International Journal of Recent Trends in Engineering & Research

ISSN (ONLINE) : 2455 – 1457

IMPACT FACTOR : 4.101

IMPLEMENTATION OF A HYBRID CLOUD APPROACH FOR SECURE AUTHORIZED DEDUPLICATION

APPANA NAGA DURGARAO[1], DAKINENI DURGAPRASAD[2]

[1]MTech Scholar, [2]Assistant Professor

[1,2]Department of Computer Science Technology, Baba Institute Of Technology And Sciences
Bakkannapalem, Madhurawada, Visakhapatnam, Andhra Pradesh -530048

***Abstract*: -**Data deduplication is one of important data compression techniques for eliminating duplicate copies of repeating data, and has been widely used in cloud storage to reduce the amount of storage space and save bandwidth. To protect the confidentiality been proposed to encrypt the data before outsourcing. To better protect data security, this paper makes the first attempt to formally address the problem of authorized data deduplication. Different from traditional deduplication systems, the differential privileges of users are further considered in duplicate check besides the data itself. We also present several new deduplication constructions supporting authorized duplicate check in hybrid cloud architecture. Security analysis demonstrates that our scheme is secure in terms of the definitions specified in the proposed security model. As a proof of concept, we implement a prototype of our proposed authorized duplicate check scheme and conduct testbed experiments using our prototype. We show that our proposed authorized duplicate check scheme incurs minimal overhead compared to normal operations.

Index Terms: - Deduplication, authorized duplicate check, confidentiality, hybrid cloud.

I. INTRODUCTION

To make data management scalable in cloud computing, deduplication has been a well-known technique and has attracted more and more attention recently. Data deduplication is a specialized data compression technique for eliminating duplicate copies of repeating data in storage. The technique is used to improve storage utilization and can also be applied to network data transfers to reduce the number of bytes that must be sent. Instead of keeping multiple data copies with the same content, deduplication eliminates redundant data by keeping only one physical copy and referring other redundant data to that copy. Deduplication can take place at either the file level or the block level. For file level deduplication, it eliminates duplicate copies of the same file. Deduplication can also take place at the block level, which eliminates duplicate blocks of data that occur in non-identical files. Although data deduplication brings a lot of benefits, security and privacy concerns arise as users' sensitive data are susceptible to both insider and outsider attacks .Traditional encryption, while providing data confidentiality, is incompatible with data deduplication. Specifically, traditional encryption requires different users to encrypt their data with their own keys. Thus, identical data copies of different users will lead to different cipher texts, making deduplication impossible. Convergent encryption has been proposed to enforce data confidentiality while making deduplication feasible. It encrypts/decrypts a data copy with a *convergent key*, which is obtained by computing the cryptographic hash value of the content of the data copy. After key generation and data encryption, users retain the keys and send the cipher text to the cloud. Since the encryption operation is deterministic and is derived from the data content, identical ldata copies will generate the same convergent key and hence the same cipher text. To prevent unauthorized access, a secure proof of ownership protocol is also needed to provide the proof that the user indeed owns the same file when a duplicate is found. After the proof, subsequent users with the same file will be provided a pointer from the server without needing to upload the same file. A user can download the encrypted

DOI : 10.23883/IJRTER.2017.3447.JRBFM

file with the pointer from the server, which can only be decrypted by the corresponding data owners with their convergent keys. Thus, convergent encryption allows the cloud to perform deduplication on the cipher texts and the proof of ownership prevents the unauthorized user to access the file.

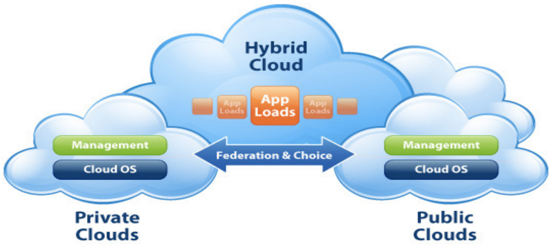
II. LITERATURE SURVEY

In archival storage systems, there is a huge amount of duplicate data or redundant data, which occupy significant extra equipments and power consumptions, largely lowering down resources utilization (such as the network bandwidth and storage) and imposing extra burden on management as the scale increases. So data de-duplication, the goal of which is to minimize the duplicate data in the inter level, has been receiving broad attention both in academic and industry in recent years. In this paper, semantic data de-duplication (SDD) is proposed, which makes use of the semantic information in the I/O path (such as file type, file format, application hints and system metadata) of the archival files to direct the dividing a file into semantic chunks (SC). While the main goal of SDD is to maximally reduce the inter file level duplications, directly storing variable SCes into disks will result in a lot of fragments and involve a high percentage of random disk accesses, which is very inefficient. So an efficient data storage scheme is also designed and implemented: SCes are further packaged into fixed sized Objects, which are actually the storage units in the storage devices, so as to speed up the I/O performance as well as ease the data management. Primary experiments have demonstrated that SDD can further reduce the storage space compared with current methods .. With the advent of cloud computing, secure data deduplication has attracted much attention recently from research cloud storage to reduce the storage size of the tags for integrity check. To enhance the security of deduplication and protect the data confidentiality, Bellare et al. showed how to protect the data confidentiality by transforming the predicatable message into unpredicatable message. In their system, another third party called key server is introduced to generate the file tag for duplicate check. Stanek et al. presented a novel encryption scheme that provides the essential security for popular data and unpopular data. For popular data that are not particularly sensitive, the traditional conventional encryption is performed. Another two-layered encryption scheme with stronger security while supporting deduplication is proposed for unpopular data. In this way, they achieved better trade between the efficiency and security of the out-sourced data. Liet al. addressed the key management issue in block-level deduplication by distributing these keys across multiple servers after encrypting the files.

III. OVERVIEWOF THE HYBRID CLOUD CONCEPTS

3.1 HYBRID CLOUD

A hybrid cloud is a cloud computing environment in which an organization provides and manages some resources in-house and has others provided externally .For example, an organization might use a public cloud service, such as Amazon Simple Storage Service(Amazon S3) for archived data but continue to maintain in house storage for operational customer data



The concept of a hybrid cloud is meant to bridge the gap between high control, high cost “private cloud” and highly callable,flexible, low cost “public cloud”. “Private Cloud” is normally used to describe a VMware deployment in which the hardware and software of the environment is used and managed by a single entity. The concept of a “Public cloud” usually involves some form of elastic/subscription based resource pools in a hosting provider datacenter that utilizes multi-tenancy. The term public cloud doesn’t mean less security, but instead refers to multi-tenancy.

The concept revolves heavily around connectivity and data portability. The use cases are numerous: resource burst-ability for seasonal demand, development and testing on a uniform platform without consuming local resources, disaster recovery, and of course excess capacity to make better use of or free up local consumption.

VMware has a key tool for “hybrid cloud” use called “vCloud connector”. It is afree plugin that allows the management of public and private clouds within the vSphere client. The tool offers users the ability to manage the console view, power status, and more from a “workloads” tab, and offers the ability to copy virtual machine templates to and from a remote public cloud offering.

IV. HYBRID CLOUD FOR SECURE DEDUPLICATION

At a high level, our setting of interest is an enterprise network, consisting of a group of affiliated clients (for example, employees of a company) who will use the S-CSP and store data with deduplication technique. In this setting, deduplication can be frequently used in these settings for data backup and disaster recovery applications while greatly reducing storage space. Such systems are widespread and are often more suitable to user file backup and synchronization applications than richer storage abstractions. There are three entities defined in our system, that is, *users*, *private cloud* and S-CSP in *public cloud* . The S-CSP performs deduplication by checking if the contents of two files are the same and stores only one of them. The access right to a file is defined based on a set of *privileges*. The exact definition of a privilege varies across applications. For example, we may define a *rolebased* privilege according to job positions (e.g., Director, Project Lead, and Engineer), or we may define a *time-based* privilege that specifies a valid time period (e.g., 2014-01-01 to 2014-01-31) within which a file can be accessed. A user, say Alice, may be assigned two privileges “Director” and “access right valid on 2014- 01-01”, so that she can access any file whose access role is “Director” and accessible time period covers 2014-01- 01. Each privilege is represented in the form of a short message called *token*. Each file is associated with some *file tokens*, which denote the tag with specified. A user computes and sends *duplicate-check tokens* to the public cloud for authorized duplicate check. Users have access to the private cloud server, a semitrusted third party which will aid in performing deduplicable encryption by generating file tokens for the requesting users. We will explain further the role of the private cloud server below. Users are also provisioned with per-user encryption keys and credentials

A. Architecture For Authorized Deduplication:

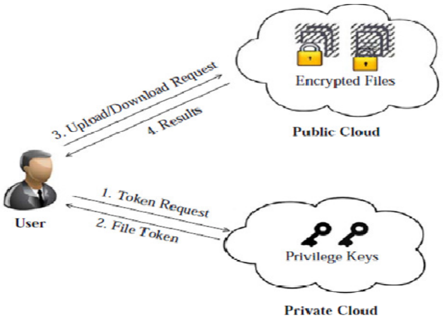


Fig.2 Architecture for Authorized deduplication

In this paper, we will only consider the file level deduplication for simplicity. In another word, we refer a data copy to be a whole file and file-level deduplication which eliminates the storage of any redundant files. Actually, block-level deduplication can be easily deduced from file-level deduplication, specifically, to upload a file, a user first performs the file-level duplicate check. If the file is a duplicate, then all its blocks must be duplicates as well; otherwise, the user further performs the block-level duplicate check and identifies the unique blocks to be uploaded. Each data copy (i.e., a file or a block) is associated with a token for the duplicate check.

- *S-CSP.* This is an entity that provides a data storage service in public cloud. The S-CSP provides the data outsourcing service and stores data on behalf of the users. To reduce the storage cost, the S-CSP eliminates the storage of redundant data via deduplication and keeps only unique data. In this paper, we assume that S-CSP is always online and has abundant storage capacity and computation power.
- *Data Users.* A user is an entity that wants to outsource data storage to the S-CSP and access the data later. In a storage system supporting deduplication, the user only uploads unique data but does not upload any duplicate data to save the upload bandwidth, which may be owned by the same user or different users. In the authorized deduplication system, each user is issued a set of privileges in the setup of the system. Each file is protected with the convergent encryption key and privilege keys to realize the authorized deduplication with differential privileges.
- *Private Cloud.* Compared with the traditional deduplication architecture in cloud computing, this is a new entity introduced for facilitating user's secure usage of cloud service. Specifically, since the computing resources at data user/owner side are restricted and the public cloud is not fully trusted in practice, private cloud is able to provide data user/owner with an execution environment and infrastructure working as an interface between user and the public cloud. The private keys for the privileges are managed by the private cloud, who answers the file token requests from the users. The interface offered by the private cloud allows user to submit files and queries to be securely stored and computed respectively.

Notice that this is a novel architecture for data deduplication in cloud computing, which consists of a twin clouds (i.e., the public cloud and the private cloud). Actually, this hybrid cloud setting has attracted more and more attention recently. For example, an enterprise might use a public cloud service, such as Amazon S3, for archived data, but continue to maintain in-house storage for operational customer data. Alternatively, the trusted private cloud could be a cluster of virtualized cryptographic co-processors, which are offered as a service by a third party and provide the necessary hardware based security features to implement a remote execution environment trusted by the users.

V. ALGORITHMS USED

In this section, we use two types of algorithms,
1). For file uploading.
2). For file downloading.

FOR UPLOADING A FILE

```
BEGIN
Step –1 Read file
Step –2 Cloud server checks for duplication
Step –3 Sends duplication response whether the file already exists or not
Step – 4 If the file does not exist
4.1 Display “file does not exist”
Step – 5 Then it uploads the file
Step – 6 If the file already exist
6.1 Display “file already exist”
END
```

FOR DOWNLOADING A FILE

```
BEGIN
Step –1 Read file
Step –2 Cloud server checks for duplication
Step –3 Sends duplication response whether the file already exists or not
Step –4 If the file exist –
4.1 Display “file exist”
Step –5 then it downloads the file
Step –6 If the file does not exist –
6.1 Display “file does not exist”
END
```

VI. IMPLEMENTATION

We implement a prototype of the proposed authorized De-duplication system, in which we model three entities as separate C++ programs. A *Client* program is used to model the data users to carry out the file upload process. A *Private Server* program is used to model the private cloud which manages the private key and handles the file token computation. A *Storage Server* program is used to model the S-CSP which stores and de-duplicates files. We implement cryptographic operations of hashing and encryption with the OpenSSL library [1].We also implement the communication between the entities based on HTTP, using GNU Libmicrohttpd [10] and libcurl [13]. Thus, users can issue HTTP Post requests to the servers. Our implementation of the **Client** provides the following function calls to support token generation and de-duplication along the file upload process.

- *FileTag(File)* – It computes SHA-1 hash of the File as File Tag;
- *TokenReq(Tag, UserID)* – It requests the Private Server for File Token generation with the File Tag and User ID;
- *DupCheckReq(Token)* – It requests the Storage Server for Duplicate Check of the File by sending the file token received from private server;
- *ShareTokenReq(Tag, {Priv.})* – It requests the Private Server to generate the Share File Token with the File Tag and Target Sharing Privilege Set;
- *FileEncrypt(File)* - It encrypts the File with Convergent Encryption using 256-bit AES algorithm in cipher block chaining (CBC) mode, where the convergent key is from SHA-256 Hashing of the file; and
- *FileUploadReq(FileID, File, Token)* – It uploads the File Data to the Storage Server if the file is Unique and updates the File Token stored.

Our implementation of the **Private Server** includes corresponding request handlers for the token generation and maintains a key storage with Hash Map.

- *TokenGen(Tag, UserID)* - It loads the associated privilege keys of the user and generate the token with HMAC-SHA-1 algorithm; and
- *ShareTokenGen(Tag, {Priv.})* - It generates the share token with the corresponding privilege keys of the sharing privilege set with HMAC-SHA-1 algorithm.

Our implementation of the **Storage Server** provides de-duplication and data storage with following handlers and maintains a map between existing files and associated token with Hash Map.

- *DupCheck(Token)* - It searches the File to Token Map for Duplicate; and
- *FileStore(FileID, File, Token)* - It stores the File on Disk and updates the Mapping.

VII. EXPERIMENTAL RESULTS

The final results of the designed system are given below. From those results we get the detailed information to Check de-duplication and upload the files, Fetching the Signs using Hashing Algorithm, Checking for Duplication, file uploading, file downloading and attacker trying to attack(block) the cloud. Detailed procedure of the proposed system is given. Based on this we confirm that securely authorized de-duplication is successfully achieved with hybrid cloud approach. The output images given as below,

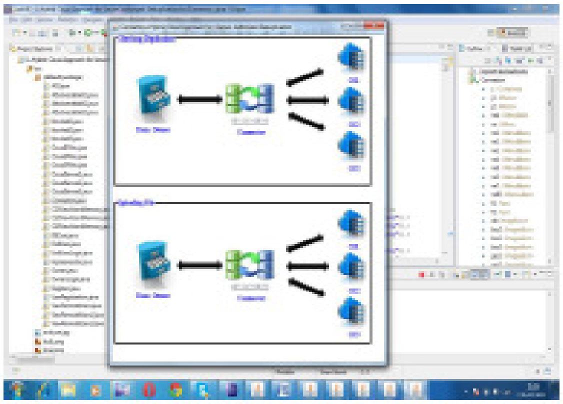


Figure 3. Checking duplication and uploading the files

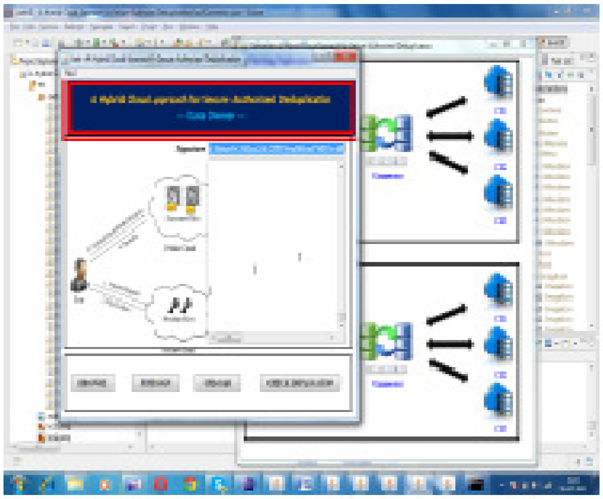


Figure 4. Fetching the Signs using Hashing Algorithm

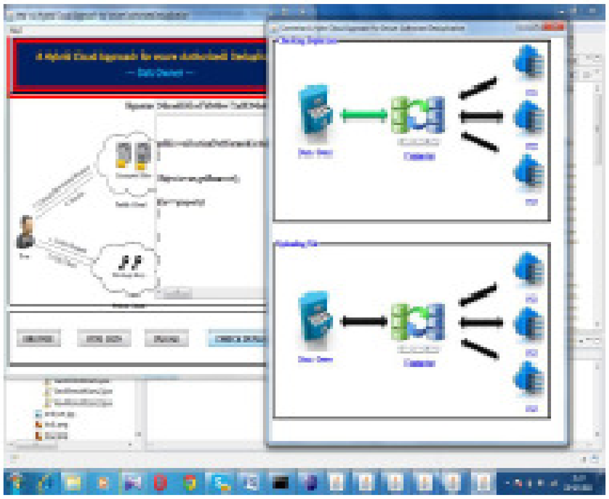


Figure5(a). Checking for Duplication.

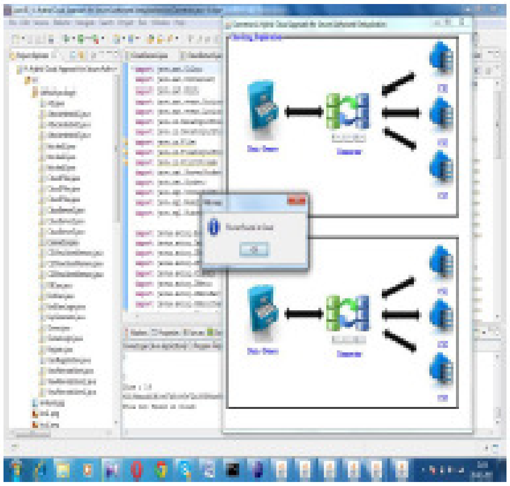


Figure 5(b). File not found in the cloud after checking for duplication

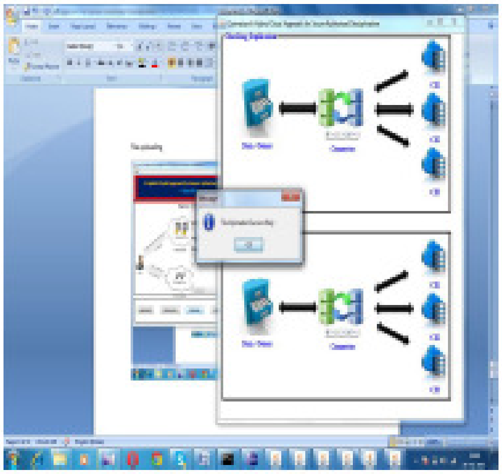


Figure7. Successfully uploaded the file

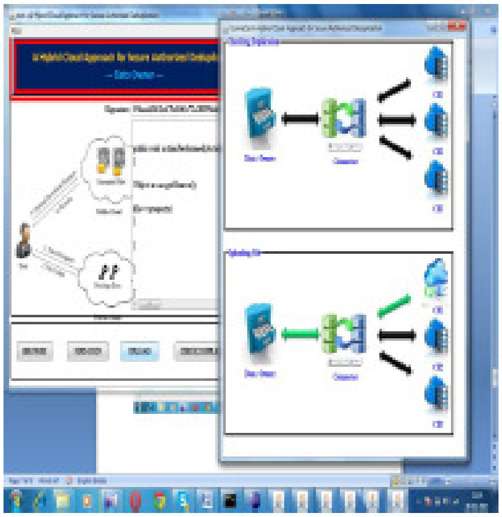


Figure6. File uploading

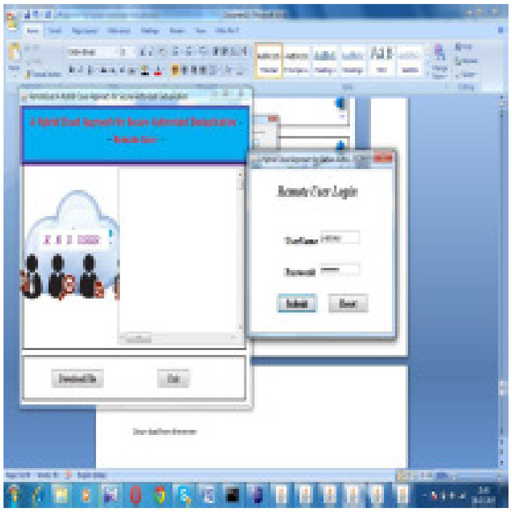


Figure8. Downloading files from the server

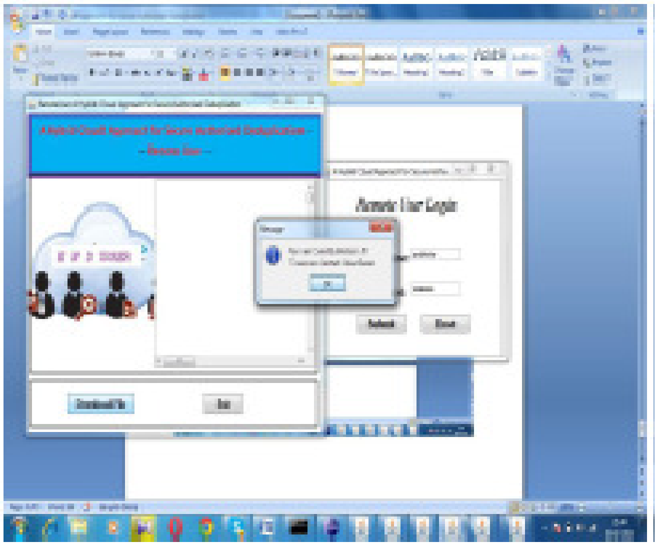


Figure9. Attacker trying to attack the cloud (Blocked)

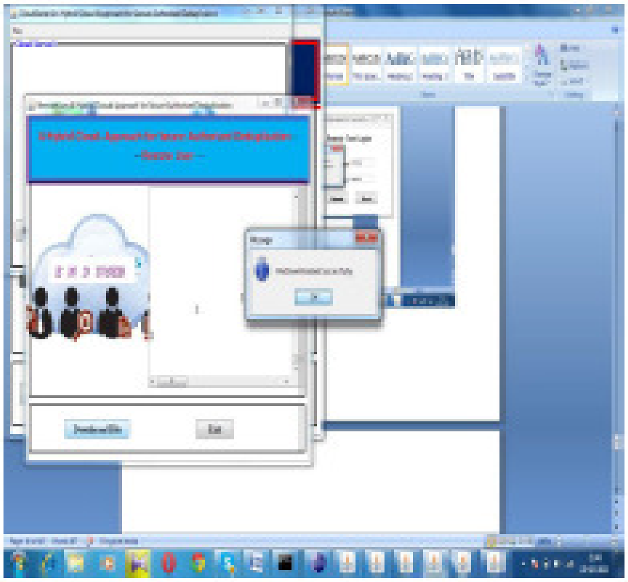


Figure10. Successfully downloading the file if a valid user is logging in

VIII. FUTURE SCOPE

It excludes the security problems that may arise in the practical deployment of the present model. Also, it increases the national security. It saves the memory by deduplicating the data and thus provide us with sufficient memory. It provides authorization to the private firms and protect the confidentiality of the important data.

REFERENCES

I. P. Anderson and L. Zhang. Fast and secure laptop backups with encrypted de-duplication. In Proc. of USENIX LISA, 2010.

II. M. Bellare, S. Keelveedhi, and T. Ristenpart.Message- locked encryption and secure deduplication. In EUROCRYPT, pages 296–312, 2013..

III. M. Bellare, S. Keelveedhi, and T. Ristenpart.Dupless: Serveraided encryption for deduplicated storage. In USENIX Security Symposium, 2013.

IV. S. Bugiel, S. Nurnberger, A. Sadeghi, and T. Schneider. Twin clouds: An architecture for secure cloud computing. In Workshop on Cryptography and Security in Clouds (WCSC 2011), 2011.

V. J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou. Secure deduplication with efficient and reliable convergent key management. In IEEE Transactions on Parallel and Distributed Systems, 2013.

VI. Bugiel, S., N¨urnberger, S., Sadeghi, A.-R., Schneider, T.: Twin Clouds: An architecture for secure cloud computing (Extended Abstract). In: Workshop on Cryptography and Security in Clouds (WCSC 2011), March 15-16 (2011)

VII. Chung, K.-M., Kalai, Y., Vadhan, S.: Improved delegation of computation using fully homomorphic encryption. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 483–501. Springer, Heidelberg (2010)

VIII. Cloud Security Alliance. Top threats to cloud computing, v. 1.0 (2010)

APPANA NAGA DURGARAOis pursuing mtech in Department of Computer Science Technology, Baba Institute of Technology And Sciences Bakkannapalem, Madhurawada, Visakhapatnam, Andhra Pradesh -530048.

DAKINENI DURGAPRASAD, CompltedB.Tech & M.Tech (M.Tech from Gitam University), present working as a Assistant Professor in Department of Computer Science Technology, Baba Institute Of Technology And Sciences Bakkannapalem, Madhurawada, Visakhapatnam, Andhra Pradesh and his experience is more than 10 yrs.