Editorial

International Journal of Advanced Computer Research, Vol 6(24)
ISSN (Print): 2249-7277 ISSN (Online): 2277-7970
http://dx.doi.org/10.19101/IJACR.2016.624016

Secure chip based encrypted search protocol in mobile office environments

Hyun-A Park *
Department of Medical Science, Kyung Dong University, Mun Mak-eub, Won Ju-City, Kangwon-do, Korea

Received: 11-May-2016; Revised: 20-May-2016; Accepted: 23-May-2016

Abstract
This paper deals with largely two security problems between the cloud computing service and trusted platform module (TPM) chip as a mobile convergent technology. At first, we solve the social issues from inside attackers, which is caused by that we regard server managers as trustworthy. In order to solve this problem, we propose encrypted DB retrieval system whose server manager cannot access on real data (plaintexts) in mobile office environments of the cloud datacenter. The other problem is that cloud computing has limitless computing resources; however, it faces with the vulnerability of security. On the other hand, the TPM technology has been known as a symbol of physical security; however, it has the severe limitation of use such as hardware constraints or limited amount of non-volatile memory. To overcome the weakness and produce synergic effects between the two technologies, we combine two applications (cloud datacenter service, TPM chip) as a mobile convergent technology. The main methods are TPM-security-client and masked keys. With these methods, the real keys are stored in TPM and the faked keys (masked keys) are implemented for computations instead of real keys. Thus, the result of the faked keys is the same as the real keys. Consequently, this system is secure against both of the insiders and outsiders, the cloud computing service can improve security weaknesses.

Keywords
Security, TPM, Cloud computing, Insiders, Collaborative computing, Synergic effects.

1.Introduction
As all In the fast changing networked computing societies, a variety of information technologies have brought up new types of IT-enabled product and service innovations in our daily-lives. For example, in South Korea, government-led IT policies have established well-organized socio-technical infrastructures using Internet, mobile technologies, and the usages of mobile devices since the last 10 years. Based on this IT infrastructure and increasing usages of smartphone and smart devices, they have currently transformed the information environment paradigm from wired to wireless or to integrate information environments. Yet, these radically developed IT-driven changes in the shifting paradigm have encountered a dilemma–That is the security problem. As for the representational problems, we can take the security weakness of cloud environments and the inside attackers. In public cloud services, we entrust our data to the cloud provider. Therefore, the data are outside of our control, for example, migration policy and virtualized resources in cloud computing environments.

*Author for correspondence

According to the "Computer Crime and Security Survey" [18], 45% of the attacks are conducted by insiders. For example, South Korea's three leading credit card companies' customer data were revealed in 2013, which was caused by insiders. In order to solve these problems, we propose Encrypted DB Retrieval System that a server manager cannot access on real data (plaintexts) in the Mobile Office Environments of the cloud datacenter. All of the data stored in the DB and even querying keywords should be encrypted. There should be no decryption in a server through all processes. In addition, even users do not know their own secret key, but they can decrypt real data by using the masked keys every time. Hence, our proposed system is secure against both of the insiders and outsiders in the Mobile Office Environments of the cloud datacenter. The main methods and results are as follows.

Main methods
Security Client: The role of TPM (Trusted Platform Module) chip - TPM chip applications support cloud security vulnerabilities. We focus on the importance of converging technologies in order to produce synergic effects between the two technologies. In other words, the core aspects related to security are run by TPM and additionally PC, and the others are

conducted in a cloud service or users' mobile devices. This is because the cloud computing service entails limitless computing power as in a pay-as-you-go arrangement [1]; however, it remains in security risks. On the other hand, TPM application technology has a high level of physical security but it has severe limitation of use such as limited command processing function [2, 9]. Hence, we use the TPM embedded PC as a Secure Client in the Mobile Office Environments.

The Masked Keys: Secret keys should not be released from the TPM chip in a PC. This is our Security Goal. To achieve this, we generate a random number every session and mask the real secret keys with that random number. Then, the masked keys can be transferred from the TPM chip and to a terminal device or cloud services. A user or a cloud service manager implements the computations with the masked keys according to the given protocols. Therefore, nobody can know the real secret keys except for the security client (TPM chip in a PC). Compared to the values, the results of the masked keys are the same as the results of the real secret keys [3, 4, 5].

Results & contribution

We can achieve the similar level of security to 'One Time Encryption'. TPM generates a unique random number every session and masks the real secret keys, then computes the results under the encrypted state with the masked keys. Hence, a user/device cannot know the real secret keys. The decrypted results are the same as the ones decrypted with real keys [3,4,5]. The proposed scheme guarantees the secure key management system. Real secret keys are stored in TPM, which provides a high level of physical security, and the real secret keys are not being released from the TPM. Instead of real keys, the faked keys masked with random numbers are used for computations. Our scheme is secure against both of the insiders and outsiders. There is no decryption in a server, because the server manager does not know anything about secret key in our scheme.

Our proposed scheme is convergent technologies in order to produce synergic effects between the cloud computing service technology and the TPM technology. With the augmented security and efficiency in our application, our scheme can protect the system from both of insiders and outsiders. Therefore, if our scheme is to be commercialized, the social security issues caused by insider attacks might be solved.

2.Related technologies

2.1TPM

1. **General:** TPM (Trusted Platform Module) is a computer chip (micro-controller) that can securely store artifacts used to authenticate the platform (Your PC or laptop).

Integrity Measurement, Storage and Reporting: Integrity measurement involves "platform characteristics" obtained by the TPM, which affect the integrity (trustworthiness). For instance, the platform characteristics include whether the machine's hardware configurations or software configurations are running well. Integrity metrics are measured and validated, and then its hash is stored in the shielded locations called the Platform Configurations Registers (PCRs). It represents a key feature and adds its value to a protected storage. Remote attestation and access to sensitive data are allowed only when the stored values in the PCRs and the current integrity metrics match, in other words, when a platform is in a valid and secure state.

Protected Storage: Sensitive data may include cryptographic keys, passwords and digital certificates. The TPM provides a protected storage by encrypting the secret data with a private key that only the TPM has access to. In addition, the TPM can bind the sensitive data to a platform by encrypting the sensitive data along with platform configuration values. The access to the sensitive data is only allowed when the stored and current platform configuration values match.

Remote Attestation: Remote attestation is the attestation-process to a remote party about the fact that a TPM has a valid EK (endorsement key). Therefore, it is a valid TPM. The remote party can trust the platform that the TPM resides on. In conclusion, it is safe to exchange data with it [19]. Trusted modules can be used in computing devices other than PCs, such as mobile phones or network equipment.

2. Weakness: The TPM spec does not provide the minimum performance requirements, so that today's commodity TPMs are slow and inefficient. This performance handicap has limited the use of TPM to the scenarios without requiring fast or frequent operations [2].

TPM provides a limited amount of non-volatile memory (NVRAM) which can be used to persist state across reboots. A region of NVRAM can be allocated

and protected, so that it can be accessed only when specified PCRs contain specified values. TPMs also provide non-volatile monotonic counters which can be updated only by an increment operation (TPM Increment Counter) [9].

2.2Cloud computing

1. General: Cloud computing is a general term for anything that involves delivering hosted services over the Internet. These services are broadly divided into three categories: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). A cloud service has three distinct characteristics that differentiate it from traditional hosting. It is sold on demand, it is elastic – a user can have as much or as little of a service as they want at any given time; and the service is fully managed by the provider. A cloud can be private or public. A public cloud sells services to anyone on the Internet. A private cloud is a proprietary network or a datacenter that supplies hosted services to a limited number of people. When a service provider uses public cloud resources to create their private cloud, the result is called a virtual private cloud. Private or public, the goal of cloud computing is to provide easy, scalable access to computing resources and IT services. Infrastructure-as-a-Service (like Amazon Web Services) provides virtual server instance API to start, stop, access and configure their virtual servers and storage. In the enterprise, cloud computing allows a company to pay for only as much capacity as is needed, and bring more online as soon as required. It is referred to as utility computing, because this pay-for-what-you-use model resembles the way electricity, fuel and water are consumed. Platform-as-a-service in the cloud is defined as a set of software and product development tools hosted on the provider's infrastructure. Developers create applications for the provider's platform over the Internet. Force.com(an outgrowth of Salesforce.com) and GoogleApps are examples of PaaS. In the software-as-a-service cloud model, the vendor supplies the hardware infrastructure; the software produces and interacts with the user through a front-end portal. SaaS is a very broad market. Services can be anything from Web-based email to inventory control and database processing [1].

2. Weakness: In cloud environments, especially in public cloud environments, there needs to be a broader understanding of security technologies than previously in non-cloud environments. That is because we are now putting data in an environment that is outside of our control. It is within the control of a third-party cloud provider. We may have some level of control, but we do not have complete control. If the physical server is attacked or compromised, it can impact all of those IT resources. It is not just about bringing solutions down. It is about access to data, or the abuse of data or the attempt of malicious actions. The attacker may be one of the cloud consumers or managers that have gained legitimate access to their virtualized IT resources with their own trust boundaries [6, 17].

3.Application scenario and model

3.1Notations

In this paper, we use the notations as follows;

- idf : identification function
- MAP : mapping function; $v(\in A_i) \rightarrow PI_J = idf_{R.Ai}(p_j)$
- A_i : attribute
- p_j : each partition for numeric data attribute A_i
- PI_J : Parition identifier
- p_u : user u's PIN(Personal Identification Number)
- $1|m_1^t$: the expression for text data m which is located in the first column(attribute).
- $|p+q|m_{p+q}^n$: the expression for numeric data m which is located in the $(p+q)-th$ column(attribute).
- $k_j \in K \in \{0,1\}^k$: keys set for each attribute
- f_k : pseudorandom function with secret key k
- t_i : tuple identifier
- $1|m_1^t,..,p|m_1^t,p+1|m_{p+1}^n,..,|p+q|m_{p+q}^n$: the expression for data one tuple
- F_j : attribute for text data
- P_j : attribute for numeric data
- g : a generator of a group G
- $g^{k_j \cdot f_k(k_j)} m_{i,j}$: column-level encryption of data m
- K_j^1, K_j^2, K_j^3 : the masked keys with random number γ to hide real key k_j
- $h()$: hash function

3.2Application and participants

Our application is a mobile office environments with storage system in the cloud datacenter. The problem is that organizations or individuals cannot trust the datacenter server managers, and these organizations or individuals cannot be sure of data integrity from the unauthorized accesses or attacks. Therefore, a security solution is required, including authentication, access control, encryption, and so on. As the solution, we select TPM as a possible method. In our scheme, TPM acts the important role of a security client as a kind of TTP as well as Integrity Measurements. In this paper, two kinds of clients are defined: a general and security clients. We grant TPM a role of a security client. Participants of our schemes are made

up of 1) a user / a terminal device, 2) TPM chip in a PC, and 3) the datacenter. A mobile device is treated as a terminal device (TD). Thus, users can authenticate themselves and can access the cloud computing services [13,14].

1) User (U_i) /Terminal Device (TD_i) is a general client. A user can access the cloud services by the terminal device and this general client manages all cloud service systems processes only except for security parts. Hence, even a user cannot know his own secret key, but he can use his masked keys whose result are the same as the real secret key.
2) TPM chip in PC (AR_{si}). TPM chip plays a special role of a security client and PC (AR_{si}) also does the role of "authority of registration" as a TTP (Trusted Third Party). The special tasks involve as follows.

- TPM chip stores secret keys and all of the information related to the secret keys are given at enrollment time. The authentication information such as PIN and $h(P_u)|P_u$ should be stored. Each attribute secret key and partition Tables for numeric data should be also stored in TPM chip.
- After a user authentication, TPM makes a legitimate user encrypt data with the masked secret keys and the other information.
- PC with TPM masks decryption keys with a random number, and transfers the masked keys and the results from the datacenter server to a terminal device.

3) Datacenter server (DS) is the storage server of cloud services. It can be private or public. We assume the server manager is untrustworthy. Thus, all of the sensitive information should be encrypted and there is no decryption process in a datacenter.

3.3DAS (Database as a service) Model

Our database model is based on the database as a service (DAS) model [8]. The properties of DAS model are that a client is perfectly trusted so that it can know and control all secret information. The client's all data except for secret information are stored in and managed by an untrustworthy server. The client has restricted computational power and storage, and relies on the server for mass computational power and storage. Because a server is untrustworthy, the server cannot learn any secret information on its DB data and can do only storage and computation according to the queries of the client. In this paper, TPM security client and the datacenter server manager have the same role in DAS mode [7, l0].

3.4Attacker types

We consider largely two types of attackers;

• Outside attackers - Outside attackers illegally try to access data that they do not have authorization.

• Inside attackers - They can be free from DB access controls. We take examples by an untrustworthy server manager or compromised customers, who can originally access to the DB.

4.TPM controlled encrypted SQL(TPM-ESQL) protocol in the cloud datacenter

In this section, we construct our proposed protocol, from the first step System SetUp and Enrollment to the sixth step Decryption.

4.1System setup and enrollment

Idf is an identification function which maps each partition p_j for a numeric data attribute A_i to an identifier $idf_{R.A_i}(p_j)$ like this; $idf: A_i \rightarrow idf_{R.A_i}(p_j)=PI_j$. A mapping function $MAP_{R.A_i}$ maps a value v within an attribute A_i's domain to the partition identifier PI_j to which the v belongs: $MAP_{R.A_i}(v)=idf_{R.A_i}(p_j)$. The secret keys for each attribute and partition Tables for numeric data attributes are offered in the user's TPM at enrollment time.

4.2Authentication and query preparation

$[U_i \rightarrow TD_i]$

1 Input; $p_u, R(1|m_1^t,..p|m_p^t, p+1|m_{p+1}^n,..p+q|m_{P+q}^n)$ relation.

$[TD_i]$

2.Compute and Transfer to a TPM chip; $h(p_u)'|E_{p_u}(R(1|m_1^t,..p|m_p^t, p+1|m_{p+1}^n,..p+q|m_{P+q}^n))$

[TPM chip in PC (AR_{si})]

3.Verify in PC ; $h(p_u)'=h(p_u)$.

Decrypt in TPM; $D(E_{h(p_u)}(k_j))=k_j, (1\le j\le p+q)$

Generate γ and a tuple identifier t_i in TPM.

Compute in PC ; $K_j^1=k_j\cdot\gamma$, $K_j^2=k_j\cdot f_k(k_j)\cdot(1-\gamma)$, $K_j^3=g^{f_k(k_j)}$.

Masked keys for a tuple t_i; $f_k(t_i), K_1^1, K_1^2, K_1^3, ...K_{p+q}^1, K_{p+q}^2, K_{p+q}^3$.

In TPM -ESQL, the data which consists of DB Tables are largely classified into two: text data and numeric data. Let R be a relation which has the following attributes set: $R=\{r_1, r_2, ...r_n\}$. $1|m_1^t$ is a text data m for the first attribute and is a numeric data m for the $(p+1)$-*th* attribute. $k_j(1\le j\le p+q)$ are secret

keys for each attribute. $j \mid E_{h(p_u)}(k_j), 1 \le j \le p+q$; This is the type stored in TPM.
1) A user inputs his/her PIN p_u and the number of n data belonging to n attributes, to TD_i.
2) TD_i computes the hash value $h(p_u)'$ and encrypts the n data with p_u.
3) $h(p_u)'$ is the received data. $h(p_u)|p_u$ is stored in the TPM. If the verification is satisfied, the TPM decrypts k_j (secret keys for each attribute), generate random numbers γ , ti. With random number γ , PC computes the masked keys K_j^1, K_j^2, K_j^3 to hide the real key k_j *(1≤j≤p+q).*

4.3DB encryption

Table 1 Partition Table for the J-th attribute pj in a relation R

R.PJ	
Partitions	PI(Partition ID)
[value$_1$, value$_2$]	PI$_{j,1}$
(value$_2$, value$_3$]	PI$_{j,2}$
…….	……
(value$_l$, value$_{l+1}$]	PI$_{j,l}$

$$idf_{R.P_j}([value_1, value_2]) = PI_{j,1},$$
$$MAP_{R.P_j}(value_y) = PI_{j,1}, \text{if } value_y \in [value_1, value_2]$$

The encrypted relation R in a datacenter server has the following attributes:

- TID: $f_k(t_i)$. This is an encryption of a tuple identifier t_i and k is a TPM's secret key.
- Column-level encrypted attributes $F_j (1 \le j \le p+q) : F_j = g^{k_j \cdot f_k(k_j)} m_{i,j}$, where k_j are the secret keys for the *j-th* attribute and $m_{i,j}$ are the values for the *j-th* attributes in the *i-th* tuple. F_j is a text data if $1 \le j \le p$, and a numeric data if $p+1 \le j \le q$. That is, F_j*(1≤j≤ p+q)* consist of p text data and q numeric data, where equality selections, equijoins, grouping, etc. are operated.
- Partitioning attributes $P_j (p+1 \le j \le q)$: Range queries for numeric data are operated by partition indexes for each attribute value. Namely, $PI_{j,x} = MAP_{R.P_j}(m_{i,j})$, where a numeric data value $m_{i,j}$ is again mapped to a partition index $PI_{j,x}$ (the *x-th* partition index of the *j-th* attribute, $1 \le x \le l$). *Table 1* shows the partition Table for the *j-th* attribute P_j. Given the above attributes, the relation R^S in a datacenter server is as follows: R^S *(TID,* $F_1, F_2, \ldots, F_{p+q}, P_1, \ldots, P_q$*).*
  [TPM chip in PC *(AR$_{si}$)*]

4. Compute in PC(Column-level Encryption); $(K_j^3)^{K_j^1} g^{K_j^2} m_{i,j} = g^{k_j \cdot f_k(k_j)} m_{i,j}, (1 \le j \le p+q)$
5. Do Partition and Transfer to DS; R^S*(TID,* $F_1, F_2, \ldots, F_{p+q}, P_{p+1}, \ldots\ldots, P_{p+q}) = (f_k(t_i), g^{k_1 \cdot f_k(k_1)} m_{i,1} \cdots g^{k_{p+q} \cdot f_k(k_{p+q})} m_{i,p+q}, PI_{p+1,x}, \ldots, PI_{p+q,x})$.

4) For all data(attributes), the PC does the column-level encryption.
5) For the data $(p+1 \mid m_{p+1}^n, .. p+q \mid m_{P+q}^n)$;numeric data, PC gets the partition indexes $PI_{j,x,}$ ($p+1 \le j \le q$) from the partition Table in TPM and sends the additional "PIs"(partition ID) to DS. If the input data from TD_i are new, TPM generates "PIs" and stores them as a new partition Table and then transfers the query to DS.

4.4Querying

[TD_i]
6. Generate; Q(query)

Compute and Transfer to PC with TPM chip; $h(p_u)' \mid E_{p_u}(Q)$
[TPM chip in PC *(AR$_{si}$)*]
7. Verify; $h(p_u)' = h(p_u)$.

Decrypt; $D(E_{p_u}(Q)) = Q$.

6, 7) This process shows that TD_i generates 'Query' and sends it to *DS* cryptographically

Query Decomposition

Certain Query Q_c^s: Certain Query selects the tuples which can be assured about the fact that they satisfy the originally given condition or not. Aggregation operations are possible in a server.

Guessing Query Q_g^s: The selected tuples to this query cannot be assured about the fact that they satisfy the original given condition unless they are decrypted. Hence, a datacenter server outputs the encrypted values for the attribute without aggregation operations. That is, if the query is for P_{p+j}, a server outputs the matching attribute values themselves, $g^{k_{p+j} \cdot f_k(k_{p+j})} m_{i,p+j}$ for F_{p+j} in the tuple[7].

A. Query for equality test

[TPM chip in PC *(AR$_{si}$)*]
8. Compute in PC; $Q^s = Q_c^s = g^{k_j \cdot f_k(k_j)} v$.

Q expresses various types of queries. If the query is for an equality test, it can be expressed as *j|v*, where *j* is an attribute to be searched and *v* is a text or numeric data. PC should transform this to a query Q^s for a datacenter server. In equality tests, Q^s consists

of only certain queries Q_c^s, not guessing queries Q_g^s. A PC encrypts v with the secret key k_j for *j-th* attribute: $g^{k_j \cdot f_k(k_j)} v$. This is a query Q^s to a datacenter server.

8)In Equality Test, $Q(=j|v) \rightarrow Q^s = Q_c^s$

B. Query for Comparison Test

[TPM chip in PC (AR_{si})]

8′. Produce (Query Plan); $Q^s = Q_c^s \vee Q_g^s$.

8′) Comparison tests are for range or MAX query of numeric data. In our encrypted search system, we use a partition method like *Table 1*. PC transforms a query Q into a query $Q^s = Q_c^s \vee Q_g^s$ with the partition Table[5,10].

4.5Searching

[*DS*]

9. Search; $Q^s = Q_c^s \vee Q_g^s$ → Result; $R^s = R_c^s \vee R_g^s$, $R_g^s = \{ f_k(t_i), g^{k_j \cdot f_k(k_j)} m_{i,j} \}$, $(p+1 \le j \le q) \in F_{i,j}$

Transfer to a Security Client; Result.

R_c^s is a final aggregation result for a certain query and R_g^s is a result for processing a guessing query.

[TPM chip in PC (AR_{si})]

10. Operate Aggregations in PC (guessing query); *m* (value *m* in *Q*) →PI

Decrypt in TPM; $f^{-1}(f_k(t_i)) = t_i$, $g^{k_j \cdot f_k(k_j)} m_{i,j} = (g^{k_j \cdot f_k(k_j)} m_{i,j}) \cdot g^{-(k_j \cdot f_k(k_j))} = m_{i,j}$, ($p+1 \le j \le q$, for some *j*) and process final result.

Generate in TPM; α.

Compute in PC; $C_1 = g^{k_j \cdot f_k(k_j)} v_{i,j}$, $C_2 = g^{f_k(k_j)}$, $C_3 = -(k_j + \alpha)$, $C_4 = f_k(k_j)\alpha$

$C_5 = g^{k_j \cdot f_k(k_j)} m_{i,j}$

Transfer to a TD; $R^{TD} = R_c^{TD} \vee R_g^{TD}$, $R_c^{TD} = R_c^s$ $=\{C_1, C_2, C_3, C_4\}$, $R_g^{TD} = \{C_5, C_2, C_3, C_4\}$

9) DS searches for certain or guessing query.

10) For guessing query, at first, the value m in Q is mapped to a partition index PI. Because all the partition Tables are stored in TPM, aggregations are operated in the PC with TPM. If the result of the first aggregation is not assured, the PC can decrypt some expected values (results). Then, finally the PC encrypts the results (certain and guessing query) with the generated random number and masked keys C_2;C_3;C_4[12,13].

4.6Decryption

[TD_i]

11. Decrypt; $C_1 \cdot C_2^{C_3} \cdot g^{C_4} = v_{i,j}$,

$C_1 \cdot C_2^{C_3} \cdot g^{C_4} = (g^{k_j \cdot f_k(k_j)} v_{i,j}) \cdot (g^{f_k(k_j)})^{-(k_j+\alpha)} \cdot g^{f_k(k_j)\alpha} = v_{i,j}$,

$C_5 \cdot C_2^{C_3} \cdot g^{C_4} = (g^{k_j \cdot f_k(k_j)} m_{i,j}) \cdot (g^{f_k(k_j)})^{-(k_j+\alpha)} \cdot g^{f_k(k_j)\alpha} = m_{i,j}$.

11) To the given above protocol, we can get real data with masked keys. It is the same value as the decryption of 10 ($g^{k_j \cdot f_k(k_j)} m_{i,j}$), which is used with real secret.

5.Experiments

This paper deals with many technologies of the combined authentication, access control, and data search. We experiment on our scheme as a prototype. However, the main purpose of our paper is not for the design of building block algorithms. The performance of our scheme definitely depends on the subordinate algorithms and how much money you pay for the cloud services, only except for some parts. By these reasons, the comparison and analysis of performance with other papers are not appropriate. We discuss the performance of our scheme with separate six phases;

1. Steps 1-3): Authentication and Query Preparation,
2. Steps 4-5): DB Encryption,
3. Steps 6-7): Querying,
4. Steps 8 &8′): Query for Equality and Comparison Test,
5. Steps 9): Searching,
6. Steps 10-11): Decryption

We implement only three parts, i.e., Steps 1-3), Steps 4-5), and Steps 10-11), which have a relatively much influence on our performance. It is true that our whole performances are determined much more by environmental factors such as network stability and speed, mobile phone capability; server's computing power other than our proposed scheme. Moreover, there was no prior study to be compared with our scheme, which considers most of the processes over authentication, access control, searching, private key management, etc. Consequently, respective analysis for each step seems to be quite proper for our scheme. The parts, Steps 6-7), 8&8′), 9) are excluded from our experiment, because the performance of these parts is more up to mobile capability, the amount of data which a user wants to search, or

environmental factors other than by the schemes we designed newly.

We experiment with both on a PC and a mobile phone. The processing power of our mobile phone is Qualcomm APQ8064 1.5 GHz Quad CPU Core on Snapdragon, 32GB eMMC, LPDDR 2GB. Our personal computer is Intel(R) Core(TM) i5-4570 CPU 3.20 Ghz processor and 8GB RAM and OpenSSL cryptography modules for cryptographic operations. As for the TPM, Infineon OPTIGA TPM SLB 9660 is run in our experiments and the chipset is SLB 9660-1.2, compliant to TPM, 1.2 Rev.116, LPC interface 24/33MHz.

We set p=10, q=5, i.e. totally all data n(p+q)=15.
Steps 1-3): The computing time of this step consists of mobile phase and PC phase. As the step for authentication and preparation for queries, masked keys are generated. Because we used common algorithms and commands from OpenSSL cryptography modules and TPM Command Table, the performance depends on the subordinate algorithms.

Steps 4-5): This step is composed of only PC phase. The step 4 consists of our newly designed protocol for all data's column-level encryption. The step 5 is composed of "getting PI (partition index)" for numeric data in TPM and transferring to DS. The result is shown in *Table 3*.

Steps 10-11): This phase is also composed of mostly our newly designed protocols. The step 10 is for the decryption process for guessing query of numeric data and generation of *in* TPM, and the masking process of final results with random number *in* PC. The step 11 is the final decryption process by masked values C_1; C_2; C_3; C_4; C_5 in terminal device. The result is shown in *Table 4*[13, 14, 15].

5.1Performance analysis

The above Tables show that the configuration of our protocol makes it possible to run on a mobile device without a problem. The total time of *Table 2* seems to take a little long time (376.18 ms), but this phase is the time for authentication and uploading of all data n (p+q) to query. In addition, the time for decryption takes only 10.4 ms in a mobile phase as shown in *Table 4.*

But, in the middle of experiments, we found some problems in processing TPM commands because there are too small kinds of available commands. For example, AES Decrypt does not have many commands, only except for *TPM LoadContext* (1024 bytes) in (TPM 1.2 Rev.116). It shows that we need to make a plan every time for complex processing configuration to run the TPM embedded PC. However, it is expected that the performance would be improved after TPM v2.0 is commercialized.

Table 2 Performance of steps 1-3 (unit: ms)

Total Time		376.18	
Mobile phase	83.18	PC phase	293

Table 3 Performance of steps 4-5 (unit: ms)

Total Time (Pc Phase)	37.38

Table 4 Performance of steps 10-11 (unit: ms)

Total Time		35.9	
PC phase	25.5	Mobile phase	10.4

6.Discussion

6.1Security

The cloud computing technologies offer a variety of IT infrastructures through extending IT environments outward by public cloud services and their migration policies. The development of cloud computing technologies makes that security and reliability has become the core issues of the information protection. The risk of the information leakage is quite high because software, data, and most of the IT resources can be provided as services. Like this, virtualized IT resources make the traditional concept of "boundaries" to protect from attackers "blurred" in the networked information societies. In addition, we cannot guarantee that the service provider or server manager is trustworthy or not. In this section, we analyze the security issues for the proposed scheme in the cloud services with the TPM and encryption as follows:

1. **Security Client:** We assigned a TPM with the highest level of physical security to the role of a security client. All processes related to security should go through a security client.
2. **Intractability of Decryption by a Server:** There is not any decryption process in a datacenter, because a datacenter server is assumed untrustworthy, so that decryption processes are allowed only in a terminal device.
3. **Entity Authentication at Access Time:** We used PIN as an entity authentication method with a TPM, and it is popular in a real world.
4. **Access Control by Encryption and Authentication:** If users can pass both of the

TPM authentication and the access authentication of cloud services, the legitimate users can generate available queries. A datacenter server can search for the encrypted data with the available encrypted queries. Users decrypt the encrypted results with a masked secret key. In other words, if a user cannot be authenticated as a valid user, the user cannot generate valid queries and cannot access encrypted data. A datacenter server cannot search without valid queries [11, 13].

5. **Managing Real Secret Keys in the TPM:** Only except for the TPM security client, real secret keys for encryption and decryption should not be known to anyone. Even a user does not know his/her own key. The secret keys are stored and managed securely in the TPM, and cannot be revealed to outward because of its high level of security properties [7, 10].
6. **Masking Real Secret Keys with Random Number:** Real secret keys are managed by the TPM, while the encrypted results from a datacenter server are decrypted in a terminal device of a general client. To achieve this, the TPM generates a unique random number every session and masks real secret keys, then it yields the results encrypted with the masked keys. Hence, a user/device cannot know the real secret keys. The decrypted results are the same as the ones decrypted with real keys, whereby the limitations of the TPM can be overcome and the security in cloud computing services can be augmented [5,16].
7. **Data Protection by Encryption:** All sensitive data in a datacenter server are encrypted. Secret keys should not be released from the TPM, and the secret keys are managed and stored securely by the TPM with the highest level security. One of the most important things is that there is no decryption in a datacenter server and the server manager and attackers cannot access the secret keys. Thus, our scheme is secure against insiders and intruders. Consequently, our scheme can prevent abuse or misuse of personal information by a server manager, and it can protect our system from privacy infringement.

6.2Efficiency

Our goal for efficiency is largely two; 1) less complex processing configuration in the TPM embedded Service System, 2) the minimum computation in a TPM and the maximum computation in a cloud. This is because that the TPM does not have so many kinds of available commands and limited computing power, while the cloud has powerful computing power and resources. The followings show our efficiency.

1. Encryption Process and Search Process

Column-level Encryption: For all data, including text and numeric data, we implement Column-level Encryption in PC. Even if this encryption requires exponential calculation, *Table 2* shows that it has no problem because the computation is appropriate for PC efficiently.

Partition Encryption: This encryption method is used for only numeric data in TPM. Guessing Queries cannot be implemented in the server, but in the TPM. This method needs additionally storing partition tables, but its computation is extremely light compared to other schemes such as homomorphism.

2. Efficient Calculation

To achieve less complex processing configuration in the TPM embedded Service System, we implement heavy computation like exponential calculation in a PC or a Server. As to Searching process, only Guessing Queries for numeric data should be implemented in the TPM chip embedded PC.

3. Total Performance

Total performance depends on the amount of data stored in the DS (datacenter server) and the amount of the price we pay for Cloud datacenter services. This is because most of the heavy computations are implemented in a datacenter server.

7.Conclusion

In this paper, we tried to solve two security issues; the inside attackers and the security vulnerabilities in cloud computing environments. Thus, our proposed scheme can protect the system from insiders as well as outsiders in the mobile office environments in the cloud datacenter. As a result, this paper includes two contributions as follows: (1) it highlights the importance of security issues in the cloud computing era of expanding the coverage of security more broadly; (2) it identifies a future converging technological solution by assembling the advantages between the cloud computing service technology and the TPM technology. For the future research, the collaborative computing environments might offer a variety of research potential for the communities of security and mobile computing. Therefore, we need to consider the security as the core issues, including ISM (Integrated Security Management) in the collaborative computing environments.

Acknowledgment

None.

Conflicts of Interest

The author has no conflicts of interest to declare.

References

[1] Fox A, Griffith R, Joseph A, Katz R, Konwinski A, Lee G, et al. Above the clouds: a Berkeley view of cloud computing. Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Rep. UCB/EECS. 2009; 28(13).

[2] Chen C, Raj H, Saroiu S, Wolman A. cTPM: a cloud TPM for cross-device trusted applications. In11th USENIX symposium on networked systems design and implementation 2014 (pp.187-201).

[3] Cheon JH, Kim WH, Nam HS. Known-plaintext cryptanalysis of the Domingo-Ferrer algebraic privacy homomorphism scheme. Information Processing Letters. 2006; 97(3):118-23.

[4] I Ferrer JD. A new privacy homomorphism and applications. Information Processing Letters. 1996; 60(5):277-82.

[5] Domingo-Ferrer J. A provably secure additive and multiplicative privacy homomorphism*. In information security 2002 (pp. 471-83). Springer Berlin Heidelberg.

[6] Gregg M. 10 Security Concerns for Cloud Computing. http://www.globalknowledge.be/content/files/documents/386696/386784. Accessed 11 April 2016.

[7] Hacigümüş H, Iyer B, Li C, Mehrotra S. Executing SQL over encrypted data in the database-service-provider model. In proceedings of the ACM SIGMOD international conference on management of data 2002 (pp. 216-27). ACM.

[8] Hacıgümüş H, Iyer B, Mehrotra S. Efficient execution of aggregation queries over encrypted relational databases. In database systems for advanced applications 2004 (pp. 125-36). Springer Berlin Heidelberg.

[9] Kotla R, Rodeheffer T, Roy I, Stuedi P, Wester B. Pasture: Secure offline data access using commodity trusted hardware. In presented as part of the 10th USENIX symposium on operating systems design and implementation (OSDI 12) 2012 (pp. 321-34).

[10] Mykletun E, Tsudik G. Aggregation queries in the database-as-a-service model. In data and applications security 2006 (pp. 89-103). Springer Berlin Heidelberg.

[11] Park HA, Hong JW, Park JH, Zhan J, Lee DH. Combined authentication-based multilevel access control in mobile application for DailyLifeService. IEEE Transactions on Mobile Computing. 2010; 9(6):824-37.

[12] Park HA, Lee DH, Zhan J, Blosser G. Efficient keyword index search over encrypted documents of groups. In IEEE international conference on intelligence and security informatics 2008 (pp. 225-9). IEEE.

[13] Al-Qayedi A, Adi W, Zahro A, Mabrouk A. Combined web/mobile authentication for secure web access control. In wireless communications and networking conference 2004 (pp. 677-81). IEEE.

[14] Ricci R, Chollet G, Crispino MV, Jassim S, Koreman J, Olivar-Dimas M, et al. Secure Phone: a mobile phone with biometric authentication and e-signature support for dealing secure transactions on the fly. In defence and security symposium 2006 (pp. 625009-16). International society for optics and photonics.

[15] Song DX, Wagner D, Perrig A. Practical techniques for searches on encrypted data. In proceedings of IEEE symposium on security and privacy 2000 (pp. 44-55). IEEE.

[16] Wagner D. Cryptanalysis of an algebraic privacy homomorphism. In information security 2003 (pp. 234-9). Springer Berlin Heidelberg.

[17] http://searchcloudapplications.techtarget.com/feature /Cloud-migrationstrategy- Consider-portability-security -overall-risk. Accessed 11 April 2016.

[18] Power R. CSI/FBI computer crime and security survey. Computer Security Journal. 2001; 17(2):20-51.

[19] Kim R. Trusted Platform Module and Privacy Promises. https://www.cs.auckland.ac.nz/courses/compsci725s2c/archive/termpapers/skim.pdf. Accessed 16 March 2016.

Hyun-A Park She received the BS degree from the Department of Mathematics at Korea University, Seoul, in 2003, and the MS and PhD degrees in information security from Korea University, Seoul, in 2005 and 2010, respectively. Currently, she is an assistant professor with KongDong University. Her main research interests include medical(health) information security, practical retrieval system on encrypted database systems. She is interested in database security, access control, privacy preserving data mining (PPDM), anonymous communication channel, privacy enhancing technology (PET), and cryptographic protocols.

Email: kokokzi@kduniv.ac.kr