A Confidentiality-Guarantee Mechanism for SaaS

Guozhen Ren, Qingzhong Li, Yuliang Shi, Lizhen Cui

A Confidentiality-Guarantee Mechanism for SaaS

Guozhen Ren, Qingzhong Li, Yuliang Shi and Lizhen Cui

School of Computer Science and Technology
Shandong University
Jinan, China
{rgz,lqz,liangyus,clz}@sdu.edu.cn

Abstract. In SaaS Applications, the data of tenants are stored in the untrusted service provider side, this case increases the risk of data leakage, and becomes the brief reason to prevent people and enterprise from taking SaaS mode for their applications. Correspondingly, confidentiality-guarantee has become the key factor of its large scale promotion. In this paper we propose a general secure mechanism that allows the sensitive data of tenants to be stored in encrypted mode, which guarantees data confidentiality on the assumption that application server is trusted, and then put forward the analysis of its security and performance.

Keywords: SaaS; data confidentiality; trusted application server

1 Introduction

In software as a service (SaaS) applications, high efficiency of deployment, upgrading, and maintenance are the motivation of its development. But in this case, the data of tenants are stored in the service provider side, which increases the risk of data leakage. No tenants wish to lose data confidentiality in any case, therefore, data confidentiality-guarantee is very important for tenant deciding to adopt SaaS.

In the related research works, the secure problems of SaaS are usually divided into secure access control, data integrity protection, data privacy protection and data confidentiality protection[4]. Some research results focused on secure access control and data integrity had been put forward, and most of them were based on corresponding mechanisms in DAS and Web fields. But in SaaS applications, business logic calculation is executed in the service provider side and multiple tenants share data storage, which is essentially different from traditional applications, such that the traditional data confidentiality control mechanism cannot be directly applied to SaaS confidentiality-guarantee mechanism.

In this paper, we put forward a confidentiality-guarantee mechanism for typical SaaS framework under the trusted application computing environment. In this mechanism, we refer to the remote attestation method for trusted computing environment [2], and give the key agreements protocols which can transfer the data

encryption key to the application server in security, and so the server can transparently encrypt/decrypt the data of tenants through the confidential engine in processing. And this can make the data of tenants to be stored in encrypted mode in database. This mechanism is secure supposing application system is trusted.

In fact, the data leakage events occurred in the data storage is the most of the actual data leakage occurrence. In our mechanism, sensitive data defined by tenants stored in the physical storage are all encrypted, thus to prevent this kind of leakage events completely. So our research work has practical significance. Specifically, this paper makes the following contributions:

1) We propose a confidentiality-guarantee framework supposing the application server is trusted, and point out its practical significance.

2) Design of a confidential-guarantee mechanism under the typical SaaS framework, and this mechanism secures data transparently to application developer.

The rest of this paper is organized as follows. Section 2 introduces the related works. Section 3 analyzes the confidential problem in general SaaS model, and gives the hypothesis and the practical application of the scene of our research. Section 4 address a security framework in SaaS. Section 5 gives the corresponding security protocols and data encryption mechanism of the framework. Section 6 analyzes the security and performance of the mechanism. Experimental results are reported in Section 7. The paper concludes in Section 8 with the summary and future works.

2 Related Work

At present, the research works for data confidentiality-guarantee mechanism for SaaS is in its infancy. In the related area, research works focused on secure access control and integrity of data in SaaS, secure Data As Services (DAS), new encryption algorithms for database and trusted computing etc.

The research works of confidentiality in DAS can be divided into two cases which depend on the data be stored in document or database. For the former, there are key word query mechanism in literature based on document encryption storage [8,9,12,13]. In these mechanisms, documents are switched to cipher before stored to untrusted service center, mechanisms offer key word search when customers use these documents, the inquired content will be decrypted and used after being transferred to client side, which realizes confidentiality protection of document content. And, the literature [1] puts forward a multiple key words query mechanism over documents encrypted, in condition of the main document encryption, it storages multiple keywords based on message hidden methods, and gives a multiple key word query mechanism which supports and/or query operation of query builder. Literature [1] also give a complete solution about database mode on base of literature[6,7,9,10]. Based on order preserving encryption algorithm and fully homomorphic encryption, the mechanism supports basic compare operator and arithmetic operator and SPJ (select-project-join) operations for SQL query, and gives the query optimization mechanism.

With the development of SaaS application, Secure problems quickly become a hot spot in the study of the industry, literature [4,14] classify secure problems

systematically. Now there are dual authentication and RBAC mapping for tenants to answer for problems in secure access control and authorized management[19]. With redundant tuples and digital signature technology, problems are partly solved in data integrity protection. But for data privacy and data confidentiality, Only SSL, AS2, S-FTP are put forward, these mechanisms can only solve data leakage problems in transmission, but can't solve leakage problems in the database level and application service level when sensitive data of tenants are stored in the untrusted service provider side. Literature [14] puts forward that data confidentiality as a challenge in SaaS.

Research works about trusted platform are abundant, and the remote attestation mechanism under the TCG framework gets closed attention, integrity measure framework IMA [15] and remote attestation mechanism based on the attribute certificate proposed by IBM research institute are typical in all of the works, in this direction, article [2] analyzes the problems of dynamic characteristic, concurrency and consistency for multiple remote attestation instance(Multi-RAI) in trusted computing environment, and propose a complete dynamic update attestation mechanism for Multi-RAI in trusted computing environment. This also provides a good reference for solving secure problems in SaaS.

In addition, in order to improve efficiency of data query over the encrypted messages, there are some research works about new practical encryption algorithm such as order preserving encryption. Literature [17,18] give practical methods and make high efficient query over the integral and string data types, which provides very good help for SaaS confidentiality study.

3 Confidential Problem Analysis for SaaS

Participators here can be Tenant, SP and Attacker in general.

1) Tenant: Data owner in SaaS, hopes its sensitive information not to be snooped during information storage, transmission, and computational process.

2) SP: Service provider in SaaS, provide data storage and application service.

3) Attacker: Any offensive person or group except data owner, including other Tenant.

Considering data processing procedure in SaaS, the attacks can be classified in 3 cases.

1) Data storage attack: Data stored in SP maybe stolen by attackers who attack the system to get privileges of authority, or obtained by the data administrator who take advantage of the privileges of authority.

2) Data processing attack: Business logic is caculated in the SP application servers. It could be obtained by attackers, or snooped by data administrator.

3) Data transmission attack: In SaaS, data will be transmitted through the Internet, and kinds of attackers may get data though the public network.

Problems in data transmission above are consistent with those in data confidentiality of other applied modals of Web. Many fully developed schemes such as SSL can be employed directly to solve these problems. And for data processing secure problems we can adopt the trusted calculating environment, and this is a assumed condition in this paper.

Our study mainly starts from solving problems of data confidentiality in data storage under the scene of typical SaaS application.

The intrinsic difference of data confidentiality between SaaS and DAS applications is that business logic operations in DAS applications are processed in the Tenant side, while these operations are calculated in SP side. Though the homomorphic encryption had been prompted, but it's still far away from practical application regarding of its efficiency and realization.

Therefore, another new idea is considered based on practical application, namely operating system and application service system selected as the reliable system, database management system selected as the general business system.

Assumed conditions in the passage:

1) Application server cannot be traced, namely based on trusted application system.

2) Database is not reliable, namely the condition that database is easy to suffer attacks and data administrator is not reliable and so on.

These assumed conditions in practical mean that there's no secret divulging in the mechanism even though the whole database was disclosed. It is helpful when SP is averse from divulging data, however, some data administrator may get personal profit by divulging the data, and attackers break though safeguard in the database to get data. So, these assumed conditions do have practical significance.

4 Main Security Framework

As shown in Fig. 1, our security framework based on a general SaaS application. The Trusted Third Party(TTP) undertakes the key management tasks of our confidential mechanism, in the practical application, it may be Certificate Authority or other Key Management Center based on PKI. Tenant and SP take the same roles defined in section 3.

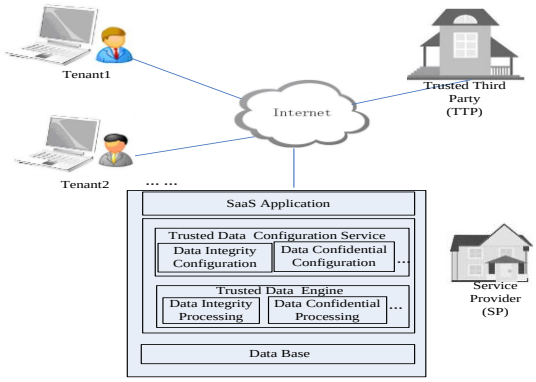

Fig. 1. Security Framework for SaaS

The mechanism includes key management protocols, secure data encryption key transmission and data encryption/decryption when Tenant taking the operation. Here are the main processes of our mechanism.

1) SP applies for its identity key from TTP.

Under the mechanism, SP must get the identity key used in later confidential process before it offers the services to tenants. The detail process is similar to the traditional certificate application, so we will not describe it in detail in this paper.

2) Tenant applies for its identity key and data encryption key from TTP.

Tenant should get its identity key and data encryption key before registering in the SP if it wants to protect its data stored at SP.

3) Tenant configures its confidential policy.

Through the configuration services provided by the mechanism Tenant can set up its confidential policy which determines the tables or attributes needs to be encrypted.

4) Key Agreements Protocol.

The system will invoke the Key Agreements Protocol when the Tenant login the application server, and the procedure will transfer the data encryption key securely to application server. In practical the data encryption key could be cached, and so get relative high performance.

5) Data Encryption/Decryption.

In the processing procedure, data need protected according to the policy will be encrypted transparently by confidential engine before being stored to physical storage, and decrypted from cipher after reading out from database.

5 Key Agreements and Algorithms

The key management in the initial phase is similar to traditional method under PKI, we will not describe it in detail here, and supposing that Tenant and SP all have obtained their own needed identity keys, and the data encryption key of Tenant here has been generated and stored at TTP. Here are the notations for this paper, in Table 1.

Table 1. Notations used in this paper

Notation	Comments
R^S	Denotes that relation R owned attributes need to be encrypted.
RA^e	Denotes that attribute A in R needs to be encrypted.
T^S	Denotes that tuple T owned encrypted attribute value.
TA^e	Denotes that the attribute value A in T is encrypted.
Ek()	Encrypt using key k, in symmetric encryption mechanism.
Dk()	Decrypt using key k, in symmetric encryption mechanism.
EPpk()	Encrypt using key pk, in asymmetric encryption mechanism.
DPsk()	Decrypt using key sk, in asymmetric encryption mechanism.
Ssk()	Sign using key sk, in asymmetric encryption mechanism.
Vpk()	Verify using key pk ,in asymmetric encryption mechanism.

5.1 Data Confidential Configuration Service

ConfidentialConfigure: Data Confidential Configuration Service supported by the mechanism, which tenants can dispose their confidential policy through it, and the confidential engine will protect data transparently base on the policy. The service can be expressed as:

Input: $R(A_1,\ldots,A_n)$;

Output: $R^S(A_1,\ldots A_i^e,\ldots)$, at least one attribute nominated to be Encrypted. and the service stores policy: $((R^S_{name1}, (A_1^e,\ldots,A_{m1}^e), \ldots)$.

5.2 Key Agreements Protocol

Without loss of practical nature, we suppose the initial keys used in the protocol had been dispatched to the Tenant, SP and TTP, as described below.

SP: Asymmetric key pair (sksp,pksp), Symmetric key (k_{sp}) used for encrypting the tenant data encryption key temporary.

Tenant: Asymmetric key pair (skti,pkti) for Tenant i.

TTP: Asymmetric key pair (skttp,pkttp), kti for Tenant i, which be used for encrypting data.

Base on the upper notations we give the detail description of the Key Agreements Protocol in Sequence diagram below. See Fig. 2.

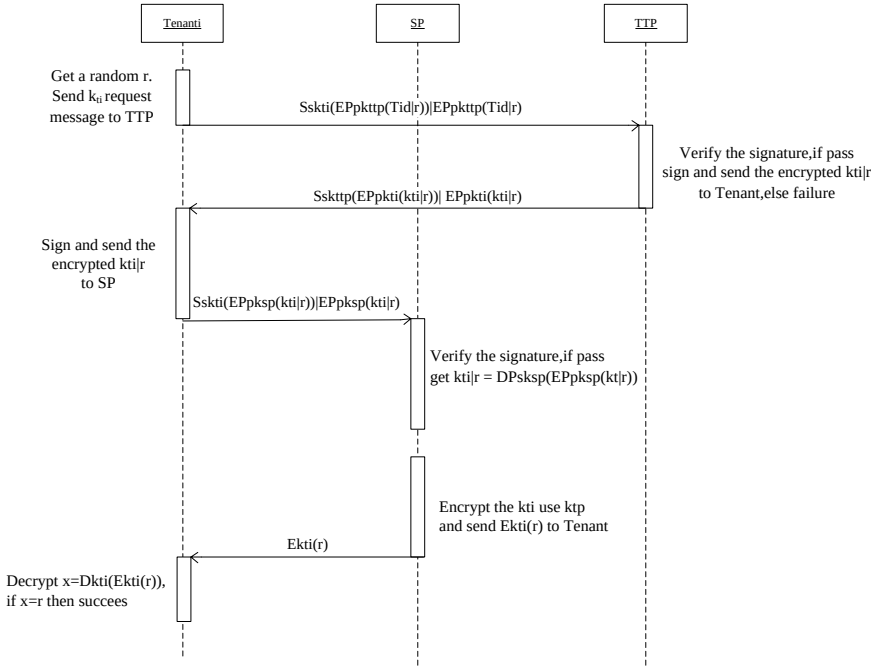

Fig. 2. Key Agreements Protocol for kti Transmission

As shown in Fig. 2, the k_{ti} is encrypted by destination public key in the protocol, and this can ensure its safety. The random number r is used to ensure the process' integrity, also to assure the k_{ti} is available after running the protocol. We use $Ek_{tp}(k_{ti})$ to substitute $EP_{pksp}(k_{ti})$ for higher performance. In the practical applications, the k_{ti} could be cached in the process procedure. This way can keep a higher efficiency.

5.3 Data Encryption/Decryption

In this mechanism, as shown in Fig. 1, the encrypt/decrypt runned by the confidential engine automatically, the detail process is transparent to the application, which imply that application developer can get higher developing efficiency under this Framework.

1) Data Encryption

Data is encrypted before the system storing it to database, and this also occurring before sending SQL for query to database management. The confidential engine invokes the encryption service according to the confidential configuration to encrypt

the attribute and substitute corresponding plaintext by the cipher in SQL text. The method is described as follows.

```
sConfig= getConfiditialConfig(Relation,i,SQLText, confidentialConfigure);
if sConfig == NULL then return else
            {
      for all Pj which R, {A_j^e} in sConfig do
               {
                  Cj=Ek_ti（Pj）;
                  Substitute(SQLText,Pj,Cj);
               }
             }
```

2) Data Decryption

Data is decrypted after the SQL result being returned from the database, and before application computing on data. The confidential engine invokes the decryption service according to the confidential configuration to decrypt the attribute and substitute corresponding cipher in SQL result set by the plaintext. The method is described as follows.

```
sConfig=getConfiditialConfig(Relation,i,SQLResult,
                                confidentialConfig);
if sConfig == NULL then return else
{
   for all Ck which R,{A^e} in sConfig do
     {
        Pk = Dk_ti (Ck);
        Substitute(SQLResult,Ck,Pk);
     }
}
```

6 Security Analysis

Considered the Key Agreements Protocol shown in Section 5, the sensitive data are only exposed in the memory of the application server, this case implies that the mechanism can ensure the data safety under the assumption that application services run in the trusted computing environment.

For the data encryption key k_{ti}, Fig 2 shows that the key is encrypted by the public key in asymmetric encryption mechanism in the whole transmission procedure, and only the message receiver can disclose it. Attacker can't get the key only if he can break the asymmetric encryption mechanism.

The sensitive data are encrypted in the physical storage, and the key used for encryption is not stored in physical storage of SP, so the data will be kept safety even if the Attacker can steal the cipher from database.

Above all, the mechanism this paper offered can ensure the sensitive data safety under the hypothesis we defined.

7 Performance Experiments

We designed two experiments to compare the cost between query over cipher and query over plain text, one for accurate query, and another for range query. The symmetric encryption mechanism in experiment is TEA.

The data scale in experiments we designed is in Table 2.

Table 2. Data Scale for Experiments.

No.	Type	Tenants	Tuples	Result scale
1	Accurate query	$10,10^2,10^3$	10^3	10
2	Range query	10^2	$10^2,10^3,10^4$	10

7.1 Experiments Configuration

The developing language is Java, the database is Oracle9i, the operating system is Windows Server2003. The memory of computer is 4G, CPU is Intel P8800(2.66GHz).

7.2 Experiments

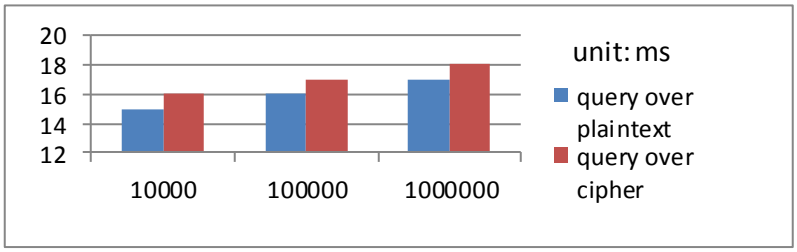

Fig. 3. Costs for Accurate Query

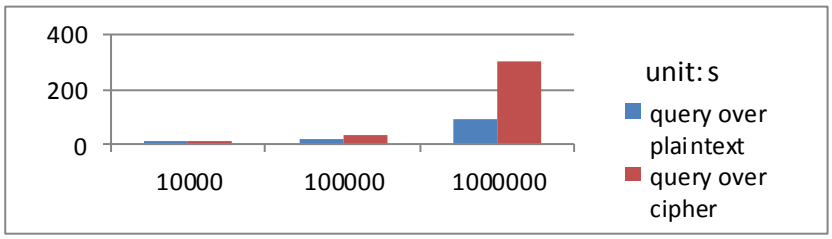

Fig. 4. Costs for Range Query

The experiment result shows that, the costs of the accurate query increase not so much, we can say that it is linear increment according to data magnitude. But for the range query, it shows a sharp increase in costs, the main reason is that the cipher has

been disordered entirely in traditional encryption we adapt, so we have to decrypt all the rows of table.

8 Conclusions and Future Works

In this paper, we give a general confidentiality-guarantee mechanism in practical SaaS applications, the main frame of the mechanism can also be easy used for data integrity protection and data privacy protection. We analyzed the security of our mechanism. The experiments result shows that it keeps high performance when used for the accurate query, but the effectiveness of the mechanism is invalidate when used in range query. Considered the strong security of the symmetric encryption mechanism the mechanism adopted, it fits for the critical and small-scale data, such as metadata and configuration data etc.

We will intend to develop the more practical encryption method in range query, and we will also pay close attention to data integrity and data privacy protection mechanism in our future works.

Acknowledge

The work is supported by the National Natural Science Foundation of China under Grant No. 61003253; the Natural Science Foundation of Shandong Province of China under Grant No.2009ZRB019YT , ZR2010FM031, ZR2010FQ010 and ZR2010FQ026; Key Technology R&D Program of Shandong Province under Grant No. 2010GGX10105.)

References

1. Hankan Hacigumus,Bijit Hore,Bala Iyer,Sharad Mehrotra. Search on Encrypted Data. IBM Search Report 2007, P385-425.
2. Feng DengGuo,Qin Yu. Research on Attestation Method for Trust Computing Environment. Chinese Journal of Computers. Vol. 31 No. 9 Sept. 2008.
3. Yuecel KaraBulut,Ike Nassi. Secure Enterprise Services Consumption for SaaS Technology Plataforms. IEEE International Conference on Data Engineering,2009.
4. Nestor Zwyhun. SaaS Data Security,Microsoft Report.
5. L. Bouganim, P. Pucheral. Chip-Secured Data Access: Confidential Data on Untrusted Servers In Proc, of VLDB 2002.
6. G. Aggarwal, M. Bawa, P. Ganesan, H. Garcia-Molina, K. Kenthapadi, U. Srivastava, D. Thomas, Y. Xu. Two Can Keep a Secret: A Distributed Architecture for Secure Database Services In Proc. of CIDR 2005.
7. E-J., GOH Secure Indexes. Technical report 2003/216, in IACR ePrint Cryptography Archive, (2003).
8. D. Song and D. Wagner and A. Perrig. Practical Techniques for Search on Encrypted Data. In Proc. of IEEE SRSP, 2000.

9. H. Hac~gumug, B. Iyer, and S. Mehrotra. Encrypted Database Integrity in Database Service Provider Model. In Proc. of Certification and Security in E-Services (CSES'O2). IFIP 1 71h World Computer Congress, 2002.
10. H. Hacigumug, B. Iyer, and S. Mehrotra. Providing Database as a Service. In Proc. Of ICDE, 2002.
11. B. Hore, S. Mehrotra, and G. Tsudik. A Privacy-Preserving Index for Range Queries. In Proc. of VLDB 2004.
12. Y. Chang and M. Mitzenmacher Privacy preserving keyword searches on remote encrypted data. In Third International Conference on Applied Cryptography and Network Security (ACNS 2005), volume 3531 of Lecture Notes in Computer Science, pp. 442-455. Springer-Verlag, 2005.
13. P. Golle, J. Staddon, B. Waters Secure conjunctive keyword search over encrypted data. In Applied Cryptography and Network Security (ACNS 2004), volume 3089 of Lecture Notes in Computer Science, pp. 31-45. Springer, 2004.
14. Chang Jie Guo,Wei Sun,Ying Huang, Zhi Hu Wang,Bo Gao A Framework for Native Multi- Tenancy Application Develoment and Management, The 9th IEEE International Conference on E-Commerce.
15. Sailer Reiner,Zhang Xiao-Lan,Jaeger Trent, Van Doorn Leendert. Design and implementation of a TCG-based intergrety mesurement architecture Proceeding of the 13th Usenix Security Symposium. San Diego,California, 2004：223－238
16. Sailer Reiner, Van Doorn Leendert,James P. Ward:The role of TPM in enterprise security. IBM Research Report RC23368,October 2004.
17. Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, Yirong Xu. Order Preserving Encryption for Numeric Data. SIGMOD 2004 June 1318,2004, Paris, France.
18. Wang Zhengfei,Wang Wei,Shi Bole. Efficient method of querying encrypt data. Computer Engineering and Applications, 2008,44(12):29-33.
19. Ma Lilin,Li hong. A Permission Model of SaaS System Base on RBAC. Computer Applications and Software. Vol. 27, no. 4, pp. 42-44. Apr 2010.