IT-DHSD: IMPLICIT TIME BASED DATA HANDLING AND SELF DESTRUCTION USING ACTIVE STORAGE OBJECT FOR CLOUD COMPUTING

Mr. Sandip N. Vende[1], Asst. Prof. Nitesh Rastogi[2]
Department of Computer Science & Engineering,
JDCT, Indore(M.P.), India

ABSTRACT

Computing is changing frequently to serve the user’s needs for security and reducing a dependency which leads researcher to develop newer technologies. Cloud computing is one of those technologies which successfully delivers the service oriented architecture to reduce the burdensome of managing the devices. But in some situation instead of providing controls to the end user it operates as a reverse effect. Data destruction is the one of that transformation in which the deletion of data is required by the user from the storage locations. But the existing destruction mechanism will leaves certain type of metadata residues from which either the data or users information can be regenerated which gives an attack prone zone for outsiders. This work proposes a novel IT-DHSD mechanism based on effective active storage object transition with privacy enabled operation to give more control to the user. In this each object contains the data and the destruction or deletion time which works as an implicit triggering condition for complete data removal. The approach also suggests synchronous modifications to different copies even with the delete operation also. At the prelim status of work the approach is completely satisfying the user’s needs of this time and will proves its efficiency in terms of performance, security and reduced overhead in near future of prototypic implementation.

Index Terms: Cloud Computing, Data Destruction, Active Storage Object (ASO), Active Object Table (AOT), Deletion Policies, Implicit Time Based Data Handling and Self Destruction (IT-DHSD);
I. INTRODUCTION

Cloud computing is the current area through which the computing ability and components can be delivered as a service to the end user. The maintenance and operational burdens are reduced from the service owner by using the different service level agreements. It is a combined concept of distributed, grid, utility and self corrective autonomic environment. Everything should be treated as a utility model like pay-per-use model by which the processing, storage, bandwidth etc can be provided to end user as a tenancy model. It is a kind of intellectual collaborated evolution for organizations by which the load of managing server based technologies and capital cost are reduced and focus on their core business operations can be increased. Such effective services are based on a layered architecture and various organizations with different service level agreements between the user, service provider and the cloud provider works effectively. This multi-tenancy characteristic of cloud gets a strong impact on its security due to its dynamic scalability, multi-providers SLA and resources, virtualized environment and huge information size [1].

Several organization works together to provide effective services to the end users and hence the risk of misalignment between them will also be higher. A slight change in behaviour of functionality of any layer between them will disrupt all the service availability and reliability. Even the data security at the third party storage server might also be affected of loosened. The aim these service layers are to provide the data security and guaranteed operations when required and after the completion the temporary or permanent files which are of no use must be removed. Data will sustain its locations and being used in operation up to a limited or defined period of time. When this period of lifecycle is over then, data should be removed with all its copies. Most of the organizations have the several policies for this data destructions based on the fixed time interval. But as of now the copies or replicas of data is getting multiplicatively increased so deleting all in a single go is very difficult. Also the deletion is not complete and some residues metadata remains at the location of the files from which the recreation of data can be performed.

Data destruction is the process of deleting the data and its overall components and copies when the lifecycle of its operations is finished [2]. The deletion should be in such a way that its reconstruction cannot be performed but many organizations are unable to achieve such behaviour and hence left a vacant space for attackers to regenerates the copies of original and forged some other services by the same. The removal of data is quite a complicated task before which the total number of copies which is generated has to be identified. Whenever a file is replicated some information needs to be attached there in its replica about its previous file location and total number of replications applied by which all the same existing copies is located and deleted. Most of the organizations are unable to perform such forensic deletion or destruction of data from the storage and always have vulnerability of data regeneration attacks. Thus this paper gives a brief study of such issues and provides a solution to overcome the existing data destruction issues.

II. BACKGROUND

As of now cloud computing is getting popularity most frequently among the various existing methodologies because of its scalable, reliable and maintainable nature. It reduces the loads on end users and providers and increases the computational and operational capabilities. The prime focus of the cloud provider is to make the data available to users always whenever required. Also when the usage of data is finished and the overall lifecycle period is over than it needs to be removed. Generally the lifecycle includes the generation, transfer, use, share, archive and destruct. All the above phases are time constraints based or use count based. The data which is most used and shared will stays for longer period and the data with less use will removed more frequently. But in current scenarios there is no such policy available for effective data destruction. It could be named in several
ways by different authors like destruction, deletion, removal, decommissioning, sanitizing, vanishing, disposal etc.

The destruction is based on time factor of usage analysis of future which clears the further scope of data usage in near future. During the fault tolerance mechanism provided by the cloud the multiple copies of data is replicated to different locations. Also when the user share download the files to local machine and then uploads it again and sends to some other users, new copy and location of the same file is generated which is not in concern of the cloud service provider. Hence at the time of deletion these files remain as replicas over the different locations which might not be deleted. Also some of the files residues in terms of their Meta data from the same locations is not been deleted or removed from which the forensic attackers might generates the forged copy. Such issues are not taken over in the current data destruction in lifecycle management or storage schemes.

Apart from the wide adoption of cloud computing services, there are some factors whose outlines have to be marked form detection of activities avoiding the juridical limit of data access. These areas include the data regeneration after deletion from any of the server or storage locations by forensic means. Thus various law enforcement acts and their applicability is clarified by the author of [3]. Thus it is clear that the how crucial is the complete and secure data destruction. Here the service provider complies some of the new policies and procedures for this timely disposal and ensures that by the use of any forensic means, the data is not recoverable. For performing the above operation several guidelines are available like ISO 270001 and NIST 800-88, to destroy data as part of the decommissioning process [4].

Several solutions for the above approach is taken over object oriented cloud and gives the improved results than the procedural based frameworks. The aim is to develop a system in which time constrained data deletion is applied in virtual machines instances [5]. The phenomenon is quite simple, instead of replicating the copy of overall file or data, the object is created for single purpose use and then destroyed. The object in use is known as active storage object. This active storage can make the device more intelligent because of supportive internal computation in storage devices [6]. By this the granularity and flexibility of storage services is increased and hence the issues is resolved about the destruction because as their usage is over the object is destroyed itself. Whenever any modification is performed automatically it is reflected to its copies. Some of the work had also implemented the desired for the P2P systems by using the distributed hash tables (DHT) [7]. In this the encryption and decryption is used even after the replica is removed and the metadata is encrypted for preventing the further regeneration of the data. This paper gives a brief study about the existing data deletion approaches presented in next section of literature survey and later on suggest a novel solution to overcome the upcoming issues in this domain.

III. LITERATURE SURVEY

Cloud computing environment supports portability of data and services with reliable behaviour. Serving this reliable behaviour of the user, aims to get secure application and data storage always while the usage and removing the less usable and other components. So due to privacy reasons the data and the user information needs to be removed completely from the storage provider locations after the terminations of SLA’s. Removing this data effectively and purely comes under the data destruction activity. During the last few years, various approaches provide the different solutions for the above mentioned issues of self and complete data destructions. Among them, few approaches show their strong presence is covered here as the surveyed literature and given as:

Taking the privacy a major concern before and after the service usage, the paper [8] proposes a scheme for Zero Data Remnance Proof (ZDRP). It is a combined evidence given by the cloud data storage provider as regards to zero data remnance proof the after the SLA period is over. The mechanism holds the various SLA’s and maintained them as a proof for destroying the data after end
of usages. In the absence of this SLA management, data don’t secure even after the deletion has taken place. For implementing the solution the paper also suggests an algorithm for complete deletion along with some SLA’s. The implementation of this can be achieved by suitable variation of the data updating mechanisms provided by open cloud service providers (CSP).

Some of the authors focus on the encryption mechanism for securing the users data and metadata. Likewise suggested in the paper [9], in which a formal cryptographic based model for secure deletion is given. According to it the deletion or removal can be monitored by several policies of data removal from storage systems whose security totally relies on some of the cryptographic functions and keys. The work regularly maintains some of the deletion class in which the members are regularly updating their entries and those who required complete removal can be erased automatically with all its related entries. A prototype implementation of the approach is proving its efficiency through Linux based file system.

Some of the authors also focused their intentions towards the deletion of less important data or used data from the P2P systems. In such systems the type of attacks occurred due to remaining residues of the deleted files is very high. Specifically the copies related to the data have to be taken over specifically because their locations are different from the actual copies. In the paper [10] a Vanish system is proposed for completely removing the data using a global scale cryptographic technique and distributed hash table (DHT). The approach had also implemented a prototype for the suggested mechanism in OpenDHT Vuze Bit Torrents application online. Practical evaluations of the approach can be applied by adding a plug-in for different browsers.

Carrying forward the above approach of Vanish and updated model Safe Vanish is proposed in [11]. This is an improved mechanism by which the data can be able to destruct itself after the end of use and increases the privacy parameter. The approach implements a threshold function k for generating the composite key. It sustains the self destructing nature by limiting the attacker's prone zone and sniffing the attacks in real systems. At the primary work stages and implementation prototypes is proving the efficiency of the suggested approach.

In the paper [12], there are three modifications suggested which includes cascading operation, tide operation and Existing Vanish mechanism. On the basis of above mechanism improvements in the existing destruction phenomenon is measured. According to cascade operation, multiple key storage system is taken as a combined system which increases the attack resistance. Similarly tide is a new key storage phenomenon through apache servers online. Various attacks and their preventions is simulated after applying the suggested approach and measured a performance improvement and applicability generalization by Vuze, OpenDHT and Vanish. The calculated result shows that these defences provide a countable improvement over the original Vuze DHT, which is impractical in most of the situations.

Thus the aim is to remove all the data and its copies completely from the server and storage locations. It makes the data privacy a stronger hand over other security parameters. Most of the existing mechanism is suggesting the approaches based on copies, but none of them focusing on complete deletion. Complete removal and self destruction is the primary aim of the approach SeDas in [13]. It is an active object based approach in which apart from creating the copies of the data some active objects is created which decreases the probability of leaving the data residues after deletions. The approach uses a time field which works as a triggered event after which the automatically destroying the data is initiated. Practical evaluations and implementation of the approach is proving its efficiency from existing approaches in more than 72 % in the case of uploading and downloading.

Carrying forward the approach of active storage, this paper gives a virtualization realization phenomenal of applications running at client ends and the data treated as an object by which the throughput and latency is increased [14]. Here the virtual machines are acting as an active object and generating keys for each of the active partitions. By using this mechanism the encrypted files are
uploaded and downloaded from the server using the agent structure. The evaluations and verifications apply in both the cases of uploading and downloading to check the authenticity of process, application and the user.

The article given by [15] presented a disk based erasing mechanism for P2P systems which can be further modified and can be used for cloud and storage technologies also. The mechanism is serving a simple understanding about the complete removal of the data from the servers or storage locations which practically containing some of the disks which needs to be erased. They are dependent on the policies, serving the user’s needs about the self data disposals after a fixed time period of data Lifecycle. User required the clean data removal from the existing medium through these policies. The article also presents issues of simple delete can’t be able to remove all the information from the storage medium and some residues remain. This from this residue data regeneration or attacks can be formed. Thus the security mechanism having complete destruction is always required. The article presents a few of the product specific information about the above issues, solutions and provides the feature oriented compared with existing products.

IV. PROBLEM STATEMENT

Cloud computing is the third party reliable system having guaranteed security services for data in case of any failure, losses or thefts. Out of which the first two is fault or failure oriented which is some uncertain conditions due o environment variables. But the theft is a planned action and comes under the category of attacks. In outsourced environment the data is stored at different storage locations of multiple data centres. While the data is in uses various security policies had let the usage of data in secure ways. But after the usage or life period of sustainability is over these data bytes are removed or deleted from their locations. Such deletion should be of time oriented or some regular interval basis. All the existing techniques destruct this data and their copies which makes the data disappears from the clients view and seems to have deleted all the related information. Also the schemes using the active object are unable to define the policy behaviour at the time of modification of file. The vanish based schemes are not been able to completely destroy the information and some of the metadata remains there until some new data is rewritten on the same locations. Also some history of data needs to be maintained for the condition if requires the older data has to be regenerated or called. So for this the suggested approaches are using the regular archival mechanism which is capable of serving the futuristic scopes of deleted or removed data call. Some of the issues which remain unaddressed after studying the related articles of existing approaches are:

Problem-I: After the data destruction the data is not removed completely and there exists some residues of metadata or user’s information from which the Sybil attack is planted or some portion of data is regenerated.

Problem-II: Use of active object reduces the chances of data residues but some of the timely destruction of those created object and the number of copies in local machines have also be in controlled and recorded distribution.

Problem-III: The extension of created active objects of destruction is fixed and is no extendable. In certain case the extension for deletion period is required and has to be provided.

Problem-IV: Key based active object generation sustains users nature in key generation so some other user is not been able to extract the same location with different behaviour information’s.
Problem-V: Key shares, length, generation and distribution mechanism needs to be secure from different users and providers itself.

So based on the above shortcoming this paper proposes a novel model for improving the self data destruction mechanism and making it as a complete and secure deletion of data and its replicas. For this decentralized approach had chosen which generates the random keys for different time based active object triggering and have a specific procedure of performing the operations on storage.

V. PROPOSED IT-DHSDAPPROACH

The temporary data or reference removal will not work in case of cloud computing because of its number of backups and archives replica copies. These copies will remain there on some unknown distributed locations. If the destruction needs to be applied completely then the approach should remove all types of copies form every locations. This is what the existing approaches is not been able to apply. This work proposes a novel IT-DHSD (Implicit Time Based Data Handling and Self Destruction) mechanism to overcome the above mentioned issues by the use of secure active storage objects with destruction time and the replica modification scheme. According to the suggested approach shown in figure 1, initially the users call for a service or its storage requirements which will creates an object of user request for storage along with the component (data).

![PROPOSED IT-DHSD SCHEME FOR SELF DATA DESTRUCTION IN CLOUD COMPUTING](image)

This object contains the data, object identifier and destruction time for object. The object creation works on the single data copy by using an object generation each time when a user demands a data. Later on the mechanism provides the updation in the object to the main data copy. After this object creation the entries with respect to each active instance generation is stored on to some centrally managing server. According to the users characteristics this object transition can be further provided security by using the encryption approaches for objects. Later on to that this encrypted secure object is stored into its locations. Similarly when user demanded its data for modification an runtime copy of this stored data in terms of object is generated which after passing the similar inserted key at the
time of encryption is passed to retrieve the data. The above mechanism can be clarified by its components working given as:

A. **User**: It is the type of user accessing the services from the application or cloud server. The number of services offered or selected by user depends upon the users proficiency level.

B. **Active Storage Object Generator**: This phase actively generates the objects of the users request by initiating the active storage object (ACO) at AOT (Active Object Table Server). This AOT server stores the information regarding to number of object generation, their recognition ID’s, Key for encryption, Random Encryption mechanism selected, and the destruction time).

C. **Secure Object Transition**: This is an application server which assures the object security by encrypting them from the user based key. Later on the similar component is responsible for retrieving or decrypting the data and sharing the secret key.

D. **Networked Storage**: It holds the actual data encapsulated in an object with the destruction time.

The server calls the deletion time or the stored object with destruction time triggers the safe removal of the data from the location. The full secure version of object is stored at the storage location with fixed destroying time so the copies can’t be created from this and if it occurs than it destroy the object copy also.

![FIGURE 2: ACTIVE OBJECT BASED DATA MODIFICATION CONSISTENCY SCHEME](image)

**Active Object Handling through Proposed Scheme**

Security can be provided by integrating the encryption mechanism with active object creation but issues related to the multiple copies and their modification remains the same. Thus to manage the multiple copies and there modification with same number of copies and reflection of consistent changes to all simultaneously can be archived in four basic steps of the scheme.

**Step 1: Local Data Copy**

A user demands the local copy of its stored data to be changed is supplied to local machine in the form of active object. For this the user sends the command request CheckIn.

**Step 2: Revert Local Copy**

In this step the user sends the modified copy of object to the server or storage location from which the object is generated. For this user executes the command CheckOut.
Step 3: Changes Update to Master Copy

After the updated local copy reach to the storage location the changes has to be reflected to the master copy for consistency of the later use of the file. For this the user commands the Commit operation.

Step 4: Updation to all Local Copies at Distributed Locations

After the above three step the unified modification is applied to the central copy which needs to be supplied to each local machine which contains the previous copy of data before the commit operation. Thus those copies also need to be updated. Thus to achieve this server fires the command Update All after which all the copies at distributed locations is changed with the modifications done the files o object.

The outline of the scheme is shown in figure 2. According to the scheme the consistency is maintained between the several copies of similar data. Thus the suggested mechanism is capable of providing the centrally controlled mechanism for the data modification. Apart from the above improvements the object storage tendency will also be make more private so as the pattern based detection can be removed.

In the above scheme the original data is not passed between different locations and users. Instead of that the copy of master data is passed in the form of active object. Now these objects can be removed after the fixed user decided destruction time automatically. This provides the effective self destruction for the generated object.

Cloud is totally an outsourced environment in which the controls are always in the providers end. So whenever the users are required to make such controlled operation of removal of data the complete deletion is not possible. Thus by implementing the above mechanism controls can be shifted to the user for each object and the unified policies can be applied for updations and destructions of the data. Practical implementation of solution can be achieved by applying the above mechanism for both HDD and SDD based storages because the networked storage can be of any type and it may occurs that a local copy of the data is generated at users end. Now when the server copy is removed completely than the local copy of data should also be removed. Even the approach will also reduce the total number of read and write operations.

Applications

Now days the destruction mechanism is applicable for various online data storing applications including the web based and mobile based version. Some of the application where the suggested scheme can be effectively used for improved security and control over data and its modifications are:

(i) Social networking
(ii) Messaging services
(iii) Mailing service
(iv) Online document sharing
(v) Record Based Systems
(vi) Enterprise Resource Planning
(vii) Business intelligence
(viii) Transportation systems
VI. EXPECTED OUTCOMES

Cloud computing is the most vibrant combination of technologies which serves the user and provider a best way to interact through some broker based structure. In such environment various transition and control shifting is measured which decreases the user control on data. In some situations user needs to remove the data from storage mechanism which is not completely performed by existing approaches. This work gives a novel mechanism which is capable of performing the act with respect to time. So there are some measured outcomes after the peer level of work given here as:

(i) **Time Based Data Control for Modification**: Active storage object (ASO) must destroy automatically apply changes consistently without any triggered explicit action.

(ii) **Complete Deletion without any Remaining Residue**: Expiration of ASO makes the data also unreachable because the key is removed from the application server so no residue of data remains from which regeneration cannot be planned.

(iii) **Secure Active Object Transition between the machines**: Encryption of active object let the privacy of transition increased.

(iv) **Unique User Based Key Generation and Secure Key Sharing**.

(v) **Less Vulnerable to Attacks Specially the Sybil and Intruder**.

(vi) **Known Timeout for Implicit Data Destruction**.

(vii) **Compatible with several storage technologies such as SSD and HDD**: System is compatible with the existing infrastructure so no need to make updations on such cost factors.

(viii) **Parallel modification to copies at distributed location**: Reduced burdensome increases the cost and effort benefits for cloud provider and service users.

(ix) **Complete synchronous operations**.

(x) **Object based structure reduces the overhead and increases performance**.

VII. CONCLUSION

Cloud computing raises the users trust on conditional storage at third party locations. This condition gives the user trust over the owned data means for any changes the modification will be uniform and will update to the all existing copy of the same data. Even with destruction all the copies should be removed completely. But the existing mechanism is unable to achieve this goal. After studying the various research articles, this paper presents a novel IT-DHSD approach for improved self data destruction mechanism satisfying the feature of complete deletion in bounded time factor. Here, suggested approach effectively uses the active storage object transition and controlled modification with consistency in nature. By this mechanism, changes applied will be reflected to each copy with synchronous operations even with the deletion or removal also. Proposed approach will serve to satisfy the user requirements for privacy and integrity based data access and provides the complete deletion of data. It works even with the distributed structures also in the same manner and will definitely proves its efficiency in near future prototypic implementations.
VIII. REFERENCES


