Self Annihilating Data Storage and Location Based Adaptive Video Streaming service for CLOUD assisted mobile users – “SADS & LBAVS”

Abstract— Cloud Computing has become a popular buzzword and it has been widely used to refer to different technologies, services, and concepts. With the use of cloud computing, here we are trying to give the location based efficient video information to the mobile users. Location Based Service(LBS) is an information service and has a number of uses in social networking today as an entertainment service, which is accessible with mobile devices through the mobile network and which uses information on the geographical position of the mobile device. While the demands of video streaming services over the mobile networks have been souring over these years, the wireless link capacity cannot practically keep up with the growing traffic load. In this project, we propose and discuss a Adaptive video streaming framework to improve the quality of video services in the location based manner. Through this system, video content can be segmented by an automatic shot/scene retrieval technology and stored in the database (DB). In the client side, two threads will be formed. One is for Video streaming and another one is for Location searching and updating. For the security purpose, we are using self destruction algorithm where the uploaded video is been destructed automatically after the user defined time. Thus the location based Video information can be streamed efficiently and securely by the mobile users.

Keywords: Cloud Computing, Video Streaming, Self Annihilation Data, Mobile Computing, SVC, Data Privacy, Location Based Service(LBS).

I. INTRODUCTION

Cloud computing, or the cloud, is a colloquial expression used to describe a variety of different types of computing concepts that involve a large number of computers connected through a real-time communication network such as the Internet. Cloud computing is a term without a commonly accepted unequivocal scientific or technical definition. In science, cloud computing is a synonym for distributed computing over a network and means the ability to run a program on many connected computers at the same time.

While the demands of video streaming services over the mobile networks have been souring over these years, the wireless link capacity cannot practically keep up with the growing traffic load. The gap between the traffic demand and the link capacity, along with time-varying link conditions, results in poor service quality of video streaming services over the mobile networks, such as intermittent disruptions and long buffering delays. Leveraging the current cloud computing technology, we propose and discuss a framework to improve
the quality of video services for mobile users, i.e., Segment video streaming. Through this system, video content can be segmented by an automatic shot/scene retrieval technology and buffered independently.

Location-based services (LBS) are a general class of computer program-level services used to include specific controls for location and time data as control features in computer programs. As such LBS is an information service and has a number of uses in social networking today as an entertainment service, which is accessible with mobile devices through the mobile network and which uses information on the geographical position of the mobile device. This has become more and more important with the expansion of the smartphone and tablet markets as well.

In the client side two threads are formed, one is for searching and updating the location and another one is for buffering and streaming the video. Thus the videos will be streamed according to the mobile user’s location. For example, if the user enters into the electrical department, electrical related videos will be streamed. Then if he enters into the mechanical department, mechanical related videos will be streamed, vice versa. These streaming will be updated automatically according to the location. Security is one of the important challenges in Cloud computing environment. Personal data and videos, and other important information that could be used and misused by a miscreant, a competitor, or a court of law. These data are cached, copied, and archived by Cloud Service Providers (CSPs), often without users’ authorization and control. Self-Annihilation Data (SAD) mainly aims at protecting the user data’s privacy. All the data and their copies become destructed or unreadable after a user-specified time, without any user intervention. In addition, the decryption key is destructed after the user-specified time. We present SAD, a system that meets this challenge through a novel integration of cryptographic techniques with active storage techniques based on T10 OSD standard. We implemented a proof-of-concept SAD prototype. Through functionality and security properties evaluations of the SAD prototype, the results demonstrate that SAD is practical to use and meets all the privacy-preserving goals described.

II. RELATED WORK

A. Adaptive Video Streaming

In the adaptive streaming, the video traffic rate is adjusted on the fly so that a user can experience the maximum possible video quality based on his or her link’s time-varying bandwidth capacity [1]. There are mainly two types of adaptive streaming techniques, depending on whether the adaptivity is controlled by the client or the server. The Microsoft’s Smooth Streaming [2] is a live adaptive streaming service which can switch among different bit rate segments encoded with configurable bitrates and video resolutions at servers, while clients dynamically request videos based on local monitoring of link quality. Adobe and Apple also developed client-side HTTP adaptive live streaming solutions operating in the similar manner. There are also some similar adaptive streaming services where servers controls the adaptive transmission of video segments, for example, the Quality Adaptive Streaming. However, most of these solutions maintain multiple copies of the video content with different bit rates, which brings huge burden of storage on the server. Regarding rate adaptation controlling techniques, TCP-friendly rate control methods for streaming services over mobile networks are proposed [3], [4], where TCP throughput of a flow is predicted as a function of packet loss rate, round trip time, and packet size. Considering the estimated throughput, the bit rate of the streaming traffic can be adjusted. A rate adaptation algorithm for conversational 3G video streaming is introduced by [5]. Then, a few cross-layer adaptation techniques are discussed [6], which can acquire more accurate information of link quality.

Recently the H.264 Scalable Video Coding (SVC) technique has gained a momentum [15]. An adaptive video streaming system based on studies the real-time SVC decoding and encoding at PC servers. The work in [17] proposes a quality-oriented scalable video delivery using SVC, but it is only tested in a simulated LTE Network.

Regarding the encoding performance of SVC, Cloud Stream mainly proposes to deliver high-quality streaming videos through a cloud-based SVC proxy, which discovered that the cloud computing can significantly improve the performance of SVC coding. The above studies motivate us to use SVC for video streaming on top of cloud computing.

B. Mobile Cloud Computing Techniques

The cloud computing has been well positioned to provide video streaming services, especially in the wired Internet because of its scalability and capability [18]. For example, the quality-assured bandwidth auto-
scaling for VoD streaming based on the cloud computing is proposed [9], and the CALMS framework [19] is a cloud-assisted live media streaming service for globally distributed users. However, extending the cloud computing-based services to mobile environments requires more factors to consider: wireless link dynamics, user mobility, the limited capability of mobile devices [20], [21]. More recently, new designs for users on top of mobile cloud computing environments are proposed, which virtualize private agents that are in charge of SVC is deployed in [16], which satisfying the requirements (e.g., QoS) of individual users such as Cloudlets [7] and Stratus [8]. Thus, we are motivated to design the AMES-Cloud framework by using virtual agents in the cloud to provide adaptive video streaming services.

C. 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 unreadable 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. [10] 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. [11] 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. [12] 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 is a system for creating messages that automatically self-destruct after a period of time. It integrates cryptographic 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 [13] may compromise the system by continuously crawling the DHT and saving each stored value before it ages out and the total cost is two orders of magnitude less than that mentioned in reference [14] estimated. They can efficiently recover keys for more than 99% of Vanish messages.

III. DESIGN AND IMPLEMENTATION

![Figure 1 - System Architecture](image-url)
A. User Registration:
The user must be registered with cloud. The registered user’s information will be stored in the cloud. Unique User ID will be generated for every registered user. Then the registered user can do file uploading and securely store their files in the cloud and they can efficiently download the required files.

B. Administrator
In this module the administrator has a major role. Here the administrator Uploading a different video files with categorized department. The user can view all uploaded videos based the department wise. Admin set video streaming locations at the time of video uploading. The video has been played depend upon this settings.

C. Cryptographic Video Upload in Cloud
The art of protecting information by transforming it (encrypting it) into an unreadable format, called cipher text. Only those who possess a secret key can decipher (or decrypt) the message into plain text. Encrypted messages can sometimes be broken by cryptanalysis, also called code breaking, although modern cryptography techniques are virtually unbreakable. As the Internet and other forms of electronic communication become more prevalent, electronic security is becoming increasingly important. Cryptography is used to protect e-mail messages, credit card information, and corporate data. So here we are using this cryptographic for video files for security purpose.

D. Self Annihilating Data Storage System:
An active storage object derives from a user object and has a time-to-live (ttl) value property. The ttl value is used to trigger the self-destruct operation. The ttl value of a user object is infinite so that a user object will not be deleted until a user deletes it manually. The ttl value of an active storage object is limited so an active object will be deleted when the value of the associated policy object is true. Interfaces extended by Active Storage Object class are used to manage ttl value. The create member function needs another argument for ttl. If the argument is 1, User Object::create will be called to create a user object, else, Active Storage Object::create will call User Object::create first and associate it with the self-destruct method object and a self-destruct policy object with the ttl value. The getTTL member function is based on the read_attr function and returns the ttl value of the active storage object. The setTTL, addTime and decTime member function is based on the write_attr function and can be used to modify the ttl value. To use the SADS system, user’s applications should implement logic of data process and act as a client node. There are two different logics: uploading and downloading.

We are using two kinds of algorithm for done this project in secure way. First algorithm called Random Key generation, the main class for done this algorithm called Random class. This algorithm mainly used for generate the secret key for file uploading and downloading. Second algorithm is AES (Advanced Encryption Slandered), which is used for encrypt the file content. Our contributions are we focus on the related key distribution algorithm, Shamir’s algorithm, which is used as the core algorithm to implement client (users) distributing keys in the object storage system. We use these methods to implement a safety destruct with equal divided key. Based on active storage framework, we use an object-based storage interface to store and manage the equally divided key. We are implementing a proof-of-concept SADS prototype. Through functionality and security properties evaluation of the SADS prototype, the results demonstrate that SADS is practical to use and meets all the privacy-preserving goals. The prototype system imposes reasonably low runtime overhead. SADS supports security erasing files and random encryption keys stored in a hard disk drive (HDD) or solid state drive (SSD), respectively.

E. Location Based Video Streaming
Each location has several Cloud Units(CU) which acts as mobile support station to support services for mobile users in this location. Cloud units in every location are connected to Cloud Service Provider. While streaming a video over the mobile, the video is being segmented by the broker layer, video content can be segmented by an automatic shot/scene retrieval technology and stored. Service Layer will be in the cloud and the broker layer will be in between application layer and the service layer. As soon as the Video is been demanded, video is transferred into the broker layer and the video is segmented as per the designed algorithm.
IV. SUMMARY

In this paper, the data (Video) can be stored in the cloud by the video service provider (vsp). That video will be getting deleted after the predefined time by the vsp. SAD architecture is used to do the self annihilation job. The video is being serviced in the location based manner. According to the location, related video is streamed. To avoid the buffering delay and to give non terminating video service, video is being segmented in the location based server. Thus the location based video streaming is established with the use of active storage system in the cloud.

V. CONCLUSION

Providing dynamic location-based service and increase the information retrieve accuracy especially in the limited mobile screen have become the important research areas in the development of location-based services. In this paper, we have proposed an Android based application to retrieve video information in mobile device accessing CSP based on the locations. While streaming the video the total video is segmented according to the bandwidth allocated and size of the video. So that the buffering delay can be reduced as the user moves one place to another. To give a service through the cloud computing technology, security is challenging one until now. To enable the secure service we proposed SADS system where the data is been deleted automatically from the cloud storage after the user specified time. Our plan to release the current SADS system will help to provide researchers with further valuable experience to inform future object-based storage system designs for Cloud services.

REFERENCES