Multi-Tenant Data Storage Security In Cloud Using Data Partition Encryption Technique

K.Venkataramana, Prof.M.Padmavathamma

Abstract— Latest Computing models like Grid Computing, Cloud Computing aims of providing a reliable, cost-efficient delivery of service to Customers. Especially Cloud computing delivers service in areas like infrastructure, Platform and software through various delivery models such as Infrastrucutre-as-a-Service(IaaS), Platform-as-a-service(PaaS), Software-as-a-Service (SaaS), Data Storage as a Service(DSaaS).Cloud computing takes advantage through in terms of delivering service to multiple tenants on same physical machine through virtualization at the same time providing security to data of multiple tenants is a challenge. In this paper we focus on security regarding data storage for SaaS Applications or platforms that are built on Multi-Tenant Architectures (MTA), which allow expanding and modeling efficiently large data management structures, such as databases. In MTA each tenant on cloud may allot separate database due to security reasons, but it is not cost effective, so single database is shared. Sharing of database leads to unauthorized access of tenants data by another tenant which is serious breach of security so to address this problem we propose DPET(DATA PARTITION ENCRYPTION TECHNIQUE) which allows encryption of each record two times before storing in partition belonging to each tenant by public, private key pairs which is known to Client and CSP (cloud Service Provider) by using Generator based encryption technique. In this technique the Tenant storing data in the database partition is encrypted twice first by tenant himself(ER) encrypted Record and second time by CSP as EER(Encrypted ER) and at the time of decryption it is by private keys of tenant. so the record to be stored is not known to CSP also as ER is sent to it. The keys information is stored in Metadata table in each partition for each tenant which should be secured by CSP , thus the tenants data is secured in cloud environment. Each partition belonging tenant is secured by different pairs of keys.

Index Terms—Cloud Computing, Multi-tenancy, Multi-tenant architecture, security, Database storage, Partition.

1 INTRODUCTION

Cloud Computing is the modern computing technique which is based on Grid, Utility Computing, quickly adopted by organizations and businesses alike to help increase profit margins by decreasing overall IT costs as well as provide clients with faster implementation of services. Service are delivered to client as three major service models—SaaS, which includes Web services such as Yahoo Flickr, Google Docs, and Microsoft website designer. These Web services perform functions traditionally done with software installed on an individual computer. The second service module is platform as a service (PaaS). This model provides computing services as websites—such as mashups or the APIs of Google Maps—as well as file storage systems—such as Drop-box or box.net. The final service model is infrastructure as a service (IaaS). It includes business-to-business (B2B) services that are usually invisible to customers. The first service to reach the market was Amazon Elastic Compute Cloud (EC2)[1]. Currently the cloud computing industry has no standard business model, instead the companies are experimenting with four different ideas—private, community, public, and hybrid models.

Various characteristics of cloud such as elasticity, metered service, on-demand service, broad network access will allow services to be shared among different users or same organization or between different organization which leads to an important cloud element multitenancy[2]. Multi-tenancy spans the layers at which services are provided. In IaaS, tenants share infrastructure resources like hardware, computation servers, and data storage devices. With SaaS, tenants are sourcing the same application (e.g., Salesforce.com), which means that data of multiple tenants is likely stored in the same database and may share the same tables as shown in figure-1. When it comes to security, the risks with multi-tenancy must be addressed at all layers.

The impact of multi-tenancy is visibility of residual data or trace of operations by other user or tenant. The majority of the cloud service providers offer multitenancy to capitalize on the associated economies of scale which also translates into savings for the end user. In fact the competitive nature of cloud computing is such that cloud service providers have to minimize the total cost of ownership of their IT infrastructure, thus introducing multitenancy is a popular way to reducing total cost of ownership.

Multitenancy has made cloud computing popular by allowing businesses to benefit from reduced costs yet continue to gain access to data and applications within a cloud environment. Multitenancy is similar in nature to multiple families in the same condominium. In the multitenancy model, many users' data and resources are located in the same computing cloud, and are controlled and distinguished through the use of tagging for the unique identification of resources owned by individual user. In a typical multitenancy situation, the users are the tenants and are provided with a level of control in order to customize and tailor software and hardware to fit their specific needs. However, multitenancy introduces a unique set of security risks, which has yet to be fully acknowledged as a serious problem by policy makers and cloud service providers.Cloud security architecture must ensure one tenant does

- *K.Venkataramana is currently pursuingPh.D in Department of Computer Science, in S.V.University, Tirupati,AndhraPradesh,India, E-mail: ramanakv4@gmail.com*
- *Prof.M.Padmavathamma is Working as Head, Department of Computer Science, in S.V.University, Tirupati,AndhraPradesh,India,. E-mail: prof.padma@yahoo.com*

not have access to another tenant's resources, such as virtual machine (VM), network bandwidth, and storage. Each tenant must be securely separated using techniques such as access control, VLAN segmentation, and virtual storage controllers.

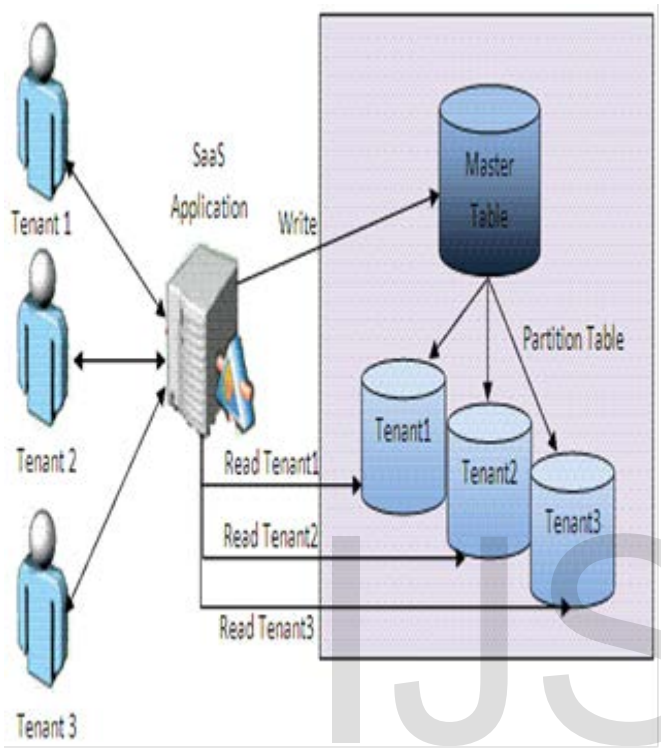

Fig-1 Multi-Tenancy in Cloud

This paper will explore the risks associated with multitenancy in section 2. Section 3 gives related work and back ground related to multi-tenancy and various security issues. In Section 4 we describe about Data Partition encryption technique for data storage in partitions of database, section 5 gives the results of DPET algorithm and concludes this paper in section 6.

2 MULTI-TENANCY SECURITY ISSUES

According to Armbrust and Fox (2010) and Feng et al (2011), the fundamental security issue with multitenancy is clients using Cloud Computing by employing single and the same computer hardware to share and process information. This presents a number of challenges in terms of compliance, security, and privacy (Bernardo and Hoang, 2010). The lack of user network isolation, In this technique, as the data stored by cloud applications used by tenants is on a same database but in a different partitions. So to ensure security for data of each tenant, in this paper we are proposing Data Partition Encryption Technique(DPET) in which each record in partition before stored is encrypted two times first by tenant's public key and by Cloud Service Provider (CSP), and decrypted only by tenant. Since the record 'R' to be stored at CSP is encrypted by tenant, 'R' cannot be revealed by CSP also. In this way the proposed DPET technique is secure by not revealing the record to other Tenants residing or using shared database.

In this technique, we assume $T_1, T_{2....}T_n$ are tenants in cloud C. Let Cloud service provider(CSP) provides sharing of database to multiple tenants $T_1, T_2, T_3 \ldots T_n$ by partitioning or segmenting the database' D'. In this paper we use Horizontal partitioning of Database so that each tenant T_i will use each partition of same database but with different encryption and decryption keys. The algorithm is given below
moreover, makes Cloud Computing vulnerable to threats, as does the lack of efficient bandwidth and traffic isolation, since malicious tenants may launch attacks to other tenants in the same cloud data centre. Existing approaches to access control on the clouds do not scale well to multi-tenancy requirements because they are based merely on individual user IDs.

The fundamental security issue with multitenancy is the very premise in which multitenancy is based upon; that is, multiple tenants sharing the same computer hardware. Indeed, using a multitenancy approach for the development of public cloud infrastructure presents a number of challenges in terms of compliance, security and privacy . One of the main challenges of using this form of multiple services is ensuring data isolation. Data management is critical as several users will be using the same system but all require privacy and confidently . Indeed multitenancy and lack of network isolation among tenants make the public cloud vulnerable to attacks [3] .

Side-channel attacks and interference among different policy domains pose daunting challenges in distributed clouds. Side-channel attacks are based on information obtained from physical implementation (for example, via time- or bandwidth-monitoring attacks). Side-channel attacks arise due to lack of authorization mechanisms for sharing physical resources. The interference among tenants exists primarily because of covert channels with flawed access control policies that allow unauthorized information flow.[4]

To provide "secure" multi-tenancy we should address the concerns of cloud skeptics, a mechanism to enforce separation at one or more layers are required [5]:

• **Application layer.** A specially written, multi-tenant application or multiple, separate instances of the same application can provide multi-tenancy at this level.

• **Server layer.** Server virtualization and operating systems provide a means of separating tenants and application instances on servers and controlling utilization of and access to server resources.

• **Network Layer.** Various mechanisms, including zoning and VLANs, can be used
to enforce network separation. IP security (IPsec) also provides network encryption at the IP layer (application independent) for additional security.

• **Storage Layer.** Mechanisms should be designed to protect tenants data by various encrypting techniques at partition level so that data of multiple tenants in partitions is secured.
Sharing software and data by multiple clients poses risks, such as Intellectual Property infringement, data infringement, and technical and industrial business sabotage (Bernardo, 2012). Cloud Computing providers therefore are responsible in ensuring that tenants cannot cross and access each other's hosted infrastructures. Lack of efficient bandwidth and traffic isolation makes multitenancy in cloud computing vulnerable, since malicious tenants may launch attacks towards co-resident tenants in the same cloud data centre [5].

Current approaches to access control on clouds do not scale well to multitenancy requirements because they are mostly based on individual user IDs [6]. By its very nature multitenancy has increased security risks due to the sharing of software and data by multiple tenants. As these collocated tenants may be competitors, if the barriers between tenants are broken down, one tenant may access another tenant's data or interfere with their applications. Indeed, cloud providers are responsible for ensuring that one customer cannot break into another customer's data and applications.

3. RELATED WORK

Various studies reveal security issues regarding multi-tenancy support, which is a prime focus of this paper. Multi-tenancy is the ability to efficiently deal with multiple administrative domains (tenants) that are using the same service which, in turn, has isolated resources belonging to particular tenants. In tenant based environment VM security , hypervisor security, platform security, Data storage security, user authentication, etc. to be considered seriously. The following are some of the threat vectors affecting the multi-tenant data center:

- Unauthorized access
- Interruption of service
- Data loss
- Data modification

In the case of software as a service (SaaS), multitenancy is almost always achieved via a database and configuration, with isolation provided at the application layer. So, at the application layer, service providers should design and implement a specific class, and then create an object of the class in a manner that serves the needs of multiple users in an effective way. Designing SaaS applications in this way will solve many issues related to multitenancy, such as the need for data security, data separation, and customized applications (to minimize the hard binding of runtime computing resources)[11].

In the article [7], discusses about approaches of managing multi-tenant data like isolated data storage and shared data storage for each tenant in tables, each approach has its own advantages and disadvantages but every approach opens up an security issue which should be solved by encryption techniques which we have proposed in this paper.

In paper by Mohamed Almorsy et.al TOSSMA[8] is based on externalizing security from the target applications. The tenants can impose security constraints to applications by an interface provided by Service provider without knowing its background where the tenant's security lies in hands of CSP.

Kamara et al.'s[9] work can be seen as the first contribution to cryptography-based cloud storage. In the Environment of public cloud infrastructure where the service provider is not completely trusted by customers, Kamara et al. designed secure cloud storage architectures for both consumer and enterprise scenarios by using non-standard cryptographic techniques, such as attributed encryption, searchable encryption, etc.

In his paper by Zarandioon et al. using attributed-based encryption and signature, proposed an user-centric privacy preserving cryptographic access control protocol called key to cloud (K2C). This protocol enables end-users to securely store, share, and manage their sensitive data in untrusted cloud storage anonymously [10].

Encryption is especially important in situations involving high-value data or privacy concerns, or when multiple tenants share the same set of database tables. The main aim is to obscure every tenant's critical data so that it will remain inaccessible to unauthorized parties even if they come into possession of it. By above study we proposed this algorithm for securing tenants data in Partitions of shared database.

4. DATA PARTITION ENCRYPTION TECHNIQUE (DPET):

In Mulit-tenant applicatins unlike IaaS where multiple tenants share resources, SaaS tenants share a database. Users of Salesforce.com or SmugMug, for instance, pay to use an application that manages their customers and photos respectively. While the value is in the application interfaces that make it easy to manage complex tasks and large data sets, the data itself is stored in a database as rows in tables that the tenants of Salesforce.com and SmugMug databases share. The customer ID is what distinguishes one row from the next. In this area, security concerns run high that misconfigured application code or an error in an access control list may put tenant information at risk of theft and misuse. For controlling access to database data, there are quite a few tools and technologies available. The new applications implemented is used for authentication and authorization of the access request so that only certain rows or fields are modifiable based on security policies that ensure that access is warranted. Encryption of data in the database is also common to protect it at rest, so that if it is ever compromised or stolen it would be difficult todecipher the underlying data.

So in this environment , as the data stored by SaaS cloud applications of tenants resides on a same database but in a different partitions security is in doubt. So to ensure security for data of each tenant, in this paper we are proposing Data Partition Encryption Technique(DPET) in which each record in

partition before stored is encrypted two times first by tenant's public key and by Cloud Service Provider (CSP), and decrypted only by tenant. Since the record 'R' to be stored at CSP is encrypted by tenant, 'R' cannot be revealed by CSP also. In this way the proposed DPET technique is secure by not revealing the record to other Tenants residing or using shared database.

In this technique, we assume $T_1, T_{2....}T_n$ are tenants in cloud C. Let Cloud service provider(CSP) provides sharing of database to multiple tenants $T_1, T_2, T_3 \ldots T_n$ by partitioning or segmenting the database' D'. In this paper we use Horizontal partitioning of Database so that each tenant T_i will use each partition of same database but with different encryption and decryption keys. In this algorithm the record is stored in encrypted format for security purpose, as the database is in cloud, to avoid unauthorized users to access data. The algorithm is given below

DPET TECHNIQUE

1. Tenant 'T_i' generates a large Prime T_p from his own credentials that has sent to Cloud Service Provider.
2. Tenant T_i computes $N=2*T_p$
3. Tenant T_i generates Cyclic group Z_N^* of order Ø(N)(Euler Totient function)
4. A subgroup $Z_{Ø(N)}^*$ subset of Z_N^* of order Ø(Ø(N)) is generated by T_i with generator $g \in Z_n^*$
5. Tenant T_i picks randomly picks up two private keys T_q and $T_r \in Z_N^*$ $T_q \equiv g^{k1} \bmod N$ and $T_r \equiv g^{k2} \bmod N$ where $k1,k2 \in Z_{Ø(N)^*}$ where g is generator for Z_N^*
6. Tenant T_i computes $N= T_q * T_r$
7. T_i chooses 'e' such that gcd(e, Ø(N))=1
8. T_i determines 'd' such that $ed \equiv 1 \bmod Ø(N)$
9. Tenant T_i computes $TP_r = e.rs_t$ such that $e.rs_t \equiv 1 \bmod Ø(N)$ and $TP_b = d.rs_d$ such that $d.rs_d \equiv 1 \bmod$ where TP_r : Tenant Private Key, TP_b: Tenant public key
   Public key<N,TP_b>
   Private key <TP_r,d,e>
10. Tenant T_i encrypts the data of each record R of database of his own partition P_i before sending to CSP and obtains encrypted record (ER) $ER = R^e \bmod n$
11. Tenant T_i sends ER_j to CSP to store in Partition P_i .
12. CSP stores ER in partition P_i of T_i after encrypting ER another time to obtain $EER=ER^{TP_b} \bmod n$
13. EER is stored in Partition P_i of Tenant T_i
14. When tenant T_i requests data from CSP hen CSP sends encrypted record EER to Tenant T_i.
15. After receiving Tenant T_i computes $R=EER^{rst} \bmod N$ to obtain original Record.
16. If Tenant T_i does not get Record R from above data then T_i assumes R is modified by CSP or intruder, so R is discarded and requests for fresh record.

5. EXPERIMENTAL RESULTS OF DPET

DPET algorithm is implemented in java and following results is obtained, the algorithm can be used for securing Tenants database record as following results are verified.

Record to be sent by Tenant :

CLOUD COMPUTING SECURITY

Prime chosen by the Tenant is Tp : 29
Tenant Computes n (=2*Tp) : 58
φ (n) : 12
g : 47
k1 : 19
k2 : 17
N=g^(k) :
f9e606b1a6b5a7b906946acf06476cafd55b2ff05b30ee5f41

TP_r (e * rs_t) (Tenant Private Key) : 3f 87 b8 ac 5c 88 ba 1d e3 63 63 a7 61 e2 bb f2 83 0c 94 25 2a fd 27 6c 12 9b 83 b5 3e 34 f0 6a bb

TP_b (d * rs_d) (Tenant Public Key) : 3f 87 b8 ac 5c 88 ba 1d e3 63 63 a7 61 e2 bb f2 83 0c 94 25 2a fd 27 6c 12 9b 83 b5 3e 34 f0 6a bb

rs_t : 7f 0f 71 58 b9 11 74 3c c4 e5 aa 00 35 e8 60 5e 8f e4 7c 4a c1 cb 0f 95 45

rs_d : 7f ff ff ff ff ff ff ff

Encrypted Record (ER) From Tenant:
700aafe0294ba0cd6ddd2295fdece87071adc166720c2d2a1d4f484 ccf0f008bfc426bedcbd7e0c58248143ef00e36c7b614a22e7f2dc2df 025642342d38f35b87af66a428e235d06e83f46fbb055c46291f6cfb5 440e6fb86da706af2926f173d523e2df5529e9002526b8bcc8089c34 eea1bd067603ea94ed45ad9179a31d319638d3c00bf16871552d67 ff88b19df56449e746c5700aafe0294ba0cd6ddd2295fdece87071a dc166720c2d2a1da22e7f2dc2df025642342d38f35b87af66a428e2 35d06e83f4be2c4a25c91c2cd1be34f7d5c6ba2ee6233284a0261708 49eb52a5cf17f30d319c38fd695dfb02f78be35ee39c596092cc746fb b055c46291f6cfb5440e6fb86da706af2926f173d523e2dc42170980 8cc403886379cf14511a0dc3bd0a11acd084fada2e41ff146d8166ed 5bdf97ab2842bd232c

Encrypted ER (EER) stored at CSP:
49eb52a5cf17f30d319c38fd695dfb02f78be35ee39c596092cc746fb b055c46291f6cfb5440e6fb86da706af2926f173d523e2dc42170980 8cc403886379cf14511a0dc3bd0a11acd084fada2e41ff146d8166ed 5bdf97ab2842bd232c80f20207c1b217f0a13f39f5bf928245ce3af36

fe91ff772443708634d8a2459288a72dc452831e829c4e9673eb9258
8c6f3c2ac8a6ce6a7aa1f179a31d319638d3c00bf16871552d67ff88
b19df56449e746c5e62ab55d88b3b240af4ed2b399dedbce5463d3
ec1c97f5abcb879754b79a8d298af8d0287dd01b26f87142afb7438
d6a6bb3700aafe0294ba0cd6ddd2295fdece87071adc166720c2d2
a1d6fbb055c46291f6cfb5440e6fb86da706af2926f173d523e2dbb6
01fc239c1407a4a554fd051c9ee6a4c6d55fce20cc8dc7ee41ff146d8
166ed5bdf97ab2842bd232c80f20207c1b217f0ac421709808cc4038
86379cf14511a0dc3bd0a11acd084fada2c6a41334a2d6b01ff448cc
f2e437c8c5688c5c8e4edf03a1e1

After Decryption at Tenant original record is obtained

PlainText :CLOUD COMPUTING SECURITY

6. SECURITY ANALYSIS

The proposed SQL commands for creating partitions with encryption and decryption mechanism is given below.

1.CREATE TABLE employ (empid NUMBER(6), ename NUMBER, doj Date) PARTITION BY RANGE (doj) (PARTITION emp_d1_2006 VALUES LESS THAN (TO_DATE('01-APR-2006','dd-MON-yyyy')) Public Key (g_{p1},p_1) PARTITION emp_d2_2007 VALUES LESS THAN (TO_DATE('01-APR-2007','dd-MON-yyyy')) Public Key (N,TP_b))

2 when insert command is given CSP will encrypt ER with its public key TP_b and stores into table.

3.Since the encrypted record is sent by tenant , cloud service provider cannot view the original record in partition also other tenants cannot view it as it is encrypted second time by CSP. Hence we are securing data of cloud applications by a stronger encryption technique.

7. CONCLUSION

Multitenancy in cloud is an inherent feature which provides various advantages in terms of resources usage in cloud environment, but at the risk of security. So in this paper we have proposed Data Partition encryption technique which allow tenants data to be secure when stored in partition of database at CSP. The algorithm is implemented and results are evaluated.

REFERENCES

[1] John Walz, David Alan Grier, "Time to Push the Cloud" ,IT Pro September/October 2010, IEEE Computer Society

[2] For a more in-depth discussion of security and legal issues in Cloud Computing, refer to "Security Guidance For Critical Areas Of Focus In Cloud Computing" V3.0, CSA from the Cloud Security Alliance at *http://www.cloudsecurityalliance.*

[3] K. Wood, M. Anderson, " Understanding the complexity surrounding multitenancy in cloud computing", *2011 Eighth IEEE International Conference on e-Business Engineering*, Vol. 1, no. , 119-124, 2011.

[4] Abdulrahman A. Al mutairi ,Walid G. Aref, et.al, "A Distributed Access Control Architecture for Cloud Computing",, IEEE SOFTWARE, IEEE COMPUTER SOCIETY,2012

[5] Paul Feresten, "Storage Multi-Tenancy for Cloud Computing", SNIA , March, 2010

[6]. W. Tsai, Q. Shao, " Role-Based Access-Control Using Reference Ontology in Clouds", *2011 Tenth International Symposium on Autonomous Decentralized Systems*, Vol. 11, no. 121-128, 2011

[7] [Website] Available at http://msdn.microsoft.com/en-us/library /aa479086.aspx#mlttntda_tde

[8] Mohamed Almorsy, John Grundy, and Amani S. Ibrahim,"TOSSMA: A Tenant-Oriented SaaS Security Management Architecture", IEEE Fifth International Conference on Cloud Computing, 2012

[9] S.Kamara,Kristin Lauter,"Cryptographic cloud storage", FC'10 Proceedings of the 14th international conference on Financial cryptograpy and data security Pages 136-149,Springer,2010

[10] Jose M. Alcaraz Calero, Nigel Edwards, Johannes Kirschnick, Lawrence Wil cock, and Mike Wray. "Toward a multi-tenancy authorization system for cloud services",IEEE Security and Privacy, pp48-55, 2010.

[11] Jinan Fiaidhi, Irena Bojanova, Jia Zhang, Liang-Jie Zhang, "Enforcing Multitenancy for Cloud Computing Environments", IT Pro, IEEE,2012.