party entity with SLA management, this framework makes use of an self-deployed intelligent agent for the following purposes namely SLA definition and agreement, SLA renegotiation, Redefining SLA and Encrypting SLA using semi homomorphic encryption, SLA violation , termination and Penalty enforcement.

SLA Definition and agreement

Intelligent agent is initialized in this phase before SLA negotiation begins. Then, the intelligent agent selects a suitable service provider from a possible list of service providers. Next steps "Define SLA" and "Agreement Establishing between Service Provide and Consumer" are dependent on each other. Then SLA terms and components are defined and commitment protocol, preferred query language are established. After negotiation on both sides, a mutual agreement is established.

SLA Renegotiation

To achieve efficient resource allocation of both users and service provider, we are using dynamic SLA which allows new service request to be added and SLA to be modified after renegotiation between two parties and SLA terms are redefined as needed which is shown in the [Fig. 2].

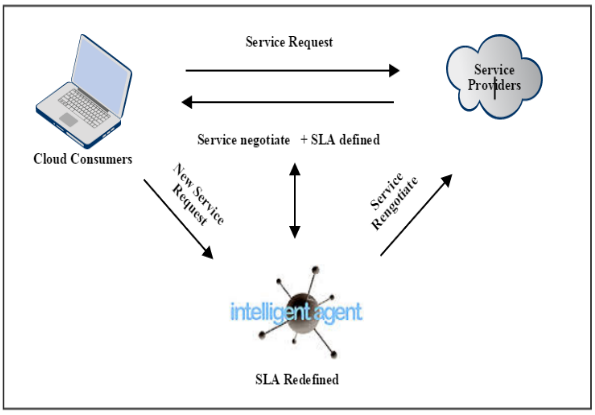

Fig. 2: Dynamic Intelligent agent based dynamic SLA management

SLA Redefinition

Since we need to modify the SLA to encompass the new service request, care should be taken to prevent unnecessary modification to the SLA components and values.

Tamper Protection for Dynamic SLA

ISSN: 0976-3104

As previously mentioned, to protect against unnecessary and unauthorized modification of SLA, we propose a new tamper resistance SLA management which will be explained in the following section. Let us explain the concept with the following scenario. Suppose user A uses his/her smart card to make a transaction with user B and the smart card is tamper resistance, then the both user A and B cant non-repudiate the transaction which occurred between them. In the scenario above, tamper resistance is left to the third party in most cases whose trustworthiness cannot be verified who in turn causes the problems of integrity between User A and User B. To overcome such difficulties, we in our mobile cloud framework propose a tamper resistance SLA renegotiation. Both the cloud consumer and cloud service provider must share a SITEKEY which is generated using Visual cryptography and they must submit their Sitekey shares which when overlaid together must ensure the authenticity of both cloud consumer and cloud service provider as explained in the following [Fig. 3].

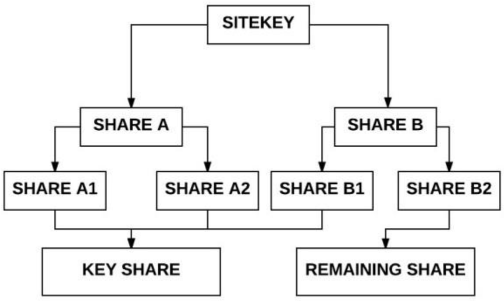

Fig. 3: Creation of site key authentication

In the first level of security, two image shares are generated from the snapshot image by the cloud consumer and service provider namely Share A and Share B. In the next level, the Share A is encrypted again to generate two shares namely Share A1 and Share A2. Likewise, the Share B is encrypted again to generate two shares namely Share B1 and Share B2. Randomly, any three shares form (Share A1/A2/B1/B2) is combined to form the key image share and remaining share is kept aside as an image share. Key image share is shared with the cloud consumer over a secure communication channel. Then the remaining share is stored in the cloud service provider secure database. The above process is explained in the Fig.3. During SLA renegotiation phase, cloud consumer enters their username or customer id along with their image share to authenticate them to the cloud service provider. Then cloud service provider retrieves their share for the corresponding cloud consumer from their database using customer id.

Then the cloud consumer's key image share and cloud service provider's remaining image share retrieved from their database are stacked over one another to retrieve the original snapshot image or Sitekey. Then the SLA renegotiation takes place between the cloud service provider and Cloud consumer and the modified SLA is stored in the semi homomorphic encrypted method [11] to ensure additional security to tamper resistance dynamic SLA management between the cloud consumer and cloud service provider as explained in the [Fig. 4].

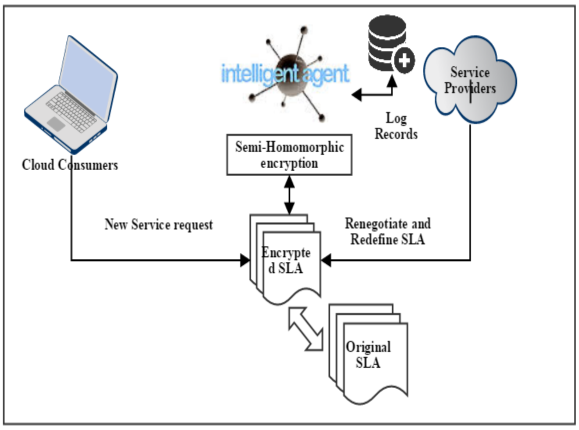

Fig. 4: Tamper Protection for SLA using semi homomorphic encryption

Algorithm for Semi-homomorphic encryption and Decryption

Step 2: Choose 4 bit random integer X' Compute R0 = AB and R1 = AC + XX'

Step 3 : Accept Number N from user by converting the SLA document into integers using ASCII encoding
Step 4: R2 = [T1 R1] mod R0 Encryption Cipher Text C = [N + T2 R2] mod R0 (T1 , T2 are a 4-bit random integer)
Step 5: Decryption N = (C mod A) mod X

RESULTS AND PERFORMANCE EVALUATION

Although achieving security is the main target of this proposal, the proposed system is simulated using Java and the following scenario was created wherein which the cloud consumers and cloud service provider used their Sitekey image to authenticate each other in turn using the following methodology.. According to the proposed methodology, we create a snapshot image as in Fig 5(a). Later, we embedded the secret message in the snapshot image using the above proposed steganography method. Then we generated two image shares namely image share A and B using (2, 2) VCS as in [Fig. 5(b) and (c)] respectively. Furthermore, we generated two image shares namely image share A1 and A2 using (2, 2) VCS from Share A as in [Fig. 5(d)] and [Fig. 5(e)] respectively. Likewise, we generated two more image shares namely image share B1 and B2 using (2, 2) VCS from Share B as in [Fig. 5(f)] and [Fig.(g)] respectively. Then we combined randomly to form a key image share and a remaining share as in [Fig. 5(h)] and [Fig. (i)] respectively. Upon overlaying those image shares, we received the merged shares as in [Fig. 5(j)].

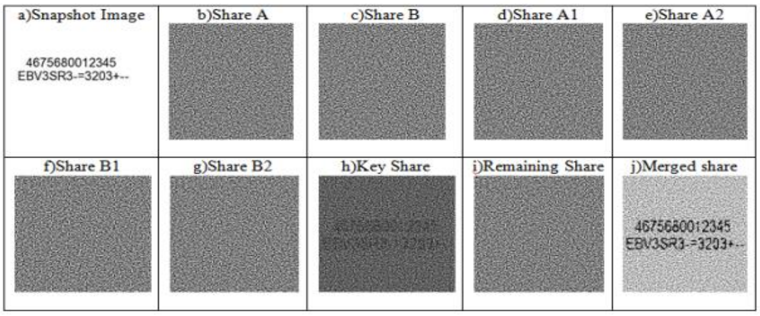

Fig. 5: Experimental results

To include further Tamper protection, we propose the use of semi-homomorphic encryption for storing the SLA by implementing the proposed framework in AWS cloud. The process of implementing the Java module in AWS cloud was explained using the following [Fig. 6].

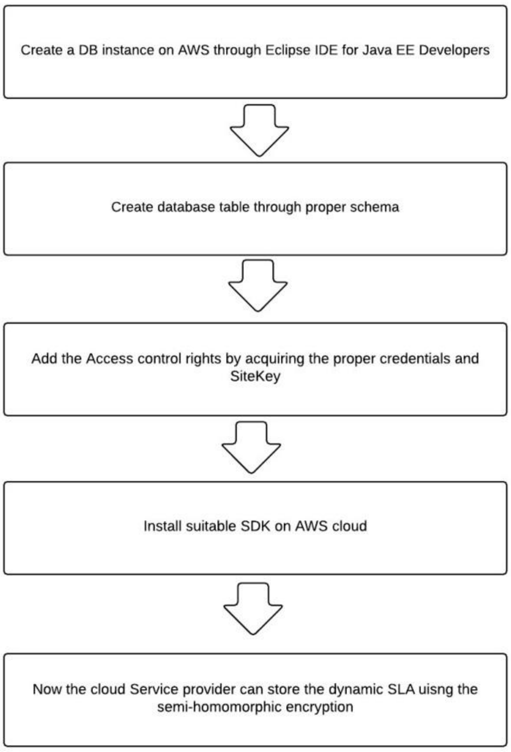
Create a DB instance on AWS through Eclipse IDE for Java EE Developers
Create database table through proper schema
Add the Access control rights by acquiring the proper credentials and SiteKey
Install suitable SDK on AWS cloud
Now the cloud Service provider can store the dynamic SLA uisng the semi-homomorphic encryption

Fig. 6: Dynamic SLA encryption using semi Homomorphic encryption

SECURITY ANALYSIS

In this section, we are analyzing the security of our proposed solution against some security attacks to know their resisting quality, advantages, disadvantages and method extension.

Secret sharing

The proposed solution is implemented using hierarchical visual secret sharing scheme where the cloud consumer's key share and cloud service provider's share are both required to retrieve the secret image. The key share is kept by the cloud consumer and the second share is stored the Cloud service provider's database in a secure manner.

Man-in-the-middle attack

Since only one share is sent across secure communication channel, the man-in-the-middle attack will not succeed in obtaining the access. Adversely, if the share is intercepted by the intruders and the share is duplicated to generate fake share. If the intruders provide the fake share in the payment website which when stacked together with cloud service providers share may retrieve the original image but the cloud service provider will detect anomalies since the fake secret message decoded will not match the cloud consumer entered secret message stored in the cloud service providers secure database suing semi homomorphic encryption.

Security image attack

Security image attack is a special type of attack against SiteKey where the image and phrase column will be replaced with a maintenance message in the fake website which will look legitimate. This can be avoided only through proper awareness among users about this attack.

CONCLUSION

In this paper, we have overcome the difficulties of trustworthiness and integrity by the use of Visual cryptography schemes. Furthermore, it prevents the identity theft by malicious insiders and phishing attack by malicious outsides by the combined use of visual cryptography and semi-homomorphic encryption technique. It also limits the information shared between the cloud service provider and the cloud consumer. It provides protection for high profile users by the use of SiteKey authentication from spear phishing attacks and the modified SLA management for dynamic SLA with looping enables tamper protection against the dynamic SLA management.

CONFLICT OF INTEREST

There is no conflict of interest.

ACKNOWLEDGEMENTS

None

FINANCIAL DISCLOSURE

None

REFERENCES

[1] Bahsoon FF. [2015] A systematic review of service level management in the cloud, ACM Comput. Surv., 48(3): 43:1-43:27

[2] Woungang C, et al. Optimal Cloud Broker Method for Cloud Selection in Mobile Inter-Cloud Computing ,Proceedings of 2015 9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing.

[3] Trapero R, et al. [2016] A novel approach to manage cloud security SLA incidents, Future Generation Computer Systems, http://dx.doi.org/10.1016/j.future.2016.06.004

[4] Cloud Security Alliance, Cloud Controls Matrix v3.0.1. Available online https://cloudsecurityalliance.org/download/cloud-controls-matrix-v3-0-1/ (last accessed in March 2017).

[5] National Institute of Standards and Technology, Cloud Computing: Cloud Service Metrics Description, NIST Public RATAX EG draft document, 2014. Available online http://www.nist.gov/itl/cloud/upload/RATAX-CloudServiceMetricsDescription-DRAFT-20141111.pdf (last accessed in March 2017).

[6] Rak M, et al. [2013] U VillanoSecurity as a service using an SLA-based approach via SPECS,Proceedings of IEEE 5th International Conference on Cloud Computing Technology and Science, CloudComp, IEEE pp. 1–6

[7] Naor M, Shamir A. [1994] Visual Cryptography, In Proceedings of Advances in Cryptography-Eurocrypt '94, LNCS Springer, 950: 1-12.

[8] Bendlin R, et al.[2011] Semi-homomorphic Encryption and Multiparty Computation", Journal of Advances in Cryptology – EUROCRYPT 2011, 6632: 169-188.

[9] Gentry C. [2009] A Fully Homomorphic Encryption Scheme", 2009.

[10] Chieng D, et al. [2002] An Architecture for Agent-Enhanced Network Service Provisioning through SLA Negotiation. In: Bustard D., Liu W., Sterritt R. (eds) Soft-Ware 2002: Computing in an Imperfect World. Lecture Notes in Computer Science, vol 2311. Springer, Berlin, Heidelberg

[11] Brenner M, et al.[2011] Secret program execution in the cloud applying homomorphic encryption, Proceedings of the 5th IEEE International Conference on Digital Ecosystems and Technologies (DEST'11), pp. 114-119

[12] Columbus L. [2014] Mobile cloud Computing Survey," Forbes (26thDecember),at http://www.forbes.com/sites/louiscolumbus/2014/12/26/ kpmgs-2014-mobile cloud-computing-survey-enterprises-quickly-moving-beyond-cost-reduction-to customer-driven-results/,Accessed by 10 Jan 2017.

[13] Alabdulatif A, et al. [2017]. Privacy-Preserving Anomaly Detection in Cloud with a lightweight Homomorphic Approach, Journal of Computer and System Sciences, 1-34