International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 17 (2015) pp 37414-37420

Client Based Security for Cloud CRM Application

M.Vanitha
Architect, Software Development
Verizon Data Services India Pvt. Ltd. Chennai, India
vanitha_amara@yahoo.co.in

C.Kavitha
Assistant Professor, Department of Computer Science
Thiruvalluvar Govt. Arts College Rasipuram, India
kavithachellappan@yahoo.com

Abstract
Cryptography plays a very critical role in securing the cloud. Cloud Security Alliance identifies Data and Applications as two primary assets of cloud. The protection afforded to these data and application must comply with the importance and sensitivity of the information. Information protection includes confidentiality, integrity and authenticity. For data confidentiality and privacy, encryption can be used to protect data during transmission, processing, storage and deletion [1]. Processing is when data, such as the executable code or sensitive authentication tokens, are kept encrypted in the memory and are decrypted only when needed. Deletion is when these sensitivity authentication tokens and data should be deleted. Deletion stage is often overlooked [2], which is the complex stage of data in cloud. Data retention assurance may be easier for the cloud provider to demonstrate while the data destruction is extremely difficult. When the SLA between the customer and the cloud provider ends, today in no way it is assured that the particular customers' data is completely destroyed or destructed from the cloud provider's storage. The proposed method identifies way to track individual customers' data and their encryption keys and provides solution to completely delete the data from the cloud provider's multi-tenant storage architecture.

Keywords—Cryptography, Multi-tenant, Encryption, CRM, SFDC, META-Data, Cloud.

I. INTRODUCTION

There are many cloud providers in industry (Google, EMC, NetApp, Microsoft, IBM, and SalesForce.com that offers various cloud services. SalesForce.Com is one of the successful cloud providers that focus on three primary areas, "The Sales Cloud", "The Service Cloud" and "Your Cloud" [3]. SalesForce.Com is the leader in cloud computing Customer Relationship Management (CRM) application and its CRM combines sales cloud and service cloud. These cloud services are cost effective due to pay- as-you-go model and are flexible enough to expand the capacity without any limit. There are researches going on that explores the impact on security posture of moving to cloud, the transparency of cloud provider , how organizations are treating the line between trust and control with regard to encryption and the way the encryption keys needs to be managed. It is still an outstanding question on whose responsibility to manage the security standards- the cloud provider or the cloud consumer [4]. Security is more important in cloud because about 65% of the cloud customer store sensitive information in cloud.

II. DATA STORAGE IN CLOUD CRM

The cloud data storage architecture does not store all the data in just one location; instead they maintain multiple copies of same data across different physical as well as logical location for fault-tolerance reasons. Almost every provider in the industry today assures for the security of the customers data in all their storage spaces. But not all the providers assure that their data are securely deleted when the customer ends contract with the provider, including the back-up files. Especially when it comes to customers like government agencies and financial organizations, it is important that their data and its copies are made completely inaccessible to attackers. Because retention policies are becoming more and more important in almost all the enterprises that chooses cloud storage. It means setting retention policies when creating data for customers and deleting it when a customer says to do so. Unlike traditional storage system where deleting is just removing the pointer to the data, in cloud storage due to the fact that data is maintained across different servers and geographical locations it is important that we build a system that assures deletion of customer's data in the providers storage in a thorough manner.

A. Cloud Security a Survey

Based on survey from Thales e-Security & Ponemon Institute on Transfer of Sensitive or Confidential data to the cloud, only 11% of respondents say their organizations are not transferring sensitive or confidential information nor have no plan to use any cloud service. German companies have the highest transfer rate at 65% and Russian organizations have the lowest transfer rate at 38 %. Security in SaaS / PaaS / IaaS is a shared responsibility between the cloud provider as well as consumer. A slightly higher group of people say that in SaaS the responsibility of consumer is high when compared to the responsibility of the Provider. 59% says the sensitive and confidential data in the IaaS / PaaS cloud storage at rest is readable and not protected by encryption, masking, tokenization or suppression. Also 45% of surveyors say data at rest is readable. Only 40% of the organizations apply their own encryption in SaaS, IaaS and PaaS environments before

data is sent to the storage. Only 34% of surveyors believe that their organization is in control of encryption keys for the application level data as well as data at rest. Still large populations of cloud consumers' believe security is a shared responsibility of both the organization and the provider. On an average only 35% of cloud storage users say that they are aware of the steps taken by cloud provider to protect the sensitive data. This exposes the risk of sensitive data being stolen at rest. The Key Management Interoperability Protocol (KMIP) is very important in terms of managing Keys in cloud.

B. Data Life Cycle and Security Model in Cloud

In any database the life cycle of data is actually decided based on the frequency of data access. The data is accessed frequently it is said to be in Active state, and if it is not accessed at all for a pre-defined number of days, then the data is said to be in Archived state. Cloud Security Alliance includes six states in data life cycle that includes Create, Store, Use/Share, Archive, and Destruct. Almost all the cloud vendor takes care of all the states in data life cycle except the data destruction state. This state is often overlooked by the provider, but this one of the state of data that poses serious security threat in public cloud. Figure 1 illustrates the data life cycle in cloud

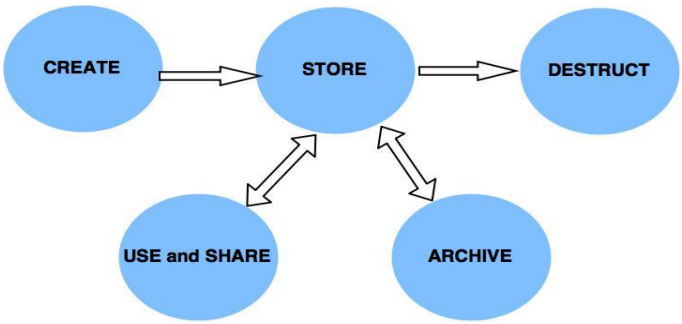

Figure 1.CPOD Architecture

C. Cloud CRM vs CRM

Before the concept of cloud computing even was developed, in traditional CRM systems mission critical industrial data are stored in the storage media internal to organization. Those data are protected by security measures such as firewalls, gateway servers to secure the data from accessing by unauthorized internal as well as external users. Whereas in Cloud based CRM applications, the storage / service providers owns the responsibility of protecting the date safely from unauthorized access. Than the intruders, the system administrators pose a serious threat in terms of unauthorized access. The service contract between the provider and the customer usually states the policies and practices to protect user's data. The contract usually comprises of service term, privacy / protection methodologies, regulations in terms of receiving and sharing data, change notification rules, scope of data protection etc. These are generally referred to as Service Level Agreements (SLA). Research says, the most of the SLAs (at least 80%) do not talk about data retention policies. Hence this research mainly concentrates on data retention.

D. SalesForce.com (SFDC)

SalesForce.com is the most popular and mostly used on-demand application development platform today supporting more than 55,000+ organizations. Enterprises and commercial business partners trust SFDC to develop robust and reliable, Internet-Scale applications. It has wide accessibility and independence, High back-up, less pricing, comparatively lower hardware costs and it has higher collaborative groups. Salesforce.com provides customer encrypted fields. It uses 128 bit Master Keys, Symmetric Key Encryption and Advanced Encryption Standard (AES). Application server manages Encryption and Decryption. SFDC administrators who has 'Manage Encryption Permissions' manages all kind of Key related activities such as creation, process and storage. However there is no mention about the deletion of these keys when the contract ends with the customer. SFDC encourages using encryption custom fields only if government regulations demands. The ownership / responsibility of data security lie in the hand of the customer than the cloud provider [3].

III. PROOF OF DELETION IN CRM APPLICATION

In this paper a CRM (Customer Relationship Management) application is described, and also an example to illustrate the proposed model is discussed. The CPOD (Cloud Proof of Deletion) system consists of three cloud components. The CPOD (comprises of CPOD Client, Policy Manager, Encryption / Decryption Engine), a storage system(Cloud database), and a CRM application (SFDC: SalesForce.com).The proposed method identifies way to track individual customer's data and their encryption keys and provides solution to completely delete the data from the cloud provider's multi-tenant storage architecture. It also ensures deletion of data copies as there are always possibilities of more than one copy of data being maintained for back-up purposes [7]. The data destruction proof shall also be provided to customer making sure that the owner's data is completely removed.

A. CPOD in CRM Application

The CPOD engine discussed her can be used in various types of data storage and the storage type considered in this paper is CRM (Sales force) objects. The application considered here is a Billing and ordering application that collects various customer sensitive information such as card number and ccv number from users and stores in the cloud data storage. In conventional non-cloud based CRM application, there is a single tenant, the owner himself who will have access to their data resource in single logical database. They store and retrieve data from their own database. If the user is trying to access data from his system, the query will reach one or many remote servers to gain access to the data. Database will also run the relevant query to get the relevant data back to the user. The data storage and retrieval is pretty transparent here, wherein only single party is involved. But in multitenant storage architecture, it is assumed that at least two tenants stores a portion of common data in the same database. But each tenant is given permission to access only his own data objects. Usually a tenant is assumed to have objects in plural

meaning first object (Account), second (Order), third (Billing) etc for each tenant. Each object will hold a unique identifier. So when a request comes in from the user associated with first tenant to either store or retrieve objects from database, this unique key is first retrieved from database to retrieve the appropriate object associated with the user of first tenant.

B. Meta-Data Driven Storage Model

Most of the multi-tenant applications use META-DATA driven architecture. The Meta data usually contains data about each tenant's data and customization. Force.Com delivers scalable, extraordinarily performable, customizable applications. Force.Com maintains Objects Meta Data table and Fields Meta Data table and the Data table itself. Every organization aka the tenant will have its own Org ID, and the objects the organization own will have Object ID (ObjID), and Object Name (ObjName). The fields Meta Data table will store all the fields tied to an object. Each field will be identified by Field ID. A Boolean field Is Indexed will be associated with every field to specify if the field is indexed or not. Every Field will also have a position identifier (FieldNum) to identify the placement of the field to store and retrieve data. These entire combined called as Force.Com's Universal Data Dictionary (UDD). As the objects and fields are maintained as Meta-Data's and not as conventional database structures and each tenants data is uniquely identified with GUID:ORGID:BUID:ObjID:FieldID combination. Because of this uniqueness this platform tolerates multi-tenant schema maintenance activities without affecting other tenants. Figure 2, illustrates the different types of tables involved in cloud storage [6].

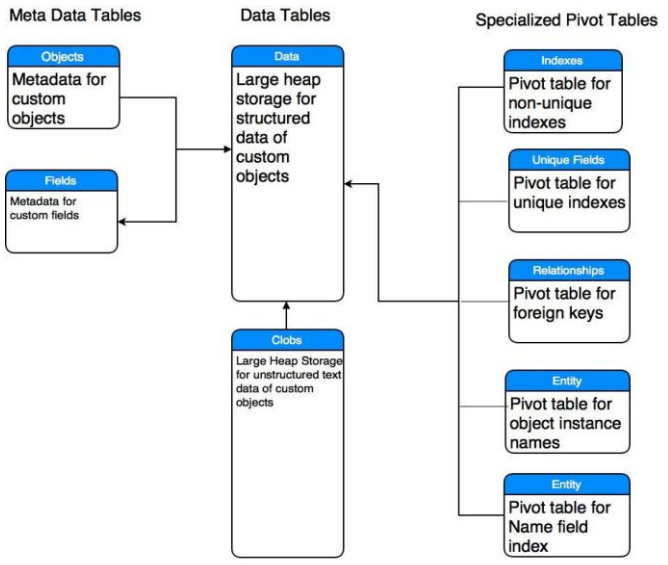

Figure 2.Tables in Cloud Storage

C. Sales Objects used in CPOD

The CRM application discussed in this paper will use following Sales Object. Contact, Account, Quote, Order and a customer object called 'Bill/Payment'. Accounts get added to the organization and each account is associated with one or more Contacts. Quote object maps with Accounts to create quote for that customer. Quote gets converted into an order. And once the order gets firmed it flows thru the Billing module for Secure Billing. Figure 3, illustrates the Sales Object Model of SalesForce application [3].

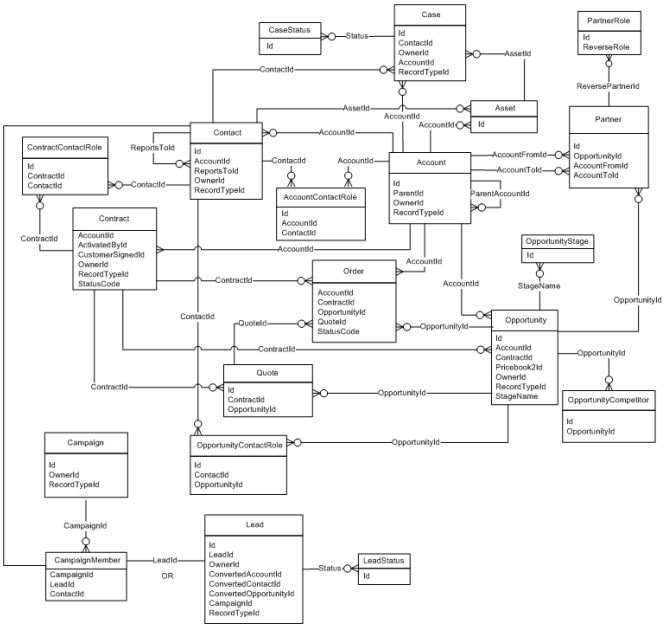

Figure 3.Sales Object Model

D. CPOD Architecture

CPOD is an application or system that provides proof of deletion for outsourced data in cloud storage. The following sections in this paper will describe about the components and the design of CPOD. The cloud customer wants to store their object in cloud database. After storing, they access their own data/objects based on the set of predefined policies. CPOD client is an application that should be deployed on top of the existing interface between the customer and provider to store and retrieve objects. CPOD protects the outsourced data in all the stages of life cycle including the last state, the deletion of data from the host. CPOD is composed of the following main components. Figure 4, illustrates the architecture of CPOD System.

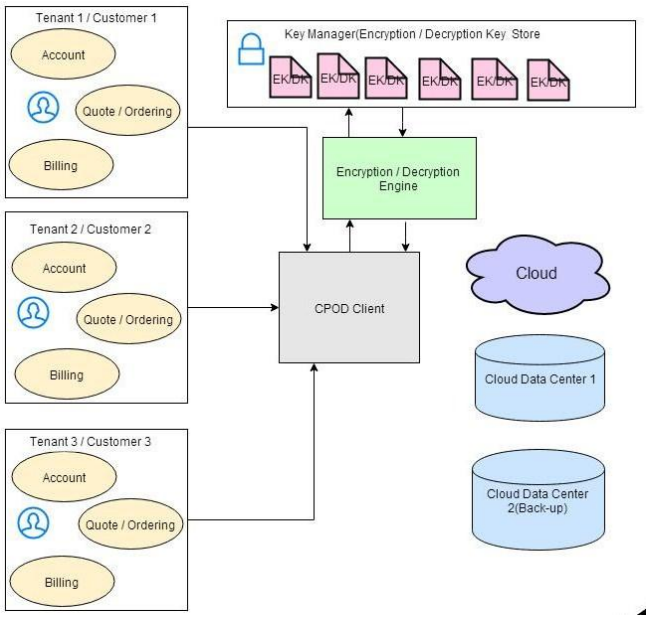

Figure 4.CPOD Architecture

E. ***CPOD Client***

CPOD Client will act as an interface between the customers and providers of cloud data storage. It is responsible for storing and retrieving objects from the cloud. It encrypts the objects before uploading it to the cloud and decrypts the objects after downloading it from cloud. As we all know, encryption is always required to keep the data objects in the cloud safe from malicious attacks and from unauthorized users. For the purpose of encryption and decryption the client interacts with Key managers (The system that maintains encryption/decryption keys for the data stored in cloud) based on the file policy. We assume that if the keys are destroyed it is not possible to recover the encrypted data.

F. ***Policy Manager***

Since cloud storage is provided by third party vendors the entire data storage in cloud should be transparently encrypted to protect confidential data from attackers with physical access to the computer. Each and every object that a customer stores should follow a set of predefined file access policies. The object attributes are specified in the policy file. So when decrypting the data object when the policy attributes are satisfied decryption will work. The other important fact that policy file will also store the access control list, list of authorized users who can access the files. The rules specified here are very expressive which specifies which private key can decrypt which cipher text. The definition of polices varies with application and the requirement. In our work, each policy file is linked to a Key Encryption Key. KEK is stored within the Key Manager (as key fragments). Each data object is associated to or owned by a user is achieved by revoking all the policy files liked to all customer data objects. When Revoke Policy files (Customer) is invoked, the corresponding KEKs are removed from the Key Manager. Without the KEK, the Symmetric Data Key cannot be derived; hence the underlying data file which was originally encrypted using Symmetric Data Key cannot be recovered forever. This stands as the Proof of Deletion in Cloud.

G. ***Types of Keys***

Keys in general are nothing but a share secret between two parties, the customer and the provider. Key managers stores the set of keys associated with files stored in clued. Key Manager maintains three kinds of keys. Key Manager takes care of generating, storing, exchanging and replacing the keys between the CPOD client and Provider exchange.

TABLE I.

Key Notations	**Meaning**
Key Notations	*Meaning*
DO	*Data Objects*
Fi	*Field Fi*
D	*Symmetric Data Key*
PFi	*Policy File*
R1i, R2i	*Random Generated Keys*
(ci, di)	*RSA public / private control key for Policy File PFi*

Key Notations used in this paper

Following are the different type of keys managed by key manager.

1) *Symmetric Data Key(SDK):* A symmetric data key is a random secret key that is generated and maintained by a CPOD client. SDK is used for decrypting or encrypting data objects via symmetric key encryption algorithm..
2) *Key Encryption Key(KEK):* The Key Encryption Key is a public/private key pair. Each Policy is associated with a Key Encryption Key pair. The Public key is shared with the customer; whereas the private control key is managed and stored secretly by the key managers. The public keys are distributed through a signed public key certificate. The data keys of the objects with same policy can be encrypted or decrypted with the help of KEK. The public/private key forms the basis of secured data deletion in cloud storage..
3) *Policy Access Key(PAK):* Similar to KEK, the policy key is also a public/private key and represents a particular policy. Policy Access Key as the name states it is associated with each policy. The private keys are associated with set of attributes or labels. Objects are encrypted under the policy over these attributes. Objects can be decrypted only when the attributes provided by the user satisfies the policy.

The whole idea behind the concept of keys is not just for securing the files stored over cloud, but also to make sure if these keys are deleted with the object associated with it are irrecoverable which forms the basis for proof of deletion.

IV. OBJECT STORAGE/ RETRIEVAL

A. Object Structure

Every object to be stored in cloud will have a unique id to store and retrieve from cloud data base, a user Meta data which uniquely identifies the data of a particular customer in case of multitenant database architecture, a metadata which is nothing but the information / data about the data that the object holds and the data itself. Every user / tenant trying to store data into the cloud will be allocated with containers or buckets. They are nothing but unique flat name space identified by a unique identifier. Objects specific to a tenant are stored in their own container. Cloud applications store, retrieve, and delete objects into this container with the help of REST APIs. Each Object will ties to its own list of fields and they are stored in the extension table but are closely tied to the parent object through the unique id.

B. CRM Process Flow

Any tenant as explained in Figure 3 CPOD Architecture trying to store data as objects into cloud storage should encrypt them prior to storing to ensure that Object / Data is safe. Especially the cloud objects discussed here such as Account / Quote & Ordering / Billing contains sensitive / critical information which when disclosed will create security with legal issues to both the cloud provider and the customer aka tenant. The order of Object creation in CRM is 1. Account 2. Quote 3. Order 4. Billing. Account is tied with Quote, Order and Billing Objects with the unique identifier Account ID. When a opportunity is WON, a quote is created for the account. When contract signed, quote converts into an Order and then it is billed. All these Objects when to be stored and retrieved from database passes through the CPOD Client. CPOD Client takes care of encrypting while storing and decrypting while retrieving. This CPOD is installed / written on top of existing object storage business logic in the client side, so that the customer / tenant has the complete control over the Keys and the Policy files. When the tenant decides to end the contract, the tenant can or will be able to delete the encryptions keys, so that the object remains inaccessible by anyone around the world including the cloud provider himself.

C. Storage and Retrieval Operations

1) Encrypt and Store Objects

i. The CPOD client want to store object DOi to the cloud. First the client requests the public key (ci, di) of the Key Encryption Key from Key manager for the File Policy PFi.

ii. The client then generates the random key (Symmetric key).

iii. The client then requests the Key manager to encrypt. The DOi.Fi (Field1) with the symmetric key and captures the value in the local cache / variable. The same process is continued for all the fields under the objects. Consider we are storing the Billing Object DO (Billing), it is expected to have Billing.AccountID, Billling.CardNumber, and Billing.SSN. These are assumed to be sensitive fields and all the fields and this object are individually encrypted with user based policy and only encrypted fields are moved to cloud for storage.

2) Decrypt and Retrieve Objects

When the Objects needs to be retrieved to the UI or needed to be processed along with other objects in the business layer the following steps are followed. The decision of what fields on objects to be managed via CPOD client are managed in Policy Files. Not all the fields in an object are sensitive and hence all of them are need to undergo client bases encryption. Research shows an impact in performance when all the fields following the encryption and decryption logic.

i. The CPOD client wants to retrieve object DOi.Fi from the cloud

ii. First the client fetches the random symmetric key (ci, di) and the encrypted Field from cloud. The field can be identified using the unique ID of the object. This unique Id is not encrypted and already available in the client side.

iii. The client then generates the secret random and requests the key manager for decryption (Symmetric key).

iv. The client needs to present all the required credentials that satisfy the attributes set in the associated Policy FPi. Policy Access Key corresponds to the set of attributes that client satisfies [8].

v. The CPOD Client then decrypts the DOi.Fi and passes it back to UI / Business layer.

3) Delete Objects

i. When the policy files associated with the customer is revoked, the Key Manager Keys are deleted automatically. Now even the CPOD client program cannot be used to recover the files associated with the particular file policy [8]. Deletion of keys is achieved via secure overwriting as explained in section IV [10].

ii. When Revoke Policy Files is invoked, the corresponding KEKs are removed from the Key Manager. Without the KEK, the Symmetric Data Key cannot be derived; hence the underlying data file which was originally encrypted using Symmetric Data Key cannot be recovered forever. This stands as the Proof of Deletion in Cloud.

iii. Deleting the KEK tied to the fields should be only attempted when the Contract/SLA ends. If they are accidently deleted, they cannot be retrieved forever.

iv. This deletion can also be time based. The time to delete can be automated by mentioned through the policy file. The time would be usually the end of the contract term, after which the sensitive data cannot be retrieved even by the customer himself.

V. KEY STORE OPERATIONS

Our goal is to maintain n key shares of a Key Encryption Key in such a way that Knowledge of any k or more key fragments makes the files easily computable; Knowledge of any k-1 or fewer key fragments leaves the files completely undetermined.

i. PUT Operation: The put operation is provided with an encryption key by the CPOD Client. The Key

Distributor partitions the key into n key shares using the key distribution protocol, and determines the destination fragment stores.

ii. GET Operation: For the get operation, the Key Assembler determines the fragment stores using the key distribution protocol and requests them for the key shares. Under normal condition, Key Assembler should retrieve at least k key fragments to reconstruct the encryption key. The key fragments retrieved are then used in for key reconstruction. The operation returns null, if an insufficient number of fragments is retrieved, or if no such encryption key is available.

iii. DELETE Operation: When a key is deleted using the delete operation, at the encryption key store determines the destination fragment stores where the key fragments are supposed to be stored across availability zones and regions, and requests the fragment store to delete the fragments. If at least nk+1 fragment stores report successful deletion of the requested fragments, the encryption key store returns success [5].

VI. SECURE OVERWRTING

Deleting data does not delete it from the disk. Usually, deleting mean that only the pointers to the data are deleted and the space can be reused by just overwrite on the same storage area. Until the space is overwritten, the data is not removed from the disk and can be read by other programs. This can be avoided by the 3-pass secure overwriting technique mentioned here. Given an n-k+1 Key Encryption Key fragments the secure overwriting function takes the Most Significant Bit of all the bytes, and overwrites it with 1 in the first pass. In the second pass the MSB will be overwritten with a random number (either 0/1). In the third pass the MSB will be overwritten with zero. This will make the bytes/fragments useless. Table II and Table III shows how the 3 Most Significant Bits have been changed before and after the secure overwriting process works on it. The secure overwriting is invoked on the Key Managers to which the Key Distributor/Assembler is connected to; it works until each and every data in the row is systematically and comprehensively rendered useless. A secure deleting mechanism for encrypted files is introduced in [9]. The party who does the deletion is trusted to delete the encryption key but is not allowed to see the key even in memory. This is achieved via a standard commutable blinding function [10].

VII. IMPLEMENTATION

CPOD Client is implemented Sales force Developer edition using Force.com. Key Storage is implemented using Oracle 10g. Encryption and Decryption algorithm is implanted using APEX inbuilt Symmetric Key Algorithm and Public Key encryption algorithm. Encrypted data are stored to the sale force database. CPOD Client interacts with cloud as follows. It is assumed that any customer who wishes to store sensitive objects / fields to cloud installs CPOD client in their machine.

VIII. EXPERIMENT AND RESULTS

In this section, we conducted a performance study on the CPOD Client. Our goal is to evaluate the performance overhead of Object storage and assured deletion. We explore the overhead from three perspectives:

i. Number of objects,
ii. Key generation time,
iii. Storage to cloud DB time.

Performance study is made by running n number of tenants across different CPOD clients simultaneously trying to store the Billing Object and three most sensitive fields in it. The following table captures the time taken for different actions throughout the file upload process.

TABLE II. **PERFORMANCE RESULTS**

# Users	# Objects	Key Gen (Secs)	Encry (Secs)	Storage (Secs)	Total Time (Secs)
1 Dev Account	1 BO, 3 Fields	0.1	0.2	1	1.4
5 Dev Account	3 Bo, 15 Fields	0.3	.12	4.5	4.65

IX. CONCLUSION AND FUTURE WORK

In this paper, we present the design and implementation of CPOD client on top of a cloud CRM. This client provides secure and cost effective assured deletion of sensitive information on the cloud storage. Current study is conducted with a developer edition that limits the number of user. The execution result shows that the time taken for the complete object storage / retrieval process may increase exponentially then number of users and their number of fields increases. Our future work will be to reduce the time taken for key generation and policy file generation processes in order to make CPOD work efficiently when working across network and storing files in third party cloud storage systems.

X. ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my supervisor Dr. C. Kavitha, who gave me all her support and encouragement to do a lot of research in cloud storage. The valuable suggestions based on multiple periodic reviews helped to improve the quality of this paper.

XI. REFERENCES

[1] M. Armbrust, A. Fox, R.Griffith, A.D. Joseph, R.Katz, A.Konwinski, G.Lee, D.Patterrson, A. Rabkin, I. Stoica, and M.Zaharia, "A View of Cloud Computing." Comm. ACM, vol.53, no.4, pp. 50-58, Apr. 2010.

[2] https://cloudsecurityalliance.org/

[3] www.salesforce.com/

[4] B. Scheneier, “File Deletion,” http://www.schneier.com/, Sept. 2009.
[5] Rivest, R., Shamir, A., and Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. Comm. A CM 21, 2 (Feb. 1978), 120-126.
[6] http://www.google.com/patents/US8095531
[7] M. Vrable, S. Savage, and G.M. Voelker, “Cumulus: Filesystem Backup to the Cloud,” ACM Trans. Storage, vol. 5, no. 4, article 14, Dec. 2009.
[8] Brent Waters, “Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization, “http://eprint.iacr.org/2008/290.pdf.
[9] Peter. Gutmann, "http://en.wikipedia.org/wiki/Gutmann_method"
[10] R. Perlman, “Secure Deletion of Data,” in Proc. 2005 IEEE International Security in Storage Workshop.