Task-Oriented Access Model for Secure Data Sharing Over Cloud

Mahmood Ahmad
Department of Computer Engineering
Kyung Hee University, South Korea
rayemahmood@oslab.khu.ac.kr

Zeeshan Pervez
School of Computing
University of the West of Scotland
PA1 2BE, UK
zeeshan.pervez@uws.ac.uk

Byeong Ho Kang
University of Tasmania
Australia
Byeong.Kang@utas.edu.au

Sungyoung Lee
Department of Computer Engineering
Kyung Hee University, South Korea
sylee@oslab.khu.ac.kr

ABSTRACT

Cloud computing has become a prevalent technology and with its increased maturity more and more data including sensitive and non sensitive, is being centralized into it . While outsourcing the sensitive data into public cloud, its prior encryption is strongly recommended. Provisioning of encryption and existing work that guarantee security and privacy concerns on sensitive data, have removed the holdouts against cloud adoption at a large. One of the main issue with this data in cloud environment is to manage user access and its auto revocation in a controlled and flexible way. The issue becomes more complex when privacy on user access has to be ensured as well to hide additional leakage of information. For automatic revocation over cloud data, access can be bounded within certain anticipated time limit so that the access expires beyond effective time period as proposed by one of the existing system as time based proxy re-encryption. This time-oriented approach is more rigid and not a one-size-fits-all solution. In certain circumstances exact time anticipation is not an easy choice. Instead, the alternate solution could be task-oriented to restrict user beyond certain number of permissible attempts to access the data. In this paper we have proposed a system that allows authorized users to access encrypted data for predefined attempts rather pre-defined time. Our approach allows user to avail permissible attempts without time restriction and at the same time also preserves the privacy aspect of user access by concealing access limit until availed.Performance analysis revealed that the cost of operations performed are within the range of `.097` to `.278 $ per 1000` requests.

General Terms

Security and Privacy

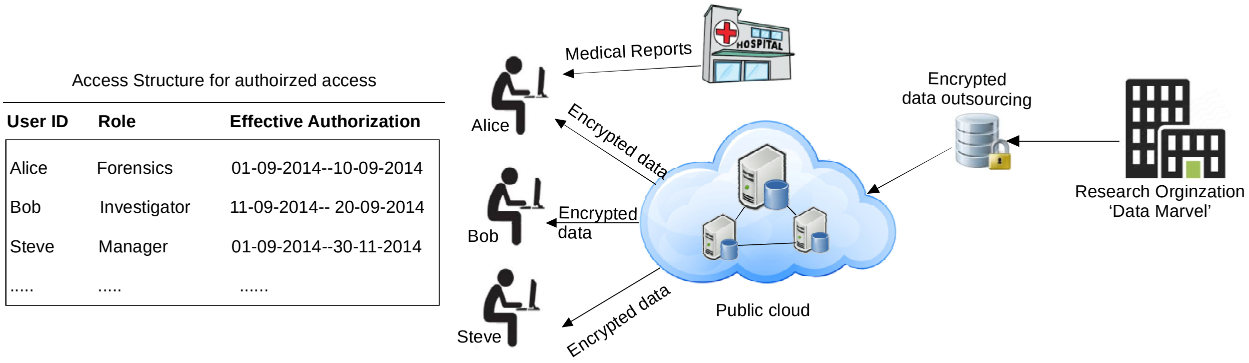
Keywords

Security, Privacy, Cloud, Access Policies

1. INTRODUCTION

The prevalent technology of cloud computing has become an increasing commercial trend where companies enjoy the on demand services under the pay-as-you-use subscription model. These cost effective and high quality data storage facilities in shape of public clouds have relieved data owner from the burden of complex data management and maintenance issues [12]. As cloud computing technology is becoming more mature, data outsourcing is also gaining momentum. With this realization, sensitive data including health records, financial data and personal files is also being centralized to avail a range of cloud services 24/7, however; to ensure privacy and security for this sensitive data, encrypted outsourcing is strongly recommended [6]. Encryption secures the unsolicited disclosure of information not only from unauthorized users but also from cloud service provider (CSP) and only users with valid decryption keys have the privilege to recover the data. with these desirable features, holdouts against cloud adoption for sensitive data are rapidly diminishing.

The encryption process resolves the issue of data security at first place, but it introduces new problems such as fine grained access control, user revocation and additional leakage of information. Considering the paper scope and our proposed solution, discussion focus will be on user access and its auto revocation whereas other issues will be addressed very briefly. To formally represent these issues, let us suppose that $\mathcal{D}$ represents sensitive data, $\mathcal{E}_s$ is an encryption scheme that encrypts $\mathcal{D}$ using key k i.e., $\mathcal{D} \xrightarrow{\mathcal{E}_s(\mathcal{D},k)} \mathcal{D}^k$. After encryption, $\mathcal{D}^k$ is outsourced in public cloud by data owner and shared with users $\langle u_1^k, u_2^k, \ldots, u_n^k \rangle$ where u_i^k, rep-



Access Structure for authoirzed access

User ID	Role	Effective Authorization
Alice	Forensics	01-09-2014--10-09-2014
Bob	Investigator	11-09-2014-- 20-09-2014
Steve	Manager	01-09-2014--30-11-2014
.....	.....	......

Figure 1: Encrypted Outsourcing of data on public cloud and access structure

resents an authorized user having key k. Usually authorization over encrypted data is not granted for lifetime and at some stage it is required to be revoked after certain *condition* holds true. For explanation let us consider that for a particular user this *condition* is imposed such that after time t the user access is required to be revoked. As a naive approach, the data owner will encrypt $\mathcal{D}$ with a new key k' i.e., $\mathcal{D}^{k'}$ and distribute the new key with remaining users $\left\langle u_1^{k'}, u_2^{k'}, \dots, u_{n-1}^{k'} \right\rangle$ so that key k become ineffective for the revoked user. If user revocation is a frequent activity then executing this process makes presence of data owner mandatory which is under utilization of cloud resources. In this situation this process needs to be delegated with cloud server so that a user access becomes ineffective whenever required without involving the data owner. With this realization, the work presented as ticket-based access control [13] tries to resolve the similar issue. It requires users to expose their tickets to the CSP with each request, however; it also exposes the effective time of each ticket. Exposing time validity can help CSP to cease user access automatically beyond her limit without involving the data owner, but it reveals the access limit of each user. The additional information that leaks this way can significantly reduce user trust and privacy of overall system. Encryption itself is not the sole guarantee for overall system trust, certain requirements from user side also need to be addressed accordingly. For example, a user having access rights for longer duration can reveal her importance over a user having access rights for shorter duration. Shifting the task of user revocation on cloud needs to be designed in such a way that it should also resist additional leakage of information. For automatic revocation that is designed to revoke a user access after predetermined period of time has been proposed by Qin Liu et. al. [8]. Their design focuses on auto revocation (time oriented access) with minimal leakage of information.

Additional leakage of information is one important factor to be considered while managing user access over encrypted data. Other than hiding user access period over encrypted data, user identities are also concealed. Instead of using user IDs, their attributes are used instead. For this purpose Ciphertex-policy attribute-based encryption (CP-ABE) [2], [9] as a promising branch of ABE [14][23] has such a property. In CP-ABE, users are identified by a set of attributes rather than an exact identity. For each eligible attribute, the user will be issued a user attribute secret key (UAK). Each data is encrypted with an attribute-based access structure, such that only the users whose attributes satisfy the access structure can decrypt the ciphertext using their UAKs. Avoiding additional leakage of information adds more towards user management over encrypted data.

To illustrate auto revocation of user access together with minimal leakage of additional information, let us consider the application scenario give in Figure 1. A research organization 'DataMarvel' has outsourced its encrypted data on a public cloud. This data includes sensitive and valuable information that can assist various law enforcement agencies to minimize the metropolitan crime rate. Recently one of the agency is solving a criminal case for which Alice, Bob and Steve have been given authorized access over encrypted data. The effective authorization time for each user is shown in the Figure 1. Alice is working on the Forensics and her access is valid for 10 days from 01-09-2014 to 10-09-2014. She also needs few medical reports from a nearby hospital. Bob authorization is from 11-09-2014 to 20-09-2014 as his part of investigation starts after Alice has finished her findings. Steve as a manager is supervising the whole activity and is responsible to compile final report, his effective authorization is from 01-09-2014 to 30-09-2014. Complete task is designed to be finished within 30 days. If this application scenario is modeled on time-based proxy re-encryption as proposed in [8], user revocation is possible to take place at predetermined period of time. Ideally each user is expected to finish assigned task within her/his effective time period. As we can see that Alice work is also dependent on medical reports that she is expecting from an external entity (in this case, hospital). If reports from hospital arrive after her effective authorization time, she cannot compile her work and needs extension. Obviously, effective authorization for other users will also need extension accordingly. In this situation where mutual dependency exist on external factors, time oriented access structure need to be replaced with more flexible approach. One way to achieve this flexibility is to replace time oriented authorization with task orientated authorization. In task oriented authorization a user authorization will remain effective for predefined number of access attempts for which authorization is granted. In this example shown in Figure 1, suppose Alice is authorized to access the encrypted data for 5 [1] times then she can consult the encrypted data with more flexibility without worrying about time restriction. With these realizations, we have proposed a task oriented access model in which access expires when user has availed her *effective authorization*. We define *effective authorization* as "number of times a user can access the authorized resources" i.e., *authorized quota*. Being independent from time restriction, our model is more flexible for user to avail her authorization. With our proposed design, we made following contributions.

- User authorization model is flexible as it is not time dependent.
- The process of user revocation does not involve data-owner.
- User concerns to preserve her access rights to be known in advance are well preserved.
- User access limit remains unknown for the whole authorized duration.

The rest of the paper is organized as follows. Section 2 presents the related work. Technical preliminaries and definitions are given in Section 3. Section 4 covers general overview and assumptions. System setup is given in Section 5. System implementation and results are given in 6. Paper concludes in Section 8.

2. RELATED WORK

We assume that user management starts with her authorization and ends with revocation over cloud data. To incapacitate any single out of N users on encrypted data, there exist few approaches. The first one, which is a naive solution is to re-encrypt the complete data and re-distribute the new decryption keys to $(N-1)$ users. This approach is quite computationally intensive when frequency of users entering and leaving the system is very high. In addition, data owner has to be online all the time to execute this operation, which is difficult to maintain 24/7. Relaxing responsibilities for data owner can be achieved using Proxy Re-Encryption (PRE) [1, 3] either through TTP CSP. PRE converts a cipher text that can be decrypted by Alice into another cipher text that can be decrypted by Bob. This whole operation hides the actual data being transformed from one key to another. Work done by [17] is considered a pioneer to combine Key-Policy ABE (KP-ABE) and PRE to delegate most of the computation tasks involved in user revocation to the CSP. Still, activating PRE by CSP awaits for an event to trigger, which is again responsibility of data owner. To handle this issue, activation of PRE has been coupled with time in such a way that user authorization expires on predetermined time [8]. This approach also conceals user effective time period from CSP to know in advance. Allowing CSP to know about user effective time period on cloud data can reveal user importance who has permission for longer time than other users.

The outsourced data in a public cloud has its own importance and value. Employing encryption on this trove of information is mostly a favorable choice by the data owner. On the other hand, hiding access patterns is more desirable and appreciated by users who will use this data. In recent research, which is cited in upcoming discussion is related to conceal user access and possible leakage of information on her behalf. Instead of giving exact IDs to authorized users, certain descriptive attributes are more suitable for identification. If Alice who is a manager in a company, instead of using Alice as her ID, the attribute of 'manager' is preferred. The same idea has been chosen in (CP-ABE) [2], [9] by employing the Ciphertext-policy attribute-based encryption (CP-ABE) Using this technique, identities of users can be concealed by using feature of attributes instead of exact IDs.

User revocation is a complex process and very few techniques have been used to handle this issue in cloud model. In [8] the idea of time based proxy re-encryption has been employed. In this work, usage of CP-ABE is extended with HABE [15, 16] by using the concept of time to trigger automatic proxy-re encryption. In this approach the granularity of time has been sliced into three layers, namely: year, month and day. This solution is prepared for situation where time anticipation can be determined in advance before a user is granted access over cloud data. Revoking user access rights using this methodology is appealing for situations where time anticipation is a trivial. In task oriented situations this solution will be least effective.

Besides revoking a user from subsequent request on cloud data, it is also very important to determine the mechanism with which CSP will evaluate and come to know that further access has to be ceased. The simple solution is to let CSP know about effective time of all users in advance.If longer access duration implies user importance then it is open to CSP with this naive solution. The work in [13], which require users to expose their tickets to CSP, may also expose effective time of each ticket. This approach might also be interesting to know for a curious CSP when two competitors are assigned on a same resource where authorization of one user is higher than the other. For users who consider this information as their business secret would avoid to go with this solution or any other similar to it. In this paper we have considered two issues, which are subset of above discussion. First, automatic user revocation and that too independent from time. Second, hiding user authorization attempts throughout her communication with cloud. For CSP, these attempts would remain unknown to predict and hard to distinguish between any number of users. In our design we have utilized the cryptographic primitives of homomorphic encryption[10] and private matching [4] to meet the desired results.

3. TECHNICAL PRELIMINARIES

In this section we will present the existing standards used during the system construction.

3.1 Homomorphic encryption

Homomorphic encryption *HE* is a form of encryption where a specific algebraic operation performed on the plaintext is equivalent to another (possibly different) algebraic operation performed on the ciphertext. An encryption scheme is said to be additive homomorphic if and only if

$$E_H(m_1) \odot E_H(m_2) = E_H(m_1 + m_2)$$

where $\odot$ is an operator. Pascal Paillier cryptosystem [10]

[1] This number is also referred as 'authorized quota'

possesses the property of additive HE which is as follows.

- Key generation: Let $N = pq$ be the RSA-modulus and g be an integer of order αN module N^2 for some integer α. The public key is (N, g) and the private key is $\lambda(N) = lcm((p-1)(q-1))$.
- Encryption: The encryption of message $m \in Z_N$ is $E_h(m) = g^m r^N \bmod modN^2$ where $r \in_R Z_N^*$
- Decryption: For ciphertext c, the message is computed from $$m = \frac{L(c^{\lambda(N)} modN^2)}{L(g^{\lambda(N)} modN^2)}$$

A scheme is said to be multiplicative homomorphic if and only if
$$E_H(m_1) \odot E_H(m_2) = E_H(m_1 \times m_2)$$
The Goldwasser-Micali (GM) cryptosystem is a semantically-secure scheme based on the quadratic residuosity problem. It has XOR homomorphic properties, in the sense that $E_H(b).E_H(b') = E(b \oplus b') modN$ where b and b' are bits and N is the public key. A homomorphic encryption is said to be semantically secure if $E(H)$ reveals no information about m_1 and m_2, hence it is computationally infeasible to distinguish between the cases $m_1 = m_2$ and $m_1 \neq m_2$ [11]

4. GENERAL OVERVIEW

In this paper we present a protocol for task oriented user access over encrypted data that is outsourced in the cloud. System initialization begins with encrypted outsourcing and credentials distribution amongst involved entities. Cloud service provider, trusted third party, data owner and authorized user are involved entities of the system which we will refer as CSP, TTP, owner and user respectively. Sensitive data is outsourced to CSP which is owned by the owner. Services of TTP are used to transform user request into partial semi-processed oblivious request before it is submitted to CSP. After the user request is evaluated obliviously, results are sent back to the user. Before explaining system initialization and user request evaluation, first we present the assumptions and notations used for the system.

4.1 Assumptions

We assume that there is no fully trusted entity in the environment and all the entities are semi-honest. Semi-honest entities behave honestly, but try to extract information beyond permitted. We also utilize services of a TTP that follows the protocol as defined and does not team up with the CSP. The protocol is designed in such a way that TTP cannot discover the authorized quota of a user access in advance, however; the only information revealed at TTP is about validity of user request i.e, valid or otherwise. TTP learns about the expiration of user requests when user has already availed her access quota. We also assume that during the protocol execution, CSP behaves honestly and perform a constant operation for each user request.

5. SYSTEM SETUP

To explain the system setup, let us consider that DataMarval is a large research organization (shown in Figure 1) responsible to assist law enforcement agencies at various levels. The variety of data and analytical reports managed by DataMarval assist significantly to lower down the crime rate. Considering the volume of data and to ensure its high availability, DataMarval is using services of a public cloud owned by *Eve.* To ensure privacy of data files $\mathcal{F}$, DataMarval encrypts $\mathcal{F}$ before outsourcing on *Eve's* cloud.

Table 1: Assumptions and notations used in the descriptive detail

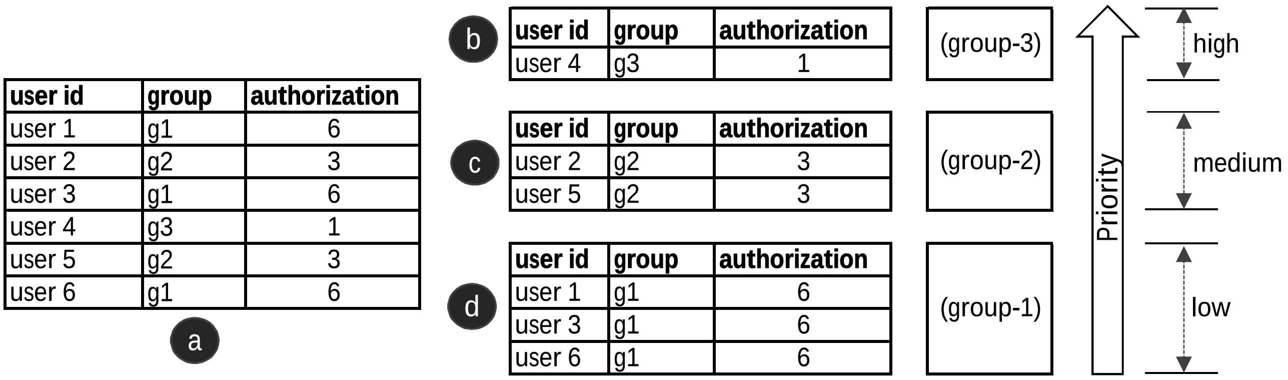
Notations	Description
$\mathcal{F}$	Data files to be outsourced
α_i	User authorized attempts over $\mathcal{F}$
λ_i	Offset value for user
$\mathcal{P}(.)$	A finite degree Polynomial created for authorized user
Θ	Lowest degree coefficient in $\mathcal{P}(.)$
$\mathcal{C}$	A Constant, obtained from $\mathcal{P}(.)$
Λ	With each user request, TTP calculates encrypted values
Δ	Oblivious Value calculated by CSP for each user request
$\mathcal{E}_s$, $\mathcal{D}_s$	Symmetric encryption and decryption algorithms
$\mathcal{E}_H$, $\mathcal{D}_H$	Homomorphic encryption and decryption algorithms
σ_{pk}, σ_{sk}	Public and secret key pair for Homomorphic encryption
k	Secret key of symmetric key algorithm

Recently, a law enforcement agency 'A' has realized that these services can significantly help its ongoing investigations. For this purpose, 'A' needs these services for her employees [2]. Hierarchical grouping of these employees is done with respect to number of authorized access. The Figure 2 presents six users assigned with three groups i.e., Figure 2-(b), Figure 2-(c) and Figure-2-(d). The task nature needs user 1,3 and 6 to finish their report within limited number of access authorization i.e., six, Figure 2-(d) . Similarity user 2 and 5 are given access limit 3, Figure-2-(c). User 4 is given access for one time only, Figure 2-b. Here the task distribution creates dependently amongst them i.e, user 2 and 5 are dependent on user 1,3 and 6 whereas user 4 is dependent on user 2 and 5. Due to the mutual dependency of groups, its not possible to anticipate the effective time slot for any group therefore the agency 'A' opts the task oriented approach where each user can consult services in their own time. Also the model avoids additional leakage of information in shape of authorized quota for each user. With this flexible model the access quota of each user can be kept secured until it is utilized fully. The proposed model also terminates user access automatically once authorization needs to be revoked without involving the data owner.

5.1 Initialization

DataMarvel (DM) outsources $\mathcal{F}$ on a public cloud owned by Eve. Details on secure outsourcing have been neglected intentionally, however; reader may refer to [17] for efficient and secure data sharing in cloud storage. The task oriented access model for a specific user formulates the access model after getting α_i. To create $\mathcal{P}(.)$ for the user, an offset value λ_i is generated such that its roots are generated from λ_i and $\lambda_i + \alpha_i$. A constant factor $\mathcal{C}$ obtained from $\mathcal{P}(.)$ is divided

[2] Employees will also be referred as users



user id	group	authorization
user 1	g1	6
user 2	g2	3
user 3	g1	6
user 4	g3	1
user 5	g2	3
user 6	g1	6

user id	group	authorization
user 4	g3	1

user id	group	authorization
user 2	g2	3
user 5	g2	3

user id	group	authorization
user 1	g1	6
user 3	g1	6
user 6	g1	6

Figure 2: Grouping and authorization scheme for trainees

into two random parts c_1 and c_2 such that $\mathcal{C} = c_1 + c_2$. A unique user id $\mho_i \in \mathbb{N}$ is also created that will be used as an echo effect and to uniquely identify the user. The details of echo effect is given in section 5.2. User is given with decryption key k and $\mho$. Highest degree coefficient from $\mathcal{P}(.)$, c_1,λ_i, and $\mho$ are given to TTP along with σ_{pk} and σ_{sk}. CSP is given with $\mathcal{E}_H(c_2, \sigma_{sk})$, least coefficient from $\mathcal{P}(.)$ encrypted under σ_{sk} and σ_{pk}. After this distribution user can now avail her authorized services when required.

5.2 User request evaluation

This section covers the process involved for user request at first time, subsequent requests and revocation of user authorization through echo effect.

For the request appearing first time, TTP picks up the λ and encrypt it using σ_{sk} i.e., $\mathcal{E}_H(\lambda, \sigma_{sk}) = \lambda^{\sigma_{sk}}$. Similarly, TTP calculates $\mathcal{E}_H(\left(\lambda^2 + c_1\right), \sigma_{sk}) = \left(\lambda^2 + c_1\right)^{\sigma_{sk}}$ and sends these encrypted values to CSP. Λ represents these encrypted values, as given in equation 1

$$\Lambda = \left(\lambda^2 + c_1\right)^{\sigma_{sk}} \tag{1}$$

After receiving Λ from TTP, CSP calculates $(\Theta \otimes^{\sigma_{pk}} c_2^{\sigma_{sk}})$ and $(\lambda^{\sigma_{sk}} \otimes^{\sigma_{pk}} \lambda^{\sigma_{sk}})$, where $\otimes^{\sigma_{pk}}$ represents the homomorphic multiplication given σ_{pk}. The value set of Θ, $c_2^{\sigma_{pk}}$, and σ_{pk} have already been communicated to CSP during the system initiation. Using σ_{pk}, CSP executes the additive homomorhpic operation $\oplus^{\sigma_{pk}}$ on these values to obtain a final oblivious vector Δ as given in equation 2.

$$\Delta = \Lambda \oplus^{\sigma_{pk}} (\Theta \otimes^{\sigma_{pk}} c_2^{\sigma_{sk}}) \oplus^{\sigma_{pk}} (\lambda^{\sigma_{sk}} \otimes^{\sigma_{pk}} \lambda^{\sigma_{sk}}) \tag{2}$$

The response of request is sent back to user through TTP where Δ is shared with TTP only. Upon receiving value of Δ, TTP decrypts it using σ_{sk} as given in equation .

$$\mathcal{D}_{\mathcal{H}}(\Delta, \sigma_{sk}) = \Phi_x \tag{3}$$

$$\Phi_x = \begin{cases} \Phi_{echo} & = \quad \textit{First and last request only} \\ \Phi_{residual} & = \quad Except\ first\ and\ last\ request \end{cases} \tag{4}$$

For the user request appearing first time, TTP stores output from equation 3 as Φ_{echo}. For each subsequent request, value recovered as Φ_x is compared with Φ_{echo}. The equality between Φ_{echo} and Φ_x is discovered only when user has availed the authorized number of attempts. Until this equality is revelad, the TTP will recover Φ_x as $\Phi_{residual}$, where $\Phi_{residual}$ is a random value giving no clue when Φ_{echo} will appear again. The proposed methodology achieves desirable results to conceal user access limit on outsourced data and also helps to terminate user access automatically.

6. IMPLEMENTATION AND RESULTS

The proposed idea is implemented and tested on Google cloud echo system using a Google App Engine (GAE)[5] and a desktop machine. The process of user registration and responsibility of third party is tested on desktop machine where as oblivious evaluation of user request is done on GAE. Google app engine SDK is used to deploy application as a web service on GAE. A user request for specific number of attempts is initialized using local machine and all parameters are shared as discussed in section5. User request is then initiated from local machine to third party where it is evaluated and its computational results are sent to Google cloud. The web service running there interacts with incoming parameters and evaluates the results. For homomorphic operations we implement the open library available as an implementation of Pascal Pallier cryptosystem [7].

The system is tested on local machine and Google App Engine [5]. Process of user registration and subsequent evaluation on behalf of TTP is executed on local machine. Initial setup and third party evaluation time is shown in Figure-3-(a) and (b). In Figure-3-a we have tested the initial setup for multiple number of attempts appearing on X-axis for with homomorphic key size 512,1024 and 2048. Here it can be seen that the only factor that increases computational time is the key size, not the number of attempts. Role of CSP and relevant execution on user request is done on Google App engine as shown in Figure-3-(c). Similarly the request evaluation on CSP is tested with key size 512,1024 and 2048. While evaluating user request on App engine we have selected the F4 front end instance. This instance has 2400-MHz processing power and 512-MB of RAM. Figure-3-(d) shows the cpu time, response time and cost analysis of user request on F4 instance. Performance analysis revealed that the cost of operations performed are within the range of .097 to .278 $ per 1000 requests. Initial setup

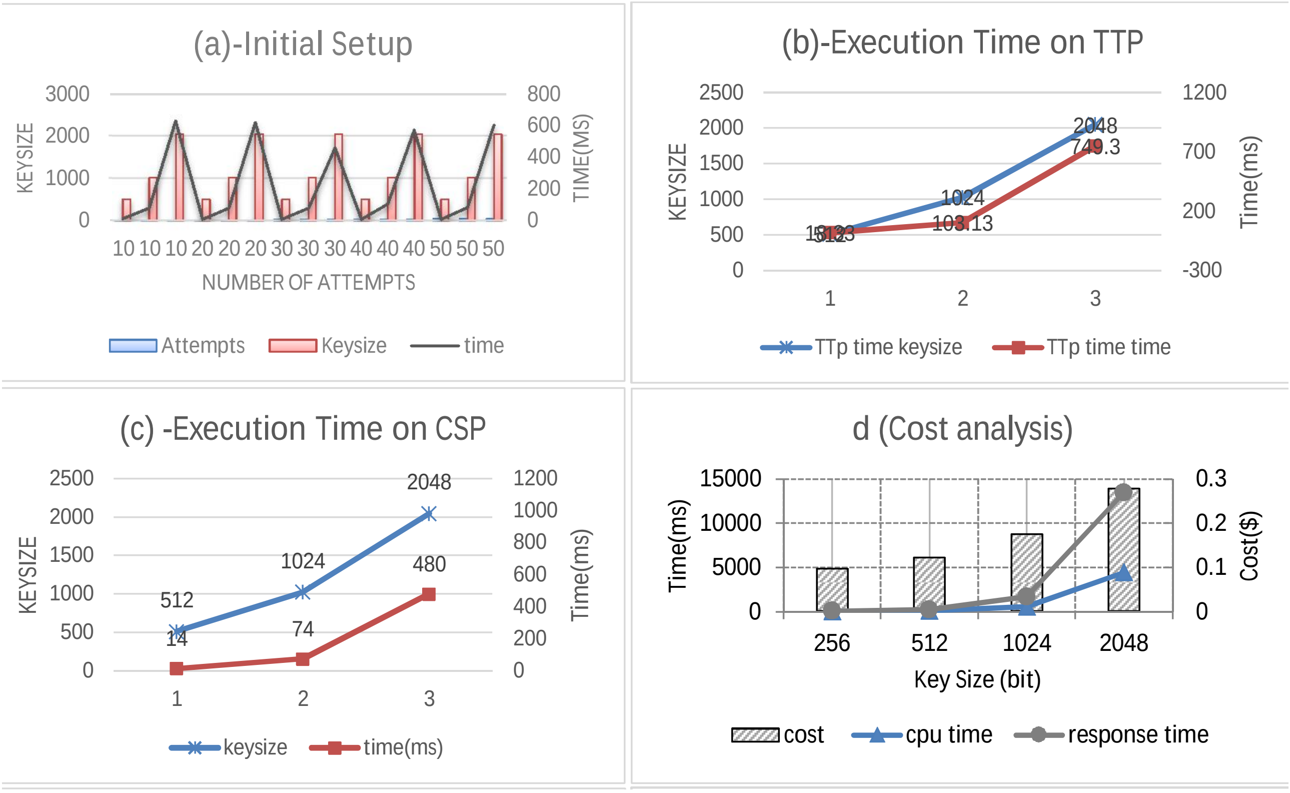


Figure 3: Evaluation Results

by owner and execution steps by third party are tested on a local machine with `Intel(R) Core(TM) i3 processor` and `4GB` of RAM. Microsoft(R) Windows7(TM) X `64`bit is the OS installed on it.

7. DISCUSSION

The main concept of system design is two fold. First, it authorize users to avail cloud resources in a more flexible way without imposing any time restriction. Secondly, the efficacy of hiding user attempts has been achieved using the cryptographic primitives i.e, Homomorphic encryption. Minimum complexity of our design have made it light weight at all frontiers which are, setup activity at owner end and partial execution at TTP and CSP. Except data owner, involved entities perform a fixed number of steps for each user request. Other than secure data outsourcing on cloud, the owner task is to crate $\mathcal{P}(.)$ based on requested parameter for authorization. Here the parameter is number of times user needs permission to access the outsourced data. Creation of $\mathcal{P}(.)$ takes fixed number of steps and does not depend on parameter received from user. Construction of $\mathcal{P}(.)$ takes constant number of operations, therefore its complexity is $\mathcal{O}(1)$. Request evaluation at CSP starts after receiving Λ, given in equation 1. Generating value Δ is also an operation with fixed number of homomorphic operations. The output of equation 2 is similar for first or last user request. Similarly calculating Λ at TTP or discovery of echo value consist on constant round operations.

8. CONCLUSIONS AND FUTURE WORK

Considering the issue of user access, inter dependency and its auto revocation on encrypted cloud data, we have proposed a task-oriented system. A user given with predefined number of access authorization over cloud data can avail it without time restriction. The proposed model also preserves user privacy on her authorization limit until it is availed. At present we used services of trusted third party under certain assumptions, however as a future work we will further improve the design by eliminating TTP, thus relying only on CSP. Currently, the model allows its authorized user to avail cloud data without imposing certain time limit, therefore, in our future work we will incorporate concept of time and task together as a hybrid approach.

9. ACKNOWLEDGEMENTS

This research was supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency)" (NIPA-2014-(H0301-14-1003) and by the Industrial Core Technology Development Program (10049079), Develop of mining core technology exploiting personal big data) funded by the Ministry of Trade, Industry and Energy (MOTIE, Korea)

10. REFERENCES

[1] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with

applications to secure distributed storage. *ACM Transactions on Information and System Security (TISSEC)*, 9(1):1–30, 2006.

[2] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *Security and Privacy, 2007. SP'07. IEEE Symposium on*, pages 321–334. IEEE, 2007.

[3] M. Blaze, G. Bleumer, and M. Strauss. Divertible protocols and atomic proxy cryptography. In *Advances in Cryptology?EUROCRYPT'98*, pages 127–144. Springer, 1998.

[4] M. J. Freedman, K. Nissim, and B. Pinkas. Efficient private matching and set intersection. In *Advances in Cryptology-EUROCRYPT 2004*, pages 1–19. Springer, 2004.

[5] Google. Google app engine. "https://appengine.google.com/.

[6] S. Kamara and K. Lauter. Cryptographic cloud storage. In *Financial Cryptography and Data Security*, pages 136–149. Springer, 2010.

[7] K. Liu. Pascal paillier. "http://www.csee.umbc.edu/ kunliu1/research/Paillier.html.

[8] Q. Liu, G. Wang, and J. Wu. Time-based proxy re-encryption scheme for secure data sharing in a cloud environment. *Information Sciences*, 2012.

[9] S. Müller, S. Katzenbeisser, and C. Eckert. Distributed attribute-based encryption. In *Information Security and Cryptology–ICISC 2008*, pages 20–36. Springer, 2009.

[10] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in cryptology?EUROCRYPT?99*, pages 223–238. Springer, 1999.

[11] P. Paillier. Trapdooring discrete logarithms on elliptic curves over rings. In *Advances in CryptologyâĂŤASIACRYPT 2000*, pages 573–584. Springer, 2000.

[12] K.-W. Park, J. Han, J. Chung, and K. H. Park. Themis: A mutually verifiable billing system for the cloud computing environment. *Services Computing, IEEE Transactions on*, 6(3):300–313, 2013.

[13] B. Patel and J. Crowcroft. Ticket based service access for the mobile user. In *Proceedings of the 3rd annual ACM/IEEE international conference on Mobile computing and networking*, pages 223–233. ACM, 1997.

[14] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *Advances in Cryptology–EUROCRYPT 2005*, pages 457–473. Springer, 2005.

[15] G. Wang, Q. Liu, and J. Wu. Hierarchical attribute-based encryption for fine-grained access control in cloud storage services. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 735–737. ACM, 2010.

[16] G. Wang, Q. Liu, J. Wu, and M. Guo. Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers. *computers & security*, 30(5):320–331, 2011.

[17] S. Yu, C. Wang, K. Ren, and W. Lou. Achieving secure, scalable, and fine-grained data access control in cloud computing. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.