International Journal of Modern Engineering Research (IJMER)
www.ijmer.com *Pp-35-41* *ISSN: 2249-6645*

Trusted SLA Monitoring for Billing System in Public Cloud computing Environment

U. Yamini[1], K. SathiyaPriya[2]
[1]*M.Tech Student, Department of Computer Science & Engineering, SRM University, Kattankulathur,*
[2]*Assistant Professor, Department of Computer Science & Engineering, SRM University, Kattankulathur, Chennai*

ABSTRACT: *This paper presents about a secure and non-obstructive billing system using the concept THEMIS. The main objective of this system is to provide a full-fledged trusted, billing system tailored from a cloud computing environment. The SLA (Service Level Agreement) sharing is done between user and CSPs (Cloud Service Provider). S-Mon forgery-resistive SLA monitoring mechanism is devised by TPM (Trusted Platform Module).*

Index Terms: *Records, Verification, Transaction processing, Pricing and Resource allocation.*

I. INTRODUCTION

Cloud computing is a large group of interconnected computer that has thousands of computers and servers all are linked and accessed via internet. The cloud services changed the economics of computing by enabling users to pay only for the capacity that they actually use. On the basis of verifiability CSPs and users can both construct credible and a verifiable usage records to prove which resources were allocated and when they were initiated. In commercial cloud service the billing transaction and management are both processed by CSP alone; there is no mutual verifiability of billing transactions. A PKI (Public Key Infrastructure) [7] is used mechanism for enforcing the verifiability of billing system. However, the computational complexity of PKI may result in a high computational overhead and intolerable billing response time because asymmetric key operations for digital signatures need to be performed with regards to both the client terminal and CSP. Thus requirements of the cloud billing systems are identified as follows which derive the architecture of the billing system.

For a credible and verifiable way of logging resource usage and a digital signature is important because it enhances the billing mechanism with mutual verifiability. Our proposed billing system, THEMIS uses the concept of a cloud notary authority (CNA) to generate mutually verifiable billing information for users and Cloud Service Providers. Further the resource usage log which is based on a one-way hash chain retains the information in its local storage for future accusations. SLA monitoring module called S-Mon, has a forgery-resistive monitoring mechanism in which even administrator of cloud system cannot modify or falsify the logged data. S-Mon exploits two hardware based security mechanisms: The Trusted Platform Module (TPM) and Trusted Execution Technology (TXT). S-Mon (1) monitors the SLA of CSP and User (2) Take action when violation is detected (3) deliver logged to CNA after service session is finished.

The rest of the paper is organized as follows: In section 2, we discuss about architecture and models of cloud computing. In section 3, we confer about related works. In section 4, we present the THEMIS architecture. In section 5, we illustrate the proposed billing protocol. In section 6, we analyze methodology. Finally in section 7, we present our conclusion and future work.

II. ARCHITECTURE AND MODELS OF CLOUD COMPUTING

2.1ARCHITECTURE OF CLOUD COMPUTING SYSTEM

It all starts with the front-end interface seen by individual users. The user's request then gets passed to the system management, which finds correct resources and then calls the system's appropriate provisioning services. These services slice out the necessary resources in the cloud, cast the appropriate web application and either creates or opens the requested document.

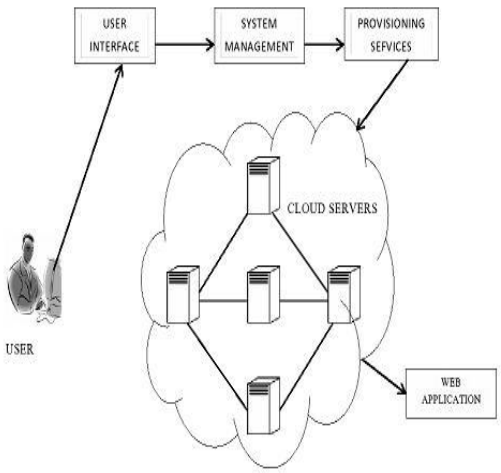

Figure1. Architecture of Cloud Computing

2.2 SERVICE MODELS

Infrastructure-as-a-service (IaaS): which is the basic cloud service model, cloud providers offers computers, and as a physical or more often as virtual machines, and other resources. Other resources in IaaS clouds include images in a virtual machine image library, raw (block) and file-based storage, load balancers, IP addresses, virtual local area networks (VLANs), firewalls, and software bundles. Users use an IaaS may wish to figure out the billed charges for the total service time and guaranteed service level.

Platform as a service (PaaS): This is a category of cloud computing services that provide a computing platform and a solution stack as a service. Along with SaaS and IaaS, it's a service model of cloud computing. In this model, the consumer producesthe software using tools and libraries from the provider. The consumer also charge software deployment and configuration settings. The provider contributes the networks, servers and archive.

Software as a service (SaaS):sometimes referred to as "on-demand software", it's a software delivery model in which software and associated data are centrally hosted on the cloud. SaaS is typically anacquired by users using a thin client via a web browser. SaaS is a common delivery model for many business applications, including auditing, collaboration, customer relationship management (CRM), management information systems (MIS), enterprise resource planning (ERP), invoicing, human resource management (HRM), content management (CM) and service desk management. If a company uses a PaaS or SaaS the accounting department of the company may require the service usage logs so as to verify the billed charges by clicking company's total number of running software program or platforms.

2.3DEPLOYMENT MODELS

Private cloud: It is entirely dedicated to the needs of a single organization. Private cloud can be on or off premises. An on-premise resides in the owner's computer room or data center and managed by the organizations own IT staff. An off-premisetakes advantage of existing facilities and expertise of an outsourcing company such as co-location hosting facility. Advantage of private cloud is that an organization can design it, change it over time, control the quality of service provided. Disadvantages are it requires investment of expertise, money and duration.

Public cloud: It is a multitenant cloud that is owned by a company that typically sells the services it provides to the general public. Advantage of public cloud is that it is always ready to use without delays, no need to invest in internal IT infrastructure.

Hybrid cloud: It is also known as cross-premises cloud uses a private and public cloud at the same time, with services spanning both deployments. Advantage of hybrid cloud are it control of security and compliance from private cloud, cost effective flexibility and scalability from public cloud and a single service spanning both.

Community cloud: It is owned, managed, shared and operated by many organizations. This open cloud can use many technologies. Community clouds are extremely complex and are a shared risk.

III. RELATED WORKS

Billing systems that track and verify the usage of computing resources have been actively studied and developed in the research area of grid and cloud computing. Many studies have analyzed already existing billing systems of grid and cloud computing environments [11], [12]. They have tried to identify the new requirements of the shift in the computing paradigm from grid computing to cloud computing. In this section, we briefly confer experimental results as we evaluate existing billing systems in terms of their security level and billing overhead. We assess the billing systems in an identical computing and network environment.

3.1EXISTING SYSTEM

The billing system with limited security concerns and the micropayment-based billing system require a relatively low level of computational complexity. The micropayment-based schemes such as PayWord [8], MiniPay [6], e-coupons [9],

System	Transaction Integrity	Non-Repudiation	Trusted SLA monitoring	Billing Latency
Billing System with limited Security	No	No	No	Avg. 4.06ms
Micropayment-based Billing System	Yes	No	No	Avg. 4.70ms

PKI-based billing system	Yes	Yes	No	Avg. 82.51ms
THEMIS	**Yes**	**Yes**	**Yes**	**Avg. 4.89ms**

and NetPay [10]. The average billing latency for billing system with limited security is 4.06ms, for micro payment-based billing system is 4.07ms. Nevertheless, these systems are inadequate in terms of transaction integrity, non-repudiation and trusted SLA monitoring. In spite of the consensus that PKI-based billing system offers a high level of security through two security functions excluding trustworthySLAmonitoring, the security comes at the price of extremely complex PKI operations with the average billing latency of 82.51ms. Consequently, when a PKI-based billing system is used in cloud computing environment the high computational complexity causes high deployment cost and high operational overload because the PKI operations must be performed by the user and the CSP.

3.2 PROPOSED SYSTEM

In this paper, we propose a secure and non-obstructive billing system called THEMIS as a remedy for the above mentioned limitations. The system uses a novel concept of a cloud notary authority (CNA) for the supervision of billing. The CNA generates mutually verifiable binding information that can be used to resolve future disputes between a user and CSP in a computational efficient way. Further scalability and fault tolerance is done in banking side by providing security for bill payment which is a web service. This leads to faster time to market, minimal computational cost, accurate, consistent and competitive pricing. The average billing latency of THEMIS is 4.89ms.

IV. THEMIS ARCHITECTURE

While deliberating on the system requirements mentioned above, we position the design of our billing system and protocol on two principles: verifiable billing with a cloud notary authority; and computationally efficient billing transactions. In this section, we presentthe overall architecture and billing process of THEMIS.

4.1. The Proposed THEMIS Infrastructure

Figure2 shows the overall architecture of our THEMIS billing infrastructure. The four major components of the architecture are listed as follows:

Cloud Service Provider (CSP): The CSP enables users to scale their capacity upwards or downwards in accordance with their computing requirements and to pay only for the capacity that they actually use.
Users: We assume that users are thin clients who use services in the cloud computing environment. To use services in such an environment, each user makes a request to the CSP with a billing transaction.

Cloud Notary Authority (CNA): The CNA provides a mutually verifiable integrity mechanism that combats the malicious behavior of users or the CSP. The method, which involves a generation of mutually verifiable binding information among all the involved entities on the basis of a one-way hash chain, is computationally active for a thin client and the CSP.

Trusted SLA Monitor (S-Mon): The S-Mon has a forgery-resistive SLA measuring and logging mechanism, which enables it to monitor SLA violations and take corrective actions in a trusted-manner.

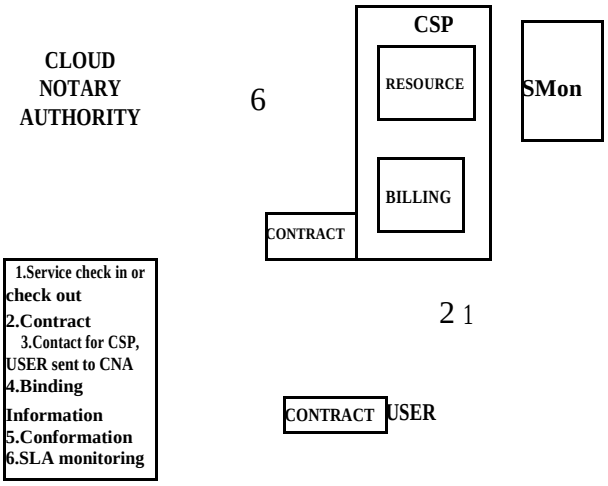

Figure 2.Overall architecture and process flow of THEMIS

After the service period is finished, the data logged by S-Mon are delivered to the CNA. We devised S-Mon so that it can be deployed as an SLA monitoring module in the computing resources of the user.

4.2 Overall Billing Process of THEMIS

After a registration phase, THEMIS can use the above components to provide a mutually verifiable billing transaction without asymmetric key operations of any entities. The registration phase involves mutual authentication of the entities and the generation of a hash chain by each existence. The hash chain element of each entity is integrated into each billing transaction on a hash chain basis; it enables the CNA to verifythe correctness of the billing transaction. In addition, SMon has a forgery-resistive SLA measuring and logging mechanism. THEMIS consequently supervises the billing; and, because of its equality, it is likely to be accepted by users and CSPs akin. The billing transactions can be performed in two typesof transactions: a service check-in for starting a cloud service session and a service check-out for finalizing the service conference. These two transactions can be made in identical way. Each billing transaction is performed by the transmission of a report, called a µ-contract. A µ- contract is a data structure that contains a hashed value of a billing framework and the hash chain element of each entity. With the sole esteem to decrypt both the µ-contract from the CSP and the µ-contract of the user, the CNA can act as a third party to attest the consistency of the billing context between the user and the CSP.

The main steps are as follows:

1. The user generates a cloud resource request message and sends it to the CSP.
2. The CSP sends the user a µ-contract-CSP message generated with a digital signature from a CSP hash chain.
3. The user generates a µ-contract with a hash chain- based digital signature of the user and sends it to the cloud notary authority.
4. The cloud notary authority performs transactions to verify the µ-contract from the user.
5. The billing process is completed when the user and the CSP receive confirmation from the cloud notary authority.
6. Finally, in the case of anassistance check-in, the S-Mon of the user's cloud resource transmits authentication data of the S-Mon to the CNA. In the case of aassistance check-out, S-Mon sends a report of the SLA monitoring results to the CNA.

V. PROPOSED BILLING PROTOCOL

During deliberating on our system design philosophy, we did our utmost to streamline the computation and communication overhead of the billing operation. Our novel µ-contract is accomplished by a hash function and it exclusively distributes keys among entities. It can optimize the computing and communication overheads of the billing mechanism and facilitates mutual verifiability and integrity for cloud resource usage. In this section, we describe the overall transactions of the proposed billing protocol. Table 1 describes the proposed protocol.

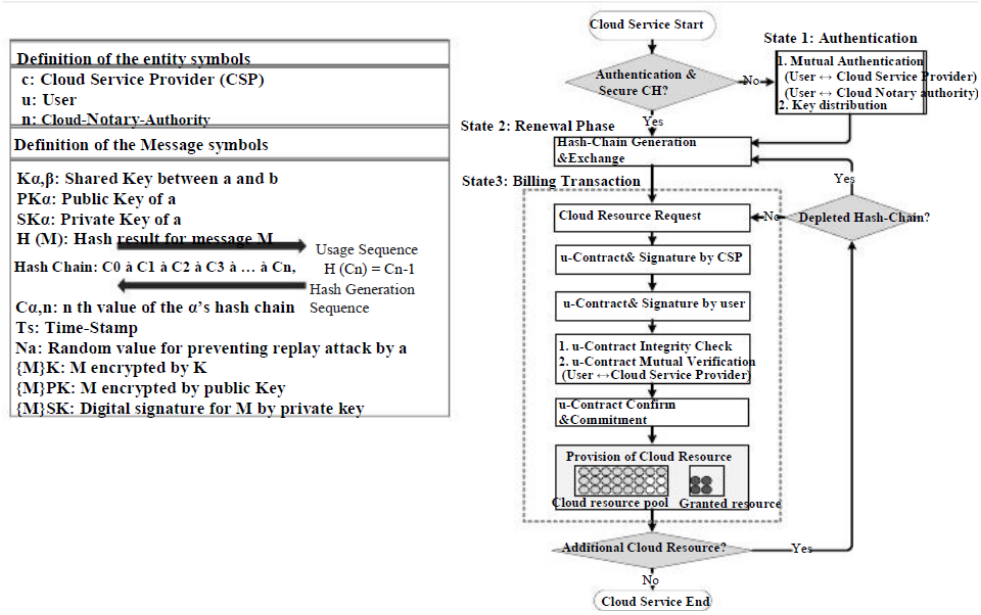

Table1. Notations of the entities and messages

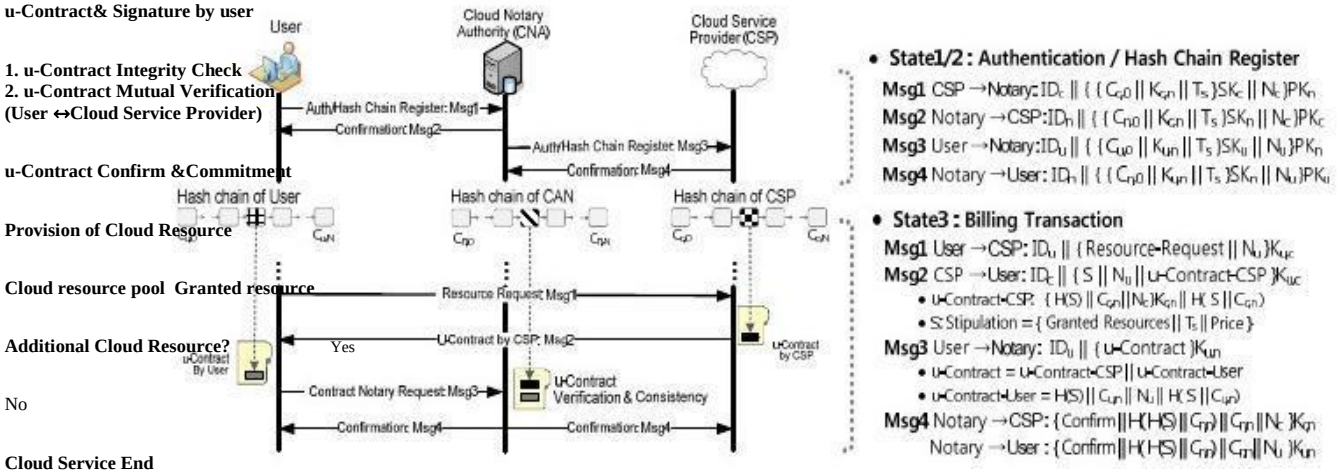

Figure1. Overall billing transactions of THEMIS

5.1. Flow Diagram of the Billing Protocol

Figure 3 present a flow diagram to present the transactions of the proposed billing protocol. Figure 4 describes the overall billing transactions and the message specification that describe the proposed protocol. The protocol consists of three states are:

State 1: When a user first accesses the CSP, mutual PKI-based authentications are performed by the user, the CSP, and the cloud notary authority. Throughout the mutual authentications, the following three keys are shared exclusively among the user, the cloud notary authority, and the CSP:

- *User↔CSP: Ku,c*
- *User↔CNA: Ku,n*
- *CSP ↔ CNA: Kc,n*

State 2: In this phase the user, the CSP, and the CNA generate a hash chain of length N by applying the hash function N times to a random value (Cu,N, Cc,N) so that a final hash (Cu,0,Cc,0) can be obtained. The user and the CSP execute to the hash chain by digitally signing the final hash (Cu,0, Cc,0), and by registering them and the keys (Ku,n, Kc,n). As shown in Figure 4, the purpose of this registration is to share with the cloud notary authority and to receive the hash chain (Cn,0) generated by the cloud notary authority. Once the commitment of the one-way hash chains is strongly completed, State 2 is skipped until the corresponding hash chain either expires or is revoked. The user generates a µ-contractby using the symmetric key operation and hash function and then transmits the contract to the cloud notary authority.

When the µ-contract is received from the user, the cloud notary authority verifies the contract and sends confirmation messages to both the user and the CSP. When the confirmation message is received from the cloud notary authority, the user and the CSP confirm the contract between the user and CSP and the billing operation is terminated.

VI. METHODOLOGY

6.1Modules

- User Interface Design
- Cloud Service Provider
- User
- Cloud Notary Authority (CNA)
- Monitor
- Action against SLA violation
- Bank

6.2Module Description

6.2.1 User Interface Design

User Interface Design have a purpose that a user to move from login page to user page of the website. In this we want to enter our user name and password provided by Service provider. If we enter the valid password and user name then only the user can move login page to user window while entering user name and password it will check username and password is match or not. If we enter any wrong username or wrong password it generates some error message. So we are preventing from unauthorized user entering into the service provider website. It will provide a good security for our project. So Service provider contain user name and password server also check the authentication of the user. It will improve the security and preventing from unauthorized user enters into the website. In our project we are using java swings for creating design. Here we are validating the users who are going to access the Service providers.

6.2.2CloudService Provider

Service provider has a job of providing a service like software to the cloud users. In our proposed method, CSP doesn't

provide billing transaction to the user. It is due to the reason if billing transaction performed in the CSP then complexity in security to be provided for billing transaction increases the overhead. If the user logged in for service, CSP validate the user whether he\she is an authenticated user or not. Once if user is found authenticated user then it waits for service check in message else it found any unauthenticated user it will send the error message. If it received the service check in message then it responds the user by transmitting the agreement and hash chain (one time key). After getting the service request from the user, CSP provide the requested service to the user. It is also have a contact with the Cloud notary authority. It will provide the service until it receive the service checkout message. The CSP enables users to scale their capacity upwards or downwards regarding their computing requirements and to pay only for the capacity that they actually use.

6.2.3User

User can access a service from the Cloud Service Provider by authenticated login process. We assume that users are thin clients who use services in the cloud computing ambiance. To start a service session in such an environment, each user makes a service check-in appeal to the CSP with a billing transaction. To boundary the service session, the user can make assistance check-out request to the CSP with a billing transaction. Once if the users send the service check-in message it can get the contract from the CSP. After receiving the one time keywords in the contract it can be able to access the service from the CSP. Now user log details are stored in Monitor for future disputes. After accessing the service, user want billing transaction. If he\she wants the bill means it should send the contract of the CSP with contract of the user to the CNA. If both the details checked by the CNA are identical then user can receive the bill binding information along with confirmation message. If any error occurred or forgery activity found from the user side then the user will receive the penalty for that.

6.2.4Cloud Notary Authority (CNA)

Cloud Notary Authority acts as a THEMIS in our cloud billing transaction. He is an authority to generate the billing transaction for the cloud service. The CNA provides a mutually verifiable integrity mechanism that combats the malicious behavior of users or the CSP. The course of action, which involves a generation of mutually verifiable binding information among all the involved entities on the basis of a one-way hash chain (One time key), is computationally efficient for a user and the CSP. If user wants billing for the service then it sends the contract of the user and contract of CSP to the CNA. In CNA it checks both the contract; if it is found as identical then it generates the bill as binding information and sends the confirmation message to the user and the CSP. If it is not identical then it receives the log details from the monitor. If forgery found at user side it sends the penalty to the user. If it found at CSP side it cancels the payment to the CSP. CNA provide the billing transaction which can be verifiable and also forgery resistive in cloud environment.

6.2.5Monitor

Monitor is a module which continuously monitors all the log activities of the CSP and the user. For monitoring it uses a technique called S-Mon. The S-Mon has a forgery-resistive SLA measuring and logging mechanism, which enables it to overseer SLA violations and take corrective actions in a devoted manner. After the service session is accomplished, the data logged by S-Mon are dispatched to the CNA. We devised S-Mon in such a way that it can be deployed as an SLA monitoring module in the computing resources of the user. Once SLA has been violated S-Mon sends all the log details to the CNA. After verifying the log details CNA perform further action. Monitor has a local repository for storing all the log details of the user to monitor the SLA for the future disputes. So it can be verifiable in future too. Here monitor plays important role against billing transaction forgery which leads to forgery resistive billing transaction.

6.2.6Action against SLA violation

Once the CNA found forgery from cloud services it can't directly take any action against them without knowing the reason. At that time it sends the message to Monitor to send the all log details about the transaction. Once it receives the log message from the monitor it compares the contract and the log details. Once the forgery found from CSP side it cancels the payment to the CSP and send the message to the CSP. If it found from the user side it assign penalty to the user according to the severity of the forgery from the user side and sends the message to the user. CNA also maintains the local repository after the action taken against the SLA violation.

6.2.7Bank

The security for the SLA from user side is further extended even during the bill payment to the bank. Once the bill is generated and when user tries to pay the bill online through any bank the amount text box is automatically filled by the admin thus user cannot change the value. Even if he tries to edit the amount the system will protect from doing so and protect from paying the wrong bill to bank. Thus scalability fault tolerance and robustness is achieved in throughout the project.

VII. CONCLUSION AND FUTURE WORK

Our motto was to provide a full-fledged verifiable and non-obstructive billing solution tailored for a cloud computing environment. To consummate this task, we thoroughly reviewed the ways in which conventional billing systems are used in the environment, and we consequently derived blueprints for our mutually verifiable and computationally efficient billing system called THEMIS. Moreover consuming conventional billing systems, we conceived and implemented the concept of a cloud notary authority that supervises billing to make it more objective and acceptable to users and CSPs

alike. THEMIS assures actual confirmation of any transaction between a user and a CSP. Furthermore, our mutually verifiable billing protocol cogently reduces the billing overhead. Our next step is to speculate the scalability and fault clemency of THEMIS. Currently, we are critiquing THEMIS from the outlook of massive scalability and strength. We believe that placing multiple devoted third parties in charge of the cloud notary authority is an appropriate way ahead, as is the case with the PKI. We work towards a THEMIS-based system with more fault clemency to scalable billing.

REFERENCES

[1] L. C. M. C. Rob Byron, Roney Cordenonsib, "Apel: An implementation of grid accounting using r-gma," UK e-Science All Hands Conference, Nottingham, September 2005.

[2] Frey, Tannenbaum, Livny, Foster, and Tuecke, "Condor-g: A computation management agent for multi-institutional grids," Cluster Computing, vol. 5, pp. 237–246, 2002.

[3] O.-K. Kwon, J. Hahm, S. Kim, and J. Lee, "Grasp: A grid resource allocation system based on ogsa," in Proc. of the 13th IEEE Intl. Sympo. on High Performance Distributed Computing. IEEE

[4] Computer Society, 2004, pp. 278–279. I. P. Release, "Tivoli: Usage and accounting manager," IBM Press, 2009.

[5] H. Rajan and M. Hosamani, "Tisa: Toward trustworthy services in a service-oriented architecture," IEEE Transactions on Services Computing, vol. 1, pp. 201–213, 2008.

[6] A. Herzberg, H. Yochai, "Mini-pay: charging per click on the web", in: 6th World Wide Web Conference, Santa Clara, USA, April 1997.

[7] Stefan Kelm, "Public Key Infrastructure", http://www.pki- page.org/, 2009

[8] R. Rivest, A. Shamir, "PayWord and MicroMint: two simple micropayment schemes", 1996 International Workshop on Security Protocols, Lecture Notes in Computer Science, vol. 1189, Springer, pp. 69–87

[9] V. Patil, R.K. Shyamasundar, "An efficient, secure and delegable micro-payment system", 2004 IEEE International

[10] Dai, X., and Grundy, J. "NetPay: an off-line, decentralized micro-payment system for thin-client applications", E- Commerce Research and Applications, 6, 2007,91–101

[11] D.Banks, J.S.Erickson, and M.Rhodes, "Towards Cloud-based Collaboration Services", Usenix Workshop HotCloud 2009.

[12] N.Santos, K.P.Gummadi, and R.Rodrigues, "Towards Trusted Cloud Computing", Usenix Workshop HotCloud 2009.