클라우드 컴퓨팅을 위한 상호 검증 가능 과금 시스템 디자인 및 구현

Design & Implementation of Mutually Verifiable Billing System for Cloud Computing Environment

박기웅, 박성규, 박규호
Ki-Woong Park, Sung Kyu Park, and Kyu Ho Park

Computer Engineering Research Lab., KAIST 373-1 Guseong-Dong Daejeon, Korea
{woongbak, skpark, kpark}@core.kaist.ac.kr

요약

일반 사용자를 위한 클라우드 컴퓨팅 서비스 제공 및 클라우드 컴퓨팅의 활성화를 위해 선행되어야 할 과제 중 하나는 상호 검증이 가능한 과금 시스템의 구축이다. 본 논문에서는 PKI기반 상호 검증이 가능한 과금 시스템을 디자인 하고 구현하였다. PKI기반 과금 시스템의 구축으로 사용자의 클라우드 자원 사용기록이 PKI 기반 디지털 서명과 함께 기록이 되므로 상호 검증이 가능하다는 장점이 있지만 빈번한 PKI 개인키 연산은 과금 시스템 및 사용자 단에 상당한 오버헤드로 작용된다. 이와 같은 문제점을 해결하기 위하여 "Cloud-Notary-Authority" 이라는 새로운 개념을 도입하여, PKI에서 제공하는 보안성능을 그대로 유지시켜 클라우드 컴퓨팅 과금 시스템에 대한 디지털 서명 및 부인방지 기능을 제공을 함과 동시에 오버헤드가 심한 보안 연산을 오프로딩 시켜 과금 시스템에 대한 오버헤드를 최소화 시키게 된다. 본 시스템은 KAIST 클라우드 테스트베드에 실질적으로 적용하여 실험을 수행하였으며, 일반 PKI 기반 과금 시스템에 비하여 약 3배 정도의 성능 향상이 있다.

Abstract

The ability to record and account for the cloud resources usage in a credible and verifiable way is a precursor to widespread cloud deployment and availability since usage information is potentially sensitive and must be verifiably accurate. In an attempt to expand Public Key Infrastructure (PKI) usage to a clo ud computing environment to provide a secure and undeniable billing mechanism, we found that the freq uent public key operations for fine-grained billing lead to excessive computation overhead or bottleneck of the billing system. To alleviate these limitations, we propose a PKI-based billing system, namely whi ch is enhanced with the delegation technology to offload complex PKI operations from the thin clients to the cloud infrastructure. In addition, we introduce the "Cloud-Notary-Authority" concept to supervise billing to make it more objective and accepted by both cloud users and providers. The proposed scheme can provide non-repudiation and integrity of the cloud resource usage records by devising a cloud-nota ry-authority that generates binding information between a user and resource usage log and retains the information in its local storage for future accusation. This work has been undertaken within the KAIST

Cloud-Testbed for Campus-wide Educational Cloud Services. According to the performance evaluation, t he throughput of our billing transactions (223.4 transactions per second) is much higher than the throug hput of the PKI-based billing transactions (which averages 61.5 transactions per second).

키워드: 클라우드 컴퓨팅, 과금, 공개키 기반 인프라
Keyword: Cloud Computing, Billing, PKI

1. INTRODUCTION

Pay-as-you-go pricing model is one of the core components of cloud computing environment. It allows users to scale capacity, both up and down, as your computing requirements change. Cloud services change the economics of computing by allowing users to pay only for capacity that users actually use. In this environment, the ability to record and account for the cloud resources usage in a credible and verifiable way is a precursor to widespread cloud deployment and availability. The reason is that usage information is potentially sensitive and must

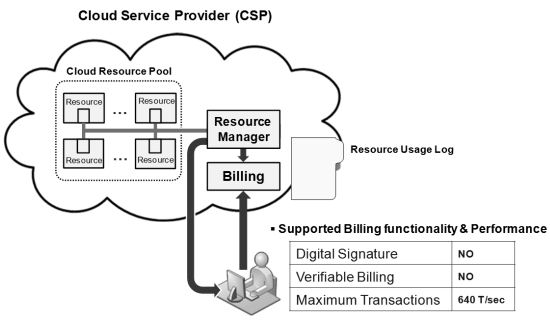


Figure 1. Conventional billing system and a brief overview of the characteristics

be verifiably accurate [1]. Based on the recordability, both cloud service provider and users are able to construct the usage record to prove not only which resources have been allocated, but also when they have been initiated in a credible and verifiable way.

As a fundamental way of enforcing the security of the billing system, digital signature and non-repudiation stand out as the two most widely used mechanisms. A full realization of trustworthy billing services based on digital signature and non-repudiation in a cloud computing environment requires deep understandings of the performance characteristics and the communication patterns of the cloud since fine-grained billing lead to excessive computation overhead or bottleneck of the billing system. From a security aspect, the PKI [17] is generally considered as the most appropriate solution for the requirements. However, the computational complexity of the PKI causes high computation overhead because asymmetric key operations of the PKI need to be performed on thin client terminals. In addition, the frequency billing requests increases rapidly in proportion to the number of users that require diverse cloud resources. The computation required to perform asymmetric key operations may result in intolerable billing response time and bottleneck of the billing system.

By thoroughly investigating conventional cloud billing systems, we identified two fundamental requirements, which drive the architecture of our billing system, are as follows:

1) Fine-Grained billing mechanism with minuscule computing overhead: the frequent PKI operations for fine-grained billing lead to excessive computation overhead or bottleneck of the billing system. To mitigate the problems, we propose a PKI-based billing protocol which is enhanced with the delegation technology to offload complex PKI

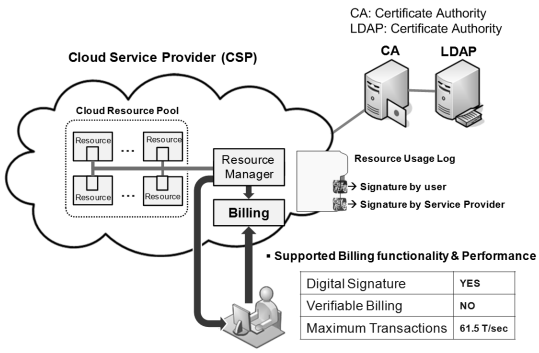


Figure 2. Billing system enhanced with PKI and its characteristics

operations from the thin clients to the cloud infrastructure. The proposed billing scheme provides users and service providers with a seamless metering and integrity for the cloud resources.

2) Support for verifiable billing mechanism by PKI: In the traditional cloud billing system, both the billing management and the billing information are processed by the cloud service provider alone. A digital signature and a non-repudiation mechanism are essential functions to provide a secure and undeniable billing mechanism [2]. Our proposed billing system can provide the two mechanisms by means of the cloud-notary-authority that generates PKI-based binding information between a user and resource usage log and retains the information in its local storage for future accusation. It supervises billing to make it more objective and could be accepted by both cloud users and providers.

The remainder of the paper is organized as follows: In Section 2, we discuss relevant works. In Section 3, we present the overall system design and components of the proposed billing system. In Section 4, we illustrate the proposed billing protocol. In Section 5, we evaluate the performance of proposed billing scheme. Finally, in Section 6, we present our conclusions.

2. Previous Work

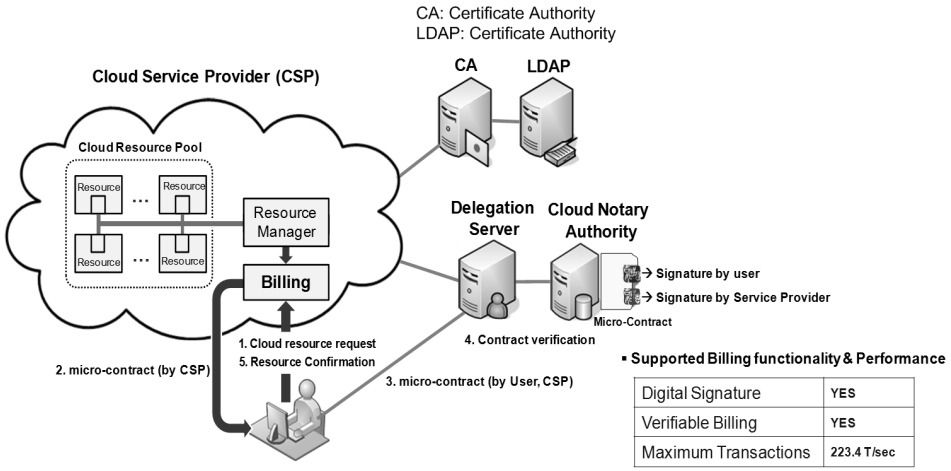


Figure 3. The proposed Billing System

A billing system for charging for computing resources has been actively studied and developed in the research area of grid/cloud computing. However, none of the billing system tackled to verifiable and fine-grained billing mechanism. In this section, we briefly discuss the experimental results as we evaluate the pre-existing billing systems in terms of the security level and the billing overhead. More comprehensive evaluations of the experimental results are described in Section V.

2.1 Traditional billing systems

Figure 1 shows the architecture and characteristics of a billing system without security concern. As the pay-as-you-go pricing model, users can scale capacity, both up and down, as users' computing requirements change by request for the cloud resources. The resource consumption is billed on a utility basis with little or no upfront cost. As a pioneer research, [3-5] identified challenges in managing resources in a Grid computing environment and proposes computational economy as a metaphor for effective management of resources. Several researchers [6-10] present a resource usage processing system, which is used to scan batch system logs to build accounting records. It is able to record and account for the grid resource usage. As shown in Figure 1, resource usage information such as CPU cycles, storage, and network bandwidth are collected via the resource monitor and charged over the billing module. APEL [6] present a billing system by processing the log information to create quantified accounting records. There are also resource management and billing frameworks suggested as part of traditional Grid approaches such as Condor/G [7], Nimrod/G [8], GRASP [9], Tivoli [10]. However, they mainly focused on presenting their distributed resource usage metering, accounting and account balancing mechanism in a distributed Grid environment rather than security concern; hence, these scheme cannot provide a digital signature and non-repudiation for the resource usage record to realize a verifiable billing system.

2.2 Billing systems enhanced with Security

In an attempt to provide a billing and accounting mechanisms enhanced with security, there have been several research works. RUS [1], DGAS [11], SGAS [12], GridBank [13], and Amazon EC2 [14] presented the basic market models and their billing and accounting frameworks enhanced with Public Key Infrastructure (PKI) for user authentication and secure communication. TeraGrid [21] presents a system to monitor their usage via a command line interface as part of their Grid infrastructure. In [12, 13], they presented an access control mechanism that uses digitally-signed certificates to define and enforce an access policy for a set of distributed resources. TTCC [23] presents a closed box execution environment to verify a security level before users receive their cloud resources.

However, these projects do not address the problem of dealing with verifiable and fine-grained billing mechanism as we do for Cloud Computing. PKI, which is a representative security infrastructure based on an asymmetric key, is generally considered to be the most appropriate solution for e-commerce and mutual authentication due to its digital signature and non-repudiation features [15]. The organization of a billing system enhanced with the PKI and its characteristics in terms of security level and billing overhead are illustrated in Figure 2. Even though PKI provides full security features, including authentication, digital signature, and

non-repudiation, it has a severe drawback when used for billing system by a thin client or heavily loaded server, that is, the billing overhead, which is mainly determined by the extremely high complexity of RSA [20] operations for encryption, decryption, and certificate validation in PKI. The estimated maximum billing transaction throughput in our experimental environment (3.2 GHz Xeon processor with 4 Gbyte RAM) is about 61.5 transactions per second as shown in Figure 2.

3. Proposed Billing System Architecture

3.1 Overall system design

To meet the demands of a verifiable and secure billing system, we incorporated PKI into our system as an underlying security infrastructure. Deliberating on the aforementioned system requirements, we based the design of our billing system and protocol on two principles:

- Verifiable Billing with Cloud-Notary-Authority: To record and account for the cloud resource usage in a secure and credible way, we devised a cloud-notary-authority which is designed to provide a computationally efficient digital signature and non-repudiation, generates binding information between a user and resource usage log and retains the information in its local storage for future accusation.
- Verifiable billing operation with minuscule computing overhead: The requirement to perform the frequent and prohibitive PKI operations for fine-grained billing lead to excessive computation overhead or bottleneck of the billing system. To provide a non-obstructive billing operation, we propose a computationally efficient PKI-based billing protocol that is based on a delegation mechanism that uses a proxy certificate [18]. The proposed protocol also provides digital signature and non-repudiation for the resource usage log by identical security functionalities as in PKI.

3.2 The Proposed Infrastructure

Figure 3 shows the overall architecture of our billing infrastructure. The four major components of our architecture are listed as follows:

- PKI: it is considered as a promising foundation consists of a CA, an LDAP.
- Cloud service provider: it allows users to scale capacity in accordance with computing requirements of users and to pay only for capacity that users actually use.
- Delegation server: it is designed to offload complex PKI-related operations from a user to the infrastructure, making it possible to realize a thin client with a low computation power. The delegation server also maintains all of the proxy certificates that contain the private keys and public keys that are delegated and signed by a user. Upon performing the authentication for the first time, the user delegates the billing operations to the delegation server by following RFC3820 [18]. The delegation server subsequently takes over the entire authentication operations until the users' proxy certificate expires.
- Cloud Notary Authority: it provides a non-repudiation mechanism for combating malicious user or service provider's behavior. The non-repudiation mechanism takes effect as long as the user uses her/his own private key in a billing process. However, after delegating its operations to the delegation server, the user no longer uses her/his

private key as a means of halting the provision of the verifiable billing mechanism. We therefore devised a cloud notary authority so that, even during the delegation process, we could bring the mechanism back into our system. The cloud notary authority investigates billing transactions, generates binding information between a user and the cloud service provider, and retains the information in its local storage for future accusation. Due to the cloud notary authority, our proposed billing system can provide a symmetric key-based non-repudiation mechanism that is computationally efficient on both the thin client and service provider. More specific details of the non-repudiation mechanism are described in Section IV.

3.3 Overall Billing Process

The overall billing transaction on the basis of our billing system is presented in Figure 3. The proposed cloud notary authority and delegation server make it possible to provide a verifiable billing with only three symmetric key operations on the user side after the delegation, thereby providing the identical security level of billing, which is identical to the billing over the PKI:

ⓐ The user generates a cloud request message and sends it to the cloud service provider.

ⓑ The cloud service provider sends a micro-contract message to a user who intends to receive the resource.

ⓒ The user generates a micro-contract with digital signature by user (with two symmetric key operations) and sends it to the delegation server.

ⓓ The delegation server performs transactions for verification of the micro-contract with the cloud notary authority and the cloud service provider on behalf of the user.

ⓔ The billing process is completed on the arrival of a confirming message from the user.

4. Proposed Billing Protocol

Deliberating on our system design philosophy, we made our best effort to streamline the computation and communication overhead of the billing operation. Our novel micro-contract, which is generated by a hash function and a cross-distributed key among the entities, can facilitate billing mechanism with optimal computation and communication overheads without compromising any of the system's PKI security standards.

4.1 Flow Diagram of the Billing Protocol

Figure 4 shows a flow diagram of the proposed billing protocol. The protocol consists of three states and the protocol safety is verified in Section 4.4:

- State 1: On the first occasion when a user accesses the cloud service provider, the user and the delegation server perform a mutual authentication based on PKI. The user then delegates the user's PKI operations to the delegation server. For this purpose, a public-private key pair is generated by the delegation server and the public key is subsequently transmitted to the user. Upon the arrival of the public key, the user generates a proxy certificate containing the public key and signs the proxy certificate with the private key. Last, the user sends the proxy certificate back to the delegation server.
- State 2: Once the delegation is successfully completed, State 1 is skipped until the corresponding proxy certificate either expires or is revoked. Throughout the mutual authentication and the delegation operations, six keys are shared exclusively among the

user, the delegation server, the notary authority, and the cloud service provider as follows:

- user ↔ cloud: Session-KUser,Cloud
- notary authority ↔ user: KNotary,User
- user ↔ delegation server: KUser,Delegation
- delegation server↔notary authority: KDelegation,Notary

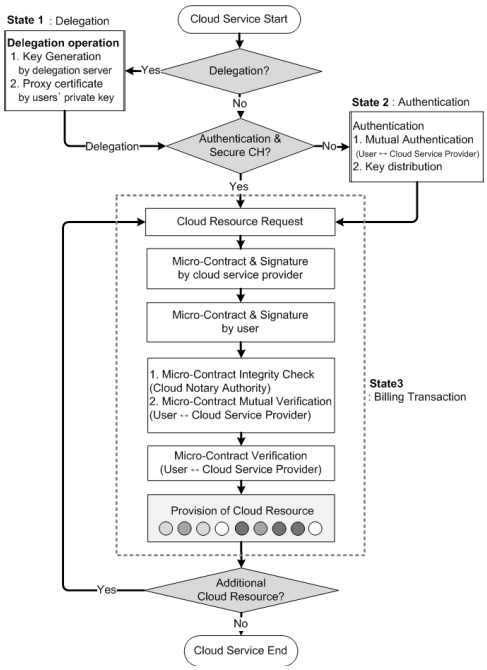


Figure 4. Flow diagram of the proposed billing protocol

- delegation server↔cloud: KDelegation,Cloud
- notary authority ↔ cloud: KNotary,Cloud

■ State 3: When a user intends to receive a cloud resource from the service provider, the user sends a resource request message and receives a micro-contract message from the service provider. The user generates a micro-contract by using the AES twice; the user then transmits the contract to the delegation server. Upon receiving the micro-contract from the user, the delegation server asks the cloud-notary-authority to verify the contract. Upon reception of the verification result from the cloud-notary-authority, the delegation server generates and sends a verified contract to the service provider and the user sends a confirmed contract to the service provider. After the validation check of the service device, the billing operation is terminated.

4.2 Description of billing protocol

As previously mentioned, the proposed protocol can provide a verifiable and non-obstructive billing operation after the delegation (State 1) and the key distribution (State 2). Any user who accesses the cloud service provider for the first time is asked to delegate the user's PKI operations. Hence, the initial billing operation takes longer than a delegated billing operation. On the other hand, the billing overhead on client and service provider can be reduced drastically after the completion of the delegation because the user and the cloud service provider can perform the billing operation by using a much simpler type of symmetric cryptography. In this section, we describe the overall transactions (State 3) of the cloud billing protocol.

Figure 5 describes the overall billing transactions, the message specification, and the notations of the entities and messages that describe the proposed protocol. When a user intends to receive a cloud resource from the service provider, generates a cloud request message (Message 1) encrypted with the Session-KUser,Cloud and sends it to the cloud service provider. On the reception of the message, the service provider transmits a granted cloud resources and micro-contract. The micro-contract contains two elements: one is the hashed value of three inputs, a unique serial number, granted resources, and a randomly generated nonce and the other one is the hashed and encrypted value

of two inputs, a unique serial number and granted resources. The serial number is updated

der and a micro-contract by user. The user the n sends Message 3 to the delegation server. Su

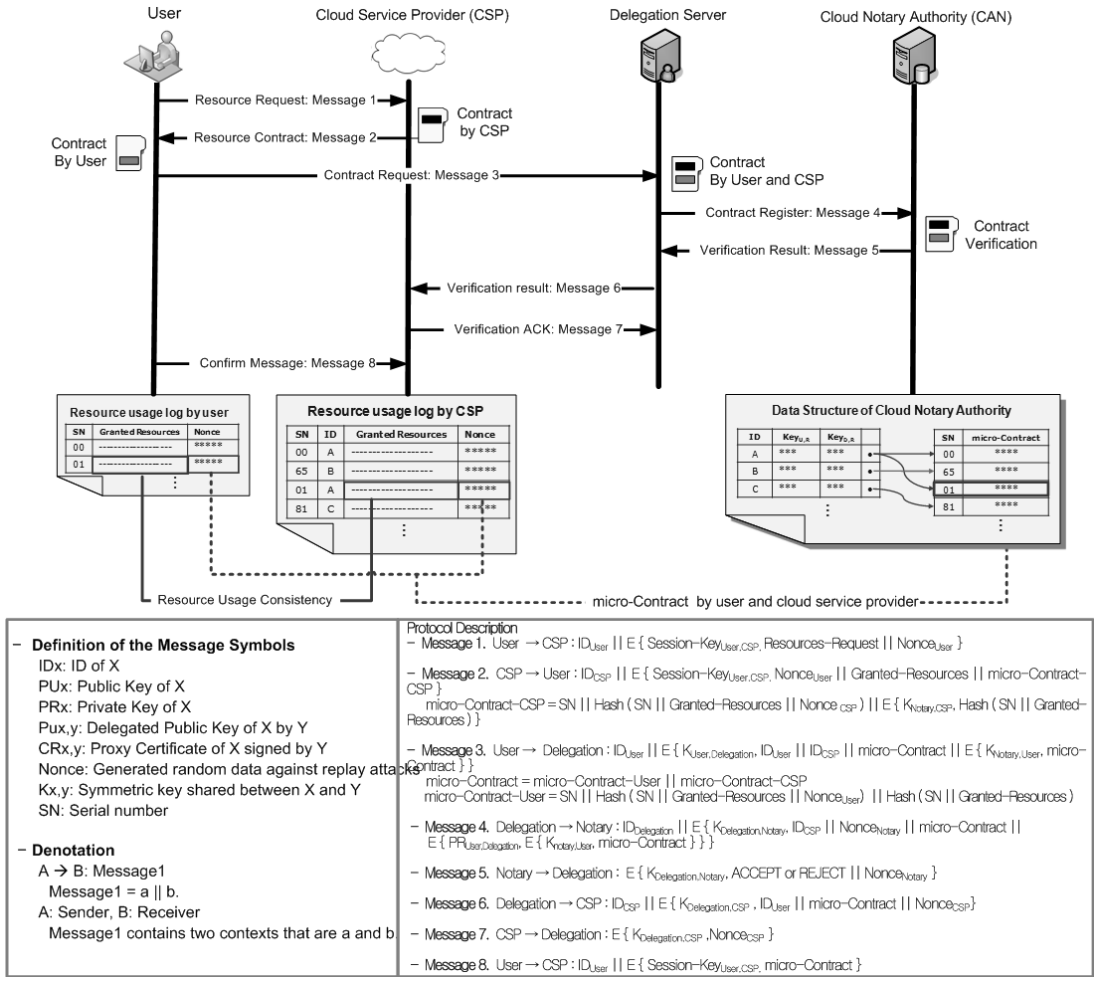


Figure 5. Overall billing transactions

for each billing transaction so that all of the serial numbers can be mapped uniquely to the user's ID and billing transactions. Furthermore, the used seed value of the generated the micro-contract is unknown to the other entities (Message 2).

Upon receiving the micro-contract, Message 2, from the service provider, the user generates a billing request message, Message 3, by combi ning both the micro-contract from service provi

bsequently, the delegation server generates Mes sage 4 as proof that the user/provider requested /assigned cloud resources and then sends this message to the notary authority for non-repudia tion. Upon the arrival of the message, the notary authority checks the validity of the resource req uest/grant message and sends the result to the delegation server in Message 4. If the delegation server receives the OK message, it sends the verification result message to the service provider

in Message 5 and Message 6. The service provider then checks the validity of the billing by using t he micro contract by the user in Message 3. Aft er that, the service provider sends the response message to the delegation server for the mutual billing in Message 7. Finally, the billing is compl eted by the user's confirm message, Message 8. As a result, the user and service provider can complete the billing transaction.

4.3 How to prove the user billing records

Our proposed protocol can provide non-repudiation and a digital signature through the micro-contract, which is generated among the entities by a hash function and distributed keys (Session-KUser,Cloud, KNotary,User, KUser,Delegation, KDelegation,Notary, KDelegation,Cloud, KNotary,Cloud). This section elaborates how to provide a verifiable billing in collaboration with the notary authority. To achieve the verifiable billing, the notary authority registers the delegation information of the user in State 1 and verifies the validity of the user's message for each billing transaction in State 3. These transactions for non-repudiation does not require any asymmetric key operations on both user and service provider; hence, it provides a level of security that is identical to that of the PKI, as well as minimized billing operation overhead.

Over the verification phase of the cloud-notary -authority (Message 4, Message 5), the notary authority retains the history data, which states that the service provider sent Message 2 and the user sent Message 3, as a type of notarized billing list (NBL). The NBL is the data structure for storing the evidence for the billing transactions, and all of the contexts are periodically stored with the signature of the notary authority. This process ensures the integrity of the billing operation by using the digital signature of the notary authority. Figure 6 illustrates how the NBL is used to prove that the user and the service provider did perform a billing transaction. Let's assume that the cloud service provider asserts that the user repudiates a billing with the cloud service provider. In this case, the service

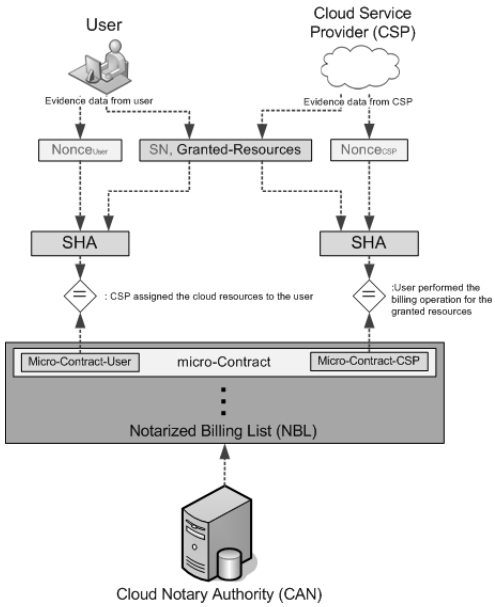


Figure 6. Verification mechanism using NBL

provider can put in a claim for a justice with the serial number included in the micro contract to the notary authority. The notary authority then requires the evidentiary data that were used to generate the micro contract message with the serial number and inputs the evidence data and the serial number to SHA-1. If the output of SHA-1 is identical with the stored data in the notary authority, it proves that the user forwarded the received the micro contract for the accused billing. The notary authority can therefore refute the user's repudiation.

4.4 Security Proof

We analyzed the protocol safety by considering the replay attacks and man-in-the-middle (MI

TM) attacks. We assumed that the underlying cryptography (AES, RSA, SHA) were invulnerable with regard to message secrecy and integrity; hence, we did not consider attacks such as cryp tanalysis and message slicing. One the other hand, any principle can place or inject a message on any link at any time. In addition, any principle can see, delete, alter, and redirect all exchanged me ssages being passed along any link or replay me ssages recorded from past communications. In this section, we confirm the safety of our proposed protocol in relation to replay attacks and MITM attacks.

Theorem 1. the proposed billing system is safe from replay attacks.

Proof: Let' s assume that the billing path is [Alice↔Delegation Server↔Notary Authority↔De legation Server↔Service Provider]. We prove the safety for the next attack types as follows:

1) A replay attack for the resource request message (Message 1) and the resource contract messages (Message 2 and Message3)

2) A replay attack for the verification messages (Message 4, Message 5 and Message 6) and v erification ACK message (Message 7)

3) A replay attack for the billing confirm message between the user and the cloud service provider (Message 8)

- In case of 1), an intruder can try to attack by sending the captured messages. However, the intruder' s attack cannot succeed because the messages include the nonce data (NonceUser) and micro-Contract which are altered in each billing transaction.

- In case of 2), an intruder can try to attack by sending the captured messages. However, the intruder' s attack cannot succeed because the m essages include the nonce data (NonceNotary and NonceCSP).

- In case of 3), an intruder cannot reuse the confirm message because the micro-Contract in cluded in the resource request message (Messa ge 3) is altered in each billing transaction. □

Theorem 2. the proposed billing is safe from MITM attacks.

Proof: We prove the safety for the next attack types as follows:

1) An MITM attack between a user and a delegation server

2) An MITM attack between a user and a service provider

3) An MITM attack between a delegation server and a service provider

- In case of 1), each entity performs a mutua l authentication over the PKI before a delegation and shares the key for a secure connection. Thus, the intruder cannot forge the billing request mes sage of the user.

- In cases of 2) and 3), the service provider, user, and the delegation server generate billing transaction messages, which are altered for each billing transaction and the messages are encrypted and transmitted by using the shared keys that are paired in a previous state. Thus, the intruder cannot successfully masquerade as the user or the service device.

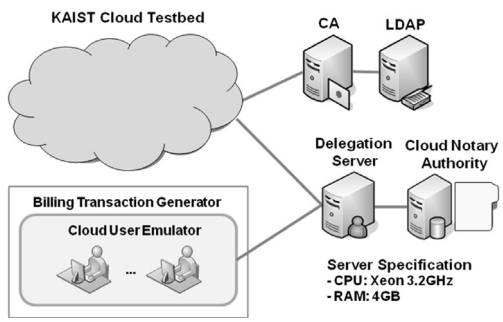


Figure 7. Experiment environment to measure the performance

5. Efficiency and Performance Evaluation

In this section, we present the performance results obtained with our prototype implementation. First, we demonstrate the overall experimental environment. We then describe the operational efficiency of the billing protocol to evaluate the performance in terms of billing overhead.

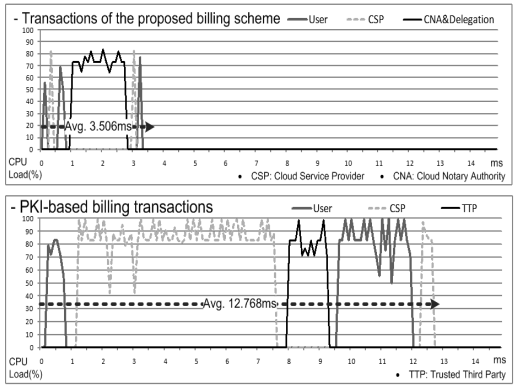
5.1 Experimental Environment

To evaluate the performance characteristics, we constructed a cloud user emulator, which are coupled to a billing transaction generator. The objective of the emulator is to simulate the processing and communications resources anticipated in a full implementation. Figure 7 shows the overall experimental environment. The emulator has operation times that are similar to actual operation times (for example, those that result from communications, billing operation, and message processing). Moreover, the user emulator is connected to our billing system and receives control signals from the billing transaction generator. The generator is a module that generates control signals to produce billing request messages by using a random generator that models the computer usage pattern of users [22].

5.2 Billing Protocol Efficiency

The performance of the billing protocol in terms of billing overhead and the consumption of processing and communication resources is an important factor to be considered when designing billing protocols. First, we analyze the efficiency in comparison with PKI in terms of computation and communication efficiency.

Figure 8 gives the number of public and private keys (RSA 1024bit) and the symmetric key (AES 128bit) operations performed with total operation time for each billing protocols. In spite of having the smaller number of cryptography operations per billing, the PKI-based billing protocol has about four time longer operation time than our billing system because it has a larger number of private/public key operations on both the client and service provider side. In the case of



System	User			Cloud Service Provider (CSP)			Trusted Third Party (CNA & Delegation)			Total Transaction Time (Computation + Communication)
	RSA 1024bit		AES	RSA 1024bit		AES	RSA 1024bit		AES	
	Public Key	Private Key		Public Key	Private Key		Public Key	Private Key		
PKI	2	1	2	1	2	2	1	0	0	Avg. 12.768 ms
	Avg. 5.402 ms			Avg. 6.357 ms			Avg. 0.429 ms			
Proposed	0	0	5	0	0	5	1	1	6	Avg. 3.506 ms
	Avg. 0.282 ms			Avg. 0.173 ms			Avg. 2.751 ms			
System Spec.	CPU: Pentium IV2.4GHz RAM : 2GB			CPU: Intel Xeon 3.2GHz RAM : 4GB			CPU: Intel Xeon 3.2GHz RAM 4GB			

Figure 8. Billing transactions and total transaction time

our billing system, the first time a user accesses to the cloud service provider, the user delegates the user billing operations to the delegation server (State 1), which requires two public key operations and one private key operation. The delegation operation time of our system is similar to operation time of the PKI-based billing. After the delegation operation, the user can perform a billing operation by processing only five symmetric key operations; this process has much shorter operation time than the corresponding process of the PKI-based billing transaction. By way of summarizing the above results, we give an ou

tline of the overall transaction time of the billing protocols. The PKI-based billing transaction tim e with the RSA 1024 bit algorithm is 12.768 m s. In the case of our billing system, the billing t ransaction can be accomplished without asymmet ric key operations after the delegation operation. As a result, we confirmed that our system has a much shorter operation time, even though it provides a PKI-based billing.

In order to measure how many cryptography cause obstructive billing overhead, we estimated the operation time of each entities for each billing transactions as shown in Figure 8. Even though the operation time of our billing system of the trusted third party side is longer than that of the PKI-based billing system, the total transaction time of user and service provider side is much shorter than that of the PKI-based billing system. In a result, the total billing transaction time of our billing system (3.506 ms) is much shorter than the PKI-based billing transaction time (12.768 ms). It is due to the gap in computing complexity of user and service provider side.

5.3 Throughput Evaluation

Figure 9 illustrates how the throughput of the billing transactions mutates as the number of billing requests per second varies. The number of billing requests per second ranges from 10 to 300. For the PKI-based billing protocol, we found that the throughput is saturated on 61.5 transactions per second as the number of billing requests increases, mainly as a result of the cryptography operations and the communication overhead on both client and server side. In the case of our system, the throughput is saturated on 223.4 transactions per second as the number of billing requests increases because the user/service provider-side overhead of our system was much smaller than that of

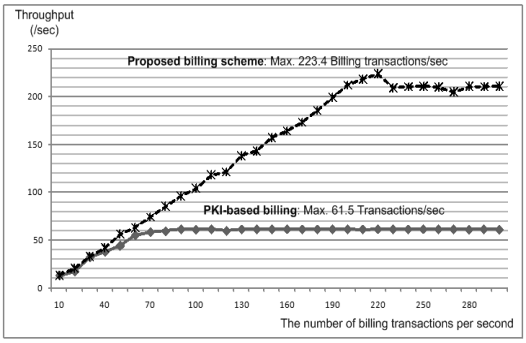


Figure 9. Throughput of the billing transactions with varying numbers of billing transactions per second

PKI-based billing overhead. This phenomenon is due to the fact that quantity of user and server provider-side operations of our system is much smaller than that of PKI-based billing. This result confirms that our billing protocol with a fine-grained verifiability can be accomplished seamlessly whenever the number of requests per second is less than 200.

6. CONCLUSION

Our task was to provide a full-fledged verifiable and non-obstructive billing solution tailored for a cloud computing environment. To accomplish this task, we thoroughly reviewed the ways in which the coventional billing system is used in t he environment, and we consequently derived the blueprints for a secure and efficient PKI-ba sed billing system. Besides utilizing conventional PKI entities in our billing system, we conceived and implemented the concepts Cloud-Notary-Au thority to supervise billing to make it more objective and accepted by both cloud users and providers. Our billing system features two remarkable achievements:

1) Billing transactions with minuscule computing overhead: the delegation server is responsible for

performing prohibitively expensive PKI operations on behalf of a user without compromising the security level of the PKI; as a result, it minimizes the computational overhead of the cloud service user-side.

2) Cloud Notary Authority: this entity ensures undeniable verification of any transaction between a cloud service user and a cloud service provider. Our billing system consequently enables a cost-effective but uncompromisingly secure developm ent of a billing system. Furthermore, our delegation mechanism significantly reduced the billing overh ead. According to the performance evaluation, the billing overhead of our billing system (which averages 3.506 ms) is much shorter than conve ntional PKI-based billing overhead (which avera ges 12.768 ms) so that the throughput of the billing transactions (223.4 transactions per seco nd) is much higher than the throughput of the PKI-based billing transactions (which averages 61.5 transactions per second).

7. REFERENCES

[1] J. Ainsworth, J. MacLaren, and J. Brooke, "Implementing a secure, service oriented acc ounting system for computational economies", IEEE International Symposium on CCGrid, vol. 2, 2005, pp. 654-660 Vol. 2.

[2] H.-Y. Lin, and L. Ham, "Authentication pro tocols with non-repudiation services in pers onal communication system", IEEE Commun ications Letters 3 (8). 1999.

[3] Regev O, Nisan N., "The POPCORN marke t: an online market for computational resour ces", Proceedings of the 1st International Conference on Information and Computation Economies (ICE ' 98), ACM Press: New York, NY 148-157.

[4] Buyya R, Abramson D, Giddy J, Stockinger H. Economic models for resource management and scheduling in Grid computing. Concurrency and Computation: Practice and Experience 20 02; 14(13-15):1507-1542.

[5] Chun BN, Culler DE. Market-based proporti onal resource sharing for clusters. Technical Report CSD-1092, Computer Science Division, UC at Berkeley, January 2000.

[6] R Byrom , "APEL: An implementation of Grid accounting using R-GMA, et al.", Proceedings of the 2005 UK e-Science All Hands Meeti ng, September 2005

[7] M. Litzkow, M. Livny, and M. Mutka, "Condor: A hunter of idle workstations," presented at the 8th Int. Conf. Distributed Computing Sys tems (ICDCS 1988), San Jose, CA, Jan. 19 88.

[8] Buyya R, Abramson D, Giddy J., "Nimrod/ G: An architecture for a resource manageme nt and scheduling system in a global Comput ational Grid", Proc. of the 4th International Conference on HPC, May 2000.

[9] Kwon, O., Hahm, J., Kim, S., and Lee, J. 20 04., "GRASP: A Grid Resource Allocation System based on OGSA", In Proc. of the 1 3th IEEE international Symposium on High P erformance Distributed Computing.

[10] IBM, "Tivoli: Usage and Accounting Mana ger", IBM Press 2009

[11] Guarise, A., Piro, R., and Werbrouck, A. "DataGrid Accounting System - Architecture", EU DataGrid, 2003.

[12] P. Gardfj¨all, E. Elmroth, L. Johnsson, O. Mulmo, and T. Sandholm. Scalable grid-wi de capacity allocation with the SweGrid Ac counting System (SGAS). Concurrency Co mputat.: Pract. Exper., 20(18):2089-2122, 2008.

[13] A. Barmouta and R. Buyya, "GridBank: A Grid Accounting Services Architecture (GASA) for Distributed Systems Sharing

and Integration", Proceedings of the 17th IEEE IPDPS 2003, April,2003, pp. 22-26.

[14] Amazon Web Services, "Amazon Elastic Compute Cloud", http://aws.amazon.com/ec2/

[15] Thompson, M., Johnston, W., Mudumbai, S., Hoo, G., Jackson, K., and Essiari, "Certificate-based access control for widely distributed resources". In Proceedings of the 8th Conference on USENIX Security Symposium

[16] Von Welchr, Ian Foster, Carl Kesselman, Olle Mulmo, Laura Pearlman, Steven Tuecke, Jarek Gawor, Sam Meder and Frank Siebenlist "X.509 Proxy Certificates for Dynamic Delegation", In Annual PKI R&D workshop, April,2004

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

[18] S. Tuecke, V. Welch, D. Engert, L. Pearlman, and M. Thompson, RFC 3820: Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile. IETF Network Working Group, 2004.

[19] Federal Information Processing Standards Publication 197 "ADVANCED ENCRYPTION STANDARD (AES)", NIST Press 2001

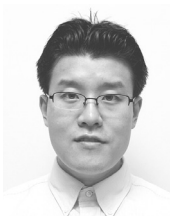
[20] J. Jonsson and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", Network Group

[21] E. Roberts, M. Dahan, J. Boisseau. TeraGrid User Portal: An Integrated Interface for TeraGrid User Information and Services TeraGrid 2008

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

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

◆ Ki-Woong Park



He received the BS degree in computer science from Yonsei University in 2005 and the MS degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST) in 2007.

He is currently working toward the PhD degree in the Division of Electrical Engineering at KAIST. He joined the Microsoft Research and he was awarded the Microsoft Fellowship for the period of 2008-2009. His research interests include security protocol, network security, and embedded systems. He is a student member of the KING, the IEEE and the IEEE Computer Society.

◆ Sung-Kyu Park

He received the BS and the MS degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST) in 2007. He is currently working toward the PhD degree in the Division of Electrical Engineering at KAIST. His research interests include flash memory control and file systems. He is a student member of the KING.

◆ Kyu Ho Park

He received the BS degree in electronics engineering from Seoul National University in 1973, the MS degree in electrical engineering from the Korea Advanced Institute of Science and Technology(KAIST) in 1975, and the DrIng degree in electrical engineering from the University of Paris XI in 1983. Since 1983, he has been a professor in the Division of Electrical Engineering and Computer Science at KAIST. From 2005 to 2006, he was the president of the Korea Institute of Next-Generation

Computing. His research interests include computer architectures, file systems, storage systems, ubiquitous computing, and parallel processing. He is a member of KISS, KITE, the Korea Institute of Next-Generation Computing, the IEEE, and the ACM.