A Self-Destructing Data System Based on Active Storage Framework

Sonali P. Gathe[1] Prof. Manoj S. Chaudhari[2]
[1]M.Tech Student [2]Professor&H.O.D
[1,2]Department of Computer Science and Engineering
[1,2]Priyadarshini Bhagwati College of Engg., Nagpur, India

Abstract— Individual information put away in the Cloud may contain account numbers, passwords, notes, and other essential data that could be utilized and abused by a contender, or an official courtroom. This information are stored, replicated, and documented by Cloud Service Providers (CSPs), frequently without clients' approval and control. Self-destructing information primarily goes for securing the client information's protection. Every one of the information and their duplicates get to be destructed or mixed up after a client determined time, with no client intercession. What's more, the decryption key is destructed after the client indicated time. In this paper, we exhibit SeDas, a framework that meets this test through a novel coordination of cryptographic methods with dynamic stockpiling systems taking into account T10 OSD standard. We actualized a proof-of-concept SeDas model. Through usefulness and security properties assessments of the SeDas model, the outcomes show that SeDas is down to earth to utilize and meets all the protection saving objectives depicted. Contrasted with the framework without self-destructing information component, throughput for transferring and downloading with the proposed SeDas acceptably diminishes by fewer than 72%, while inertness for transfer/download operations with self-destructing information instrument increments by fewer than 60%

Key words: Cloud storage, data sharing and key aggregated encryption

I. Introduction

There is no concern about the fact that cloud computing has been growing day by day. Companies all over the world are tuning to different cloud platform for their I.T needs. It is the fact that, 80 % of assets of 1000 companies was used various aspect of cloud computing in the year 2013 & 20% do so without buying single piece of hardware.

What is cloud computing?

U.S national institute of standards and technology gives definition: The cloud enables convenient ,on demand network access to a shared pool of computing resources-networks, servers, storage applications, and services among others several key features are crucial to the cloud offering and to today's dynamic business environment. Cloud computing fails in some security measures the threat to the security of clouds is data theft attacks. Data sharing is one of the important aspects of cloud storage. In this paper we show how to share the data with the others in cloud storage in a secure and efficient manner.

We are proposing new public key cryptosystem that produces constant size cipher texts such that it authorize and delegate the decryption rights for any set of cipher text. Here we are using Microsoft azure as cloud storage and AES algorithm for the encryption. Data which is going to be share is divided into four parts using simple random sampling after that there is key generation using auto key generation. By using this key encryption is performed and data is going to be encrypted which is going to be store in different containers of cloud storage. Now if authorize user wants this data than data owner give Master secret key that possess the power of all keys to that legal user now this Master secret key will be enter by the legal user than automatically data is merge from different cloud storage containers and it is going to be downloaded by the legal user. In this way data is going to share in secure manner without losing confidentiality of data which is our main objective.

A. Features Of Cloud Computing:

Cloud computing is defined by five features

- On-demand self-service: Establish manage and terminate services on your own, without involving service provider.
- Broad network access: Use a standard browser to access the user interface, without any unusual software adds-on or specific operating system requirement.
- Resource pooling: Share resources and cost across large pool of users, allowing for centralized and increased peak load capacity.
- Rapid elasticity: Leverage capacity as needed, when needed and give it back when it is no longer required.
- Measured services: Consume resources as service and pay only for resources use.

B. Service Models Of Cloud Computing:

Various service models of cloud computing are:

1) Software as a Service (SaaS): In a Software as a Service model pre-made applications, along with any operating system, software, hardware and network.
2) Platform as a Service (PaaS): In a Platform as a Service model user can develop his own software and applications. Basically it provides platform to develop various software and applications.
3) Infrastructure as a Service (IaaS): In Infrastructure as a Service user can develop or install his own operating system, software and applications rather than purchasing servers, network devices, etc.

C. Deployment Models Of Cloud Computing:

The deployment models of cloud computing are categorizes according to their accessibility, structure of organization and provisioning location. They are:

1) Public cloud: Public cloud is depend on standard cloud computing model in which service provider makes all the resources available to general public over the internet.
2) Private cloud: Private cloud is acquire by an organization which is hosted externally or internally and manage internally or by trusted third party. Private cloud can only access by members, employs and trusted third party of organization.
3) Community cloud: In community cloud the infrastructure is shared among several organization from some specific community which is manage internally or

by trusted third party and also host internally or externally.

4) Hybrid cloud: Hybrid cloud is the combination of public, private or community clouds.

D. Data Self-Destruct:

The self-destructing data system in the Cloud environment should meet the following requirements: i) How to destruct all copies of the data simultaneously and make them unreadable in case the data is out of control? A local data destruction approach will not work in the Cloud storage because the number of backups or archives of the data that is stored in the Cloud is unknown, and some nodes preserving the backup data have been offline. The clear data should become permanently unread- able because of the loss of encryption key, even if an attacker can retroactively obtain a pristine copy of that data; ii) No explicit delete actions by the user, or any third-party storing that data; iii) No need to modify any of the stored or archived copies of that data; iv) No use of secure hardware but support to completely erase data in HDD and SSD, respectively.

Tang et al. [11] proposed FADE which is built upon standard cryptographic techniques and assuredly deletes files to make them unrecoverable to anyone upon revocations of file access policies. Wang et al. [12] utilized the public key based homomorphism authenticator with random mask technique to achieve a privacy-preserving public auditing system for Cloud data storage security and uses the technique of a bilinear aggregate signature to support handling of multiple auditing tasks. Perlman et al. [13] present three types of assured delete: expiration time known at file creation, on-demand deletion of individual files, and custom keys for classes of data.

Vanish [1] is a system for creating messages that automatically self-destruct after a period of time. It integrates crypto- graphic techniques with global-scale, P2P, distributed hash tables (DHTs): DHTs discard data older than a certain age. The key is permanently lost, and the encrypted data is permanently unreadable after data expiration. Vanish works by encrypting each message with a random key and storing shares of the key in a large, public DHT. However, Sybil attacks [3] may compromise the system by continuously crawling the DHT and saving each stored value before it ages out and the total cost is two or- ders of magnitude less than that mentioned in reference [14] es- timated. They can efficiently recover keys for more than 99% of Vanish messages. Wolchok et al. [3] concludes that public DHTs like VuzeDHT [15] probably cannot provide strong enough security for Vanish. So, Geambasu et al. [14] proposes two main countermeasures.

Although using both OpenDHT [16] and VuzeDHT might raise the bar for an attacker, at best it can provide the maximum security derived from either system: if both DHTs are insecure, then the hybrid will also be insecure. OpenDHT is controlled by a single maintainer, who essentially functions as a trusted third party in this arrangement. It is also susceptible to attacks on roughly 200 PlanetLab [17] nodes on which it runs, most of which are housed low-security research facilities. Vanish is an interesting approach to an important privacy problem, but, in its current form, it is insecure [3].

To address the problem of Vanish discussed above, in our previous work [4], we proposed a new scheme, called Safe Vanish, to prevent hopping attack, which is one kind of the Sybil attacks [18], [19], by extending the length range of the key shares to in- crease the attack cost substantially, and did some improvement on the Shamir Secret Sharing algorithm [20] implemented in the Vanish system. Also, we presented an improved approach against sniffing attacks by way of using the public key cryptosystem to prevent from sniffing operations.

However, the use of P2P features still is the fatal weakness both for Vanish and Safe Vanish, because there is a specific at- tack against P2P methods (e.g., hopping attacks and Sybil at- tacks [3]).

In addition, for the Vanish system, the survival time of key attainment is determined by DHT system and not controllable for the user. Based on active storage framework, this paper pro- poses a distributed object-based storage system with self-destructing data function. Our system combines a proactive approach in the object storage techniques and method object, using data processing capabilities of OSD to achieve data self-destruction. User can specify the key survival time of distribution key and use the settings of expanded interface to export the life cycle of a key, allowing the user to control the subjective life-cycle of private data.

E. Object-Based Storage and Active Storage

Object-based storage (OBS) [21] uses an object-based storage device (OSD) [22] as the underlying storage device. The T10 OSD standard [22] is being developed by the Storage Networking Industry Association (SNIA) and the INCITS T10 Technical Committee. Each OSD consists of a CPU, network interface, ROM, RAM, and storage device (disk or RAID subsystem) and exports a high-level data object abstraction on the top of device block read/write interface.

With the emergence of object-based interface, storage devices can take advantage of the expressive interface to achieve some cooperation between application servers and storage devices. A storage object can be a file consisting of a set of ordered logical data blocks, or a database containing many files, or just a single application record such as a database record of one trans- action. Information about data is also stored as objects, which can include the requirements of Quality of Service (QoS) [23], security [24], caching, and backup. Kang et al. [25] even implemented the object-based model enables storage class memories (SCM) devices to overcome the disadvantages of the current interfaces and provided new features such as object-level reliability and compression. In recent years, many systems, such as Lustre [26], Panasas [27] and Ceph [28], using object-based technology have been developed and deployed. Since the data can be processed in storage devices, people attempt to add more functions into a storage device (e.g., OSD) and make it more intelligent and refer to it as “Intelligent Storage” or “Active Storage” [5]–[10]. For instance, IDISK [29] and SmAS Disk [30] can offload application codes to disks, but the disks respond to I/O requests of clients passively. A stream-based programming model has been proposed for Active Disk [31]–[33], but the stream is allowed to pass through only one disklet (user-specific code).

Today, the active storage system has become one of the most important research branches in the domain of intelligent storage systems. For instance, Wickremesinghe et al. [34] proposed a model of load-managed active storage, which strives to integrate computation with storage access in a way that the system can predict the effects of offloading computation to Active Storage Units (ASU). Hence, applications can be configured to match hardware capabilities and load conditions. MVSS [35], a storage system for active storage devices, provided a single framework to support various services at the device level. MVSS separated the deployment of services from file systems and thus allowed services to be migrated to storage devices.

There have been several efforts to integrate active storage technology into the T10 OSD standard. References [5], [7], [8], and [10] all proposed their own implementation of active storage framework for the T10 OSD standard. These implementations either are preliminary or validate their systems on a variety of data intensive applications and fully demonstrate the advantage of object-based technology. Our work extends prior research (such as Qin et al.'s [5], John et al.'s [7], Devulapalli et al.'s [8] and Xie et al.'s [10]) in this area by considering data self-destruction.

II. MICROSOFT AZURE

Microsoft Azure provides a Microsoft Windows Server based computing environment for applications and persistent storage for both structured and unstructured data, as well as asynchronous messaging. Azure also provides a range of services that helps us connect users and on-premises applications to cloud-hosted applications, manage authentication, use inter-service messaging, and implement data management and related features such as caching.

Azure also includes a range of management services that allows us to control all these resources, either through a web-based user interface (a web portal) or programmatically. In most cases there is a REST-based API that can be used to define how our services will work. Most management tasks that can be performed through the web portal can also be performed using the API.

A. Microsoft Azure's Compute Environment

The Azure compute environment consists of a platform for applications and services hosted within one or more roles. The types of roles we can implement in Azure are:

Azure Compute (Web and Worker Roles). An Azure application consists of one or more hosted roles running within the Azure data centers. Typically there will be at least one web role that is exposed for access by users of the application. A web role is supported by Internet Information Service (IIS) 7.0 and ASP.NET. The application may contain additional roles, including worker roles, that are typically used to perform background processing and support tasks for web roles.

Virtual Machine (VM role). This role allows us to host our own custom instance of the Windows Server 2008 R2 Enterprise or Windows Server 2008 R2 Standard operating system within an Azure data center.

B. Data Management in Microsoft Azure:

Azure, SQL Azure, and the associated services provide opportunities for storing and managing data in a range of ways. The following data management services and features are available:

– Azure Storage. This provides four core services for persistent and durable data storage in the cloud. The four storage services are listed below:

1) The Azure Table Service:-□The Azure Table Service provides a table-structured storage mechanism and supports queries for managing the data. The Azure Table Service is a No SQL offering that provides schema-less storage. It is primarily aimed at scenarios where large volumes of data must be stored, while being easy to access and update.
2) The Binary Large Object (BLOB) Service:-□The Binary Large Object (BLOB) Service provides a series of containers aimed at storing text or binary data. It provides both Block BLOB containers for streaming data, and Page BLOB containers for random read/write operations.
3) The Queue Service:- The Queue Service provides a mechanism for reliable, persistent messaging between role instances, such as between a web role and a worker role.
4) Azure Drives:-Azure Drives provide a mechanism for applications to mount a single volume NTFS VHD as a Page BLOB, and upload and download VHDs via the BLOB.

– SQL Azure Database. This is a highly available and scalable cloud database service built on SQL Server technologies that supports the familiar T-SQL-based relational database model. It can be used with applications hosted in Azure, and with other applications running on-premises or hosted elsewhere.

– Data Synchronization. SQL Azure Data Sync is a cloud-based data synchronization service built on Microsoft Sync Framework technologies. It provides bi-directional data synchronization and data management capabilities, allowing data to be easily shared between multiple SQL Azure databases and between on-premises and SQL Azure databases.

– Caching. This service provides a distributed, in-memory, low latency and high throughput application cache service that requires no installation or management, and dynamically increases and decreases the cache size as required. It can be used to cache application data, ASP.NET session state information, and for ASP.NET, page output caching.

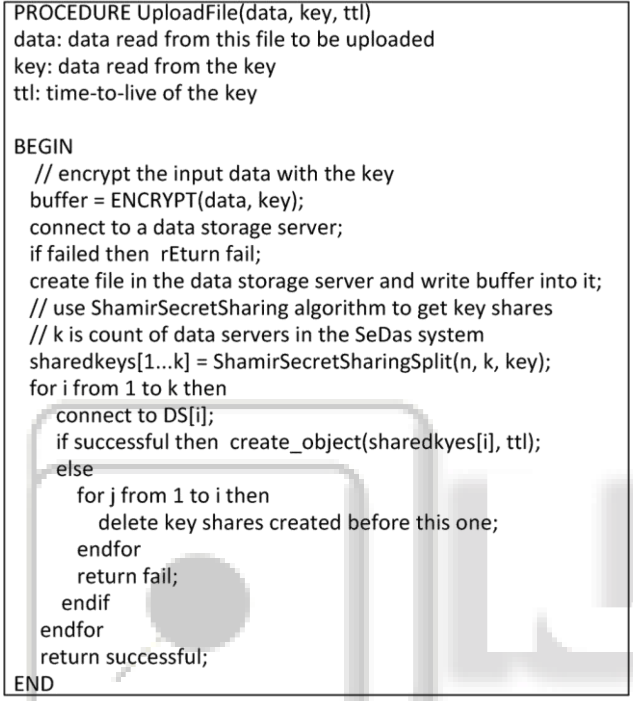
III. KEY AGGREGATE ENCRYPTION (KAC)

Initially we describe the main components of KAC than we describe the application of KAC in cloud storage.

– Components of KAC:

KAC scheme will have five algorithms that run in polynomial time. Data owner will maintain the public system parameter through Setup which generates pair of public key/Master secret key through KeyGen. Encryption of messages will be done through Encrypt by anyone or data owner and also decides which cipher text class is associated with the plaintext message which is to be encrypted. Owner of the data will use master secret key to generate an aggregate decryption key for a particular set of cipher text classes through Extract. Now the keys are passed to delegates in secure manner at last any user with an aggregate key can decrypt any cipher text through Decrypt.

1) Setup: - It is implemented by the owner of data to setup an account on non-trusted server.
2) KeyGen: - It is implemented by the owner of the data to generate Public/Master secret key pair.
3) Encrypt: - It is implemented by anyone or data owner who wants to encrypt data.
4) Extract: - It is implemented by the data owner for giving decrypting power for particular set of cipher text classes to whom that want to access this data.
5) Decrypt: - It is implemented by the user that received aggregate key through Extract.



```
PROCEDURE UploadFile(data, key, ttl)
data: data read from this file to be uploaded
key: data read from the key
ttl: time-to-live of the key

BEGIN
  // encrypt the input data with the key
  buffer = ENCRYPT(data, key);
  connect to a data storage server;
  if failed then  rEturn fail;
  create file in the data storage server and write buffer into it;
  // use ShamirSecretSharing algorithm to get key shares
  // k is count of data servers in the SeDas system
  sharedkeys[1...k] = ShamirSecretSharingSplit(n, k, key);
  for i from 1 to k then
     connect to DS[i];
     if successful then  create_object(sharedkyes[i], ttl);
     else
        for j from 1 to i then
           delete key shares created before this one;
        endfor
        return fail;
     endif
   endfor
   return successful;
END
```

Fig. 1:

– Application of KAC in Cloud Storage:

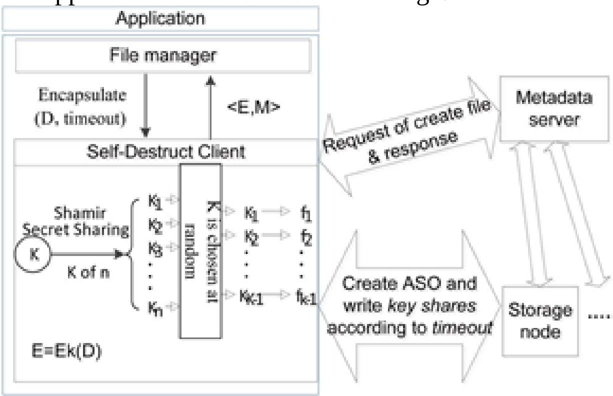


Fig. 2:

– IMPORTANT STEPS: - Following are the important step which perform during accessing of cloud server

1) Authentication And Authorization
2) File Encryption by KAC
3) Cloud data sharing
4) File Decryption by KAC
5) Regeneration of Data

1) Authentication and Authorization:- Authentication and Authorization process are the required of the Verifying the User Originality and Appropriate Session Activities of the Registered User.
2) File Encryption by KAC: - After user login, the user can able to store the files into the cloud in an encryption manner. So that the encrypted files which are stored in the cloud cannot be decrypted normally by other users or hacker's.
3) Cloud data sharing: - The user initially uploads's files data to the cloud, and shares it with other users. The shared files are encrypted by the owner. Whenever the other user's want to access or decrypt the file required key permission to be provided by data Owner.
4) File Decryption by KAC: - The aggregate keys are sent in mail to other user by the original user. The file will be decrypted by the aggregate key's which was generated by the KAC. Finally, any user with an aggregate key can decrypt any cipher text.
5) Regeneration of Data: - In Case of Corruption of any Share of data or loss of any share on cloud, that particular share will be regenerated from a backup storage.

IV. Conclusion

Data privacy is one of the most important aspects of the cloud storage. There are several mathematical tools and cryptographic technique that involve several keys for single. In this paper our main focuses is on how to compress secret key in a public key cryptographic environment that support delegation of secret keys for the different classes of cipher text in cloud storage also the person that want to access to the particular data will get constant size aggregate key. Our method is more convenient than the hierarchical key assignment scheme that can only reduce spaces if all the holders of key share a similar set access. Limitation of this method is the predefined bound of the number of cipher text classes as in cloud storage the number of cipher text are increases rapidly.

References

[1] Cheng-Kang Chu, Sherman S. M. Chow, Wen-Guey Tzeng, Jianying Zhou, and Robert H. Deng Senior Member, IEEE key –Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage IEEE Transactions on Parallel and Distributed Systems. , 25(2) , 468.2014

[2] C Wang, S.S.M.Chow, Q.Wang, K.Ren, and W.Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Trans.Computers, vol. 62, no. 2, pp. 362–375, 2013

[3] Lingfang Zeng, Shibin Chen, Qingsong Wei, and Dan Feng, "A Self-Destructing system Based on Active Storage Framework," IEEE Trans. Magnetics., vol. 49, no. 6, July 2013.

[4] S. S. M. Chow, C.-K. Chu, X. Huang, J. Zhou, and R. H. Deng,"Dynamic Secure Cloud Storage with Provenance," in Cryptography and Security: From Theory to Applications – Essays Dedicated to Jean-Jacques Quisquater on the Occasion of His 65th Birthday, ser. LNCS, vol. 6805. Springer, 2012, pp. 442–464

[5] Y. Xie, K.-K. Muniswamy-Reddy, D. Feng, D. D. E. Long, Y. Kang, Z. Niu, and Z. Tan, "Design and evaluation of oasis: An active storage framework based

on T10OSD standard," Massive Storage Systems and Technologies (MSST), 27th IEEE Symp. , 2011.

[6] M. Wei , L. M. Grupp , F. E. Spada and S. Swanson "Reliably erasing data from flash-based solid state drives", Proc. 9th USENIX Conf. File and Storage Technologies (FAST) , 2011

[7] S. W. Son, S. Lang, P. Carns, R. Ross, R. Thakur, B. Ozisikyilmaz, W. K. Liao, and A. Choudhary, "Enabling active storage on parallel I/O software stacks," Proc. IEEE 26th Symp. Mass Storage Systems and Technologies (MSST), 2010.

[8] M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken, "Dynamic and Efficient Key Management for Access Hierarchies," ACM Transactions on Information and System Security (TISSEC), vol.12,no. 3, 2009.

[9] V.Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data, "in Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06). ACM, 2006, pp. 89–98.

[10] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," in Proceedings of Advances in Cryptology - EUROCRYPT 03, ser. LNCS,vol. 2656. Springer, 2003, pp. 416–432.