ISSN:0975-9646

A Secured Adaptive Mobile Video Streaming and Efficient Social Video Sharing in the Clouds

V.Manasa , M.Vikram

Dept of CSE, Sri Venkateswara College of Engineering&JNTU- Anantapur, INDIA

ABSTRACT ---- The video traffic over mobile networks have been increasing enormously but the wireless link capacity cannot keep up with the traffic. This gap results in poor service quality of video streaming over mobile networks. Inculcating cloud computing technology into mobile networks, a new framework is introduced called Secured AMES-Cloud containing two parts: AMoV (Adaptive Mobile Video streaming) and ESoV (Efficient Social Video sharing) .For each user,AMov constructs a private agent to adjust streaming flow based on link quality using scalable video coding technique. ESoV allows social network interactions among users and private agents' prefetch user requested videos in advance. Here, security is provided to each user so that their videos cannot be seen by others unless the user requires and cannot be seen by cloud providers using Homomorphic and Incremental encryption.

Keywords: Scalable Video Coding, Cloud Computing, Mobile networks, Adaptive Video Streaming, Social Video Sharing, Homomorphic Encryption, Incremental Encryption

I. INTRODUCTION

Over the past decade, more traffic is increased due to different forms of video (TV, Internet, File sharing using P2P, Video on Demand –VOD etc.,) streaming and downloading by the mobile users in particular. Video streaming is not an issue in wired networks but wireless networks (mobile users) has been suffering from sharing of videos over limited bandwidth of links. Though 3G and LTE have been introduced to cope up with the bandwidth, the efforts were not successful due to rapid increase of mobile users.

While receiving videos via 3G/4G mobile networks, users suffer from long buffering time to load video and interruptions due to limited bandwidth and link fluctuations. Thus, it is important to increase the quality of video streaming in mobiles using networking and computing resources effectively.

The quality of mobile video streaming can be improved using two aspects:

- **Scalability**: Video Streaming service must be compatible with multiple mobile devices having various video resolutions, computing powers, wireless links and so on. Capturing multiple bit rates of same video may increase the burden on servers in terms of storage and sharing. To resolve this issue, the Scalable Video Coding (SVC) technique has been introduced. **Scalable Video Coding** (**SVC**) is the name for the Annex G extension of the H.264/MPEG-4 AVC video compression standard. SVC standardizes the encoding of a high-quality video bit stream that also contains one or more subset bitstreams. A subset video bitstream is derived by dropping packets from the larger video to reduce the bandwidth required for the subset bitstream. A subset bitstream can represent a lower spatial resolution, or a lower temporal resolution, or a lower quality video signal (each separately or in combination) compared to the bitstream it is derived from.

The following modalities are possible:

- ➢ Temporal (frame rate) scalability: The motion compensation dependencies are structured so that complete pictures (i.e. their associated packets) can be dropped from the bit stream.
- ➢ Spatial (picture size) scalability: Video is coded at multiple spatial resolutions. The data and decoded samples of lower resolutions can be used to predict data or samples of higher resolutions in order to reduce the bit rate to code the higher resolutions.
- ➢ SNR/Quality/Fidelity scalability: Video is coded at a single spatial resolution but at different qualities. The data and decoded samples of lower qualities can be used to predict data or samples of higher qualities in order to reduce the bit rate to code the higher qualities.
- ➢ Combined scalability: A combination of the 3 scalability modalities described above.

- **Adaptability:** Traditional video streaming techniques designed by considering relatively stable traffic links between servers and users perform poorly in mobile environments .Thus the fluctuating wireless link status should be properly dealt with to provide 'tolerable" video streaming services. To address this issue, we have to adjust the video bit rate adapting to the currently time-varying available link bandwidth of each mobile user. Such adaptive streaming techniques can effectively reduce packet losses and bandwidth waste.

Cloud computing techniques are used to provide scalable resources to service providers to serve mobile users. Hence, clouds are used for large scale real time video services. Many Mobile cloud computing technologies have provided private agents for serving mobile users e.g., Cloudlet. This is because, in cloud multiple threads can be created dynamically based on user demands.

Social Network Services (SNS's) have occupied a major role recently. In SNS's user can share, comment, post the videos among friends and groups. Users can follow their favourites depending on their interest in which their followers are likely to watch popular person posts. E.g., Twitter, Facebook.

II. RELATED WORK

A. *Adaptive Video Streaming Techniques*

In adaptive streaming, the video traffic is adjusted so that user can view more quality video depending on user link's bandwidth capacity. There are two kinds of adaptive streaming techniques based on whether adaptivity is controlled by the client or the server. Microsoft's smooth streaming is streaming service provided by server. Adobe and Apple is an example of client side controlled adaptive streaming service. But, both services maintain redundant copies of video which increases burden of storage on server.

Rate adaptation controlling techniques used TCP-friendly control methods for streaming services to detect the link quality so that adaptation can be done accurately. But by using this technique the servers have to always control which results in large workload. To overcome this issue, the H.264 Scalable Video Coding (SVC) technique has been introduced. Through this technique quality oriented scalable video can be delivered. The high quality videos can be achieved using cloud-based proxy because cloud computing improves the performance of SVC coding.

B. Mobile Cloud Computing Techniques

Mobile cloud computing (MCC) is simply cloud computing in which at least some of the devices involved are mobile. Many mobile devices have significant constraints imposed upon them because of the importance and desirability of

smaller sizes, lower weights, longer battery life and other features. This often severely constrains hardware and software development for these devices. Cloud computing allows devices to avoid these constraints by letting the more resource intensive tasks be performed on systems without these constraints and having the results sent to the device. Thus, cloud computing for mobile devices is a very appealing and potentially lucrative trend. Recently usage of private agents has been implemented to satisfy the requirements of users. Thus, the cloud is designed using virtual agents to provide adaptive video streaming services. The CALMS framework is a cloud assisted live media streaming service for globally distributed users.

III CLOUD FRAMEWORK

The cloud framework includes two parts: Adaptive Mobile Video streaming and Efficient Social Video sharing. The framework is as shown in Fig. 1

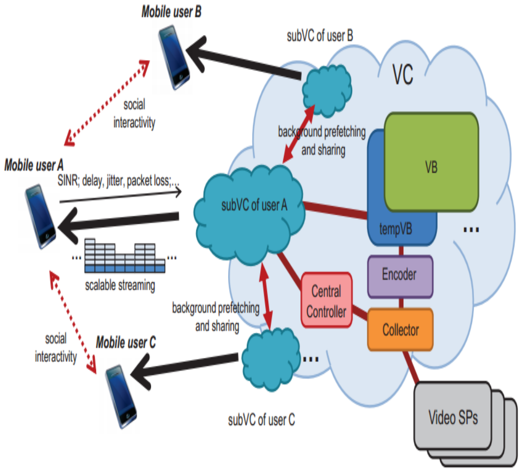

Fig. 1. Cloud framework

The whole video storing and streaming system in the cloud is called the Video Cloud (VC). In the VC, there is a large-scale video base (VB), which stores the most of the popular video clips for the video Service Providers (VSPs). A temporal video base (tempVB) is used to cache new candidates for the popular videos, while tempVB counts the access frequency of each video. The VC keeps running a collector to seek videos which are already popular in VSPs, and will re-encode the collected videos into SVC format and store into tempVB first. By this 2-tier storage, the Cloud can keep serving most of popular videos eternally. The management work will be handled by the controller in the VC.

Specialized for each mobile user, a sub-video cloud (subVC) is created dynamically if there is any video streaming demand from the user. The sub-VC has a sub video base (subVB), which stores the recently fetched video segments. The video deliveries among the subVCs and the VC in most cases are actually not "copy", but just "link" operations on the same file eternally within the cloud data center. There is also encoding function in subVC and if the mobile user demands a new video, which is not in the subVB or the VB in VC, the subVC will fetch, encode and transfer the video. During video streaming, mobile users will always report link conditions to their corresponding subVCs, and then the subVCs offer adaptive video streams. Note that each mobile device also has a temporary caching storage, which is called local video base (localVB), and is used for buffering and prefetching.

As the cloud service may across different places, or even continents, so in the case of a video delivery and prefetching between different data centers, an transmission will be carried out, which can be then called "copy". And because of the optimal deployment of data centers, as well as the capable links among the data centers, the "copy" of a large video file takes tiny delay.

IV AMOV: ADAPTIVE MOBILE VIDEO STREAMING

A. *Scalable Video Coding (SVC):*

For a particular bit rate, if the link's bandwidth differs much, the video streaming terminates frequently. In SVC, the three lowest scalability is combinely called as Base Layer (BL) and enhanced are called Enhanced Layers (EL).Hence if BL is guaranteed to deliver, a better video quality can be obtained.

Using SVC encoding techniques, the server need not to check the link quality or client. The client can decode the video and watch though some packets are lost. Yet this is not bandwidth-efficient because of packet loss. So, SVC-based video streaming must be controlled at the server side to use bandwidth efficiently.

V ESOV: EFFICIENT SOCIAL VIDEO SHARING

A. *Social Content Sharing:*

In Social Network Services, one can post the videos in the public to his/her subscribers to watch it, one can directly recommend the video to his/her friends or one can get noticed by publisher for new or popular videos. The probability of watching a video by recipients shared by one user is called “Hitting Probability” which will help in prefetching the video to avoid the delay. The amount of prefetched segments is determined by strength of social activities.

The social activities in social networks can be categorized into three types from view of recipient:

- Subscription: User can subscribe to a video publisher according to his interest. Since the subscriber may not watch all the subscribed videos, this can be considered as “Median”.
- Direct Recommendation: User can directly recommend a video to his friend in particular so the watching of video will have high probability. This is considered as “Strong”.
- Public Sharing: Each user has a timeline which shows all recent activities performed by user. The videos watched by the user will be known to his/her friends. Since not many people show the interest to watch video without direct recommendation, this is considered as “Weak”.

B. Prefetching Levels:

The social activities of mobile users can be defined through three prefetching levels:

- Parts: Because the videos published by subscription are watched with median probability, pushing a part of BL and EL segments is sufficient.
- All: The video shared by Direct Recommendation will be watched with high probability, pushing all BL and ELs is necessary so that user can watch the video with good quality without buffering.
- Little: Since the video shared by Public Sharing has low probability, prefetching of only BL segment of first window is enough.

TABLE I- SOCIAL ACTIVITIES AND BACKGROUND PUSHING STRATEGIES

Source	**Direct recommendation**	**Subscription**	**Public sharing**
VB →subVB	All	Parts	Little
subVB→locVB	All	Parts	Little
subVB→locVB	Parts	Little	None

VI Video Storage And Streaming Flow By Amov And Emos

The two parts AMOV and EMOS perform video streaming and sharing based on cloud computing platform. AMOV and EMOS use private agents to achieve better quality video without any buffering and interruptions by prefetching the user interested videos.

The flowchart of streaming video using AMOV and EMOS is given in below figure. To exchange the videos among the localVBs, subVBs, tempVB and the VB, a video map (VMAP) is used to refer the requested segments.

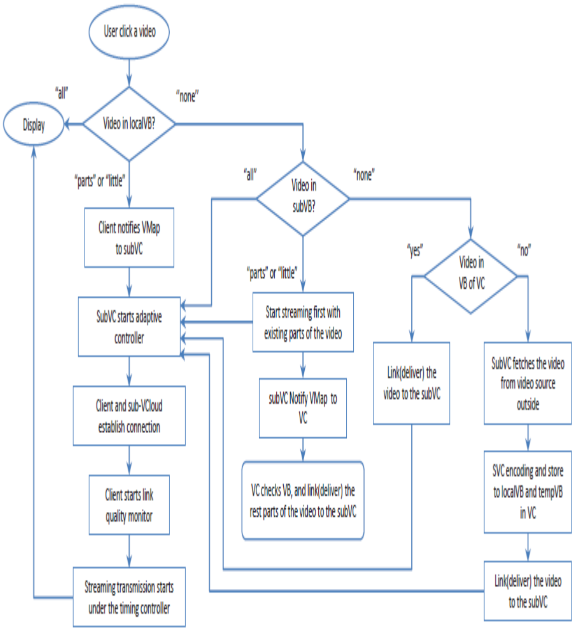
User click a video
"all"
"none"
Display
Video in localVB?
"parts" or "little"
"all"
Video in subVB?
"none"
Client notifies VMap to subVC
"yes"
Video in VB of VC
"no"
"parts" or "little"
SubVC starts adaptive controller
Start streaming first with existing parts of the video
Link(deliver) the video to the subVC
SubVC fetches the video from video source outside
Client and sub-VCloud establish connection
subVC Notify VMap to VC
Client starts link quality monitor
VC checks VB, and link(deliver) the rest parts of the video to the subVC
SVC encoding and store to localVB and tempVB in VC
Streaming transmission starts under the timing controller
Link(deliver) the video to the subVC

Fig.2. Working flow of video streaming in VC and subVC

Once a mobile user starts to watch a video, the localVB will be checked first for prefetched segments of video to display .If there is none or some parts, the client will report to subVC using VMap. If subVC has parts in subVB, the subVC will transmit the segments. But if there is also none in subVB, the tempVB and VB in center VC will be checked. If there is no video in VC, the collector in VC will fetch from external sources and re-encode video into SVC format, then subVC will transfer to the user. If video is shared among subVCs at predefined frequency threshold (e.g., 10 times per day), it will be loaded to the tempVB of VC so that it can be shared at higher frequency (e.g., 100 times per day). In such a way, the subVB and VB store fresh and popular videos for re-usage.

VII Video Sharing Over Untrusted Cloud Storage Providers

The specific security requirements for securing video storage in the cloud can be summarized as follows:

1. Video stored on the cloud should be kept private and the cloud storage provider should not be able to compromise the video confidentiality by any means.
2. The video owner has full control over authorization of video sharing. With authorization given by the owner, the designated user can then access the video kept on the cloud. Nevertheless, the process should not give the cloud provider any right to access the video.
3. Video access authorization is designated to the intended user only. Other users, who are not the permission holder, should not be able to exercise the permissions to access the video.

The above requirements of secure video sharing must be achieved via an untrusted cloud storage provider. It is necessary that the cloud storage provider helps to enforce the authorization policy for data access, but the enforcement should not reveal any information to the cloud storage provider or enable the cloud storage provider have excessive privileges to allow unauthorized access. The requirements can be achieved by using either homomorphic encryption or incremental encryption.

Homomorphic encryption is a cryptography scheme where algebraic operations applied on the ciphertext are directly reflected in the corresponding plaintext. Simply put, this allows a third party to compute the sum of two encrypted numbers, and when this encrypted result is returned to the user, it can be decrypted with the original key, and the result is the same as the sum of the two numbers in plaintext form. This allows multiple parties to cooperatively generate a piece of ciphertext without knowing the plaintext that others work on.

Incremental encryption allows the computation of the final ciphertext based on the initial ciphertext and the change of the plaintext. The mechanism allows users to have trusted video storage and sharing over untrusted cloud storage providers. Being able to implement trusted services on untrusted cloud storage providers allows users to manage their video on any cloud storage provider, eliminating the required trust on the providers. The video leakage prevention scheme is illustrated in Fig.3. The general idea is to encrypt the video before storing it in the cloud. On sharing the video, the encrypted video will be re-encrypted without being decrypted first. The re-encrypted video will then be cryptographically accessible only to the authorized user with the corresponding token.

The whole process does not reveal the clear video to the cloud provider at any stage, preventing the video being shared without the permission from the video owner. During the sharing, the video is always in its encrypted form, though at different stages it may be encrypted with different keys. There is no single stage that the video is decrypted into its clear form before it is delivered to the authorized users. This ensures that the whole sharing process will not disclose the information of the video to any parties.

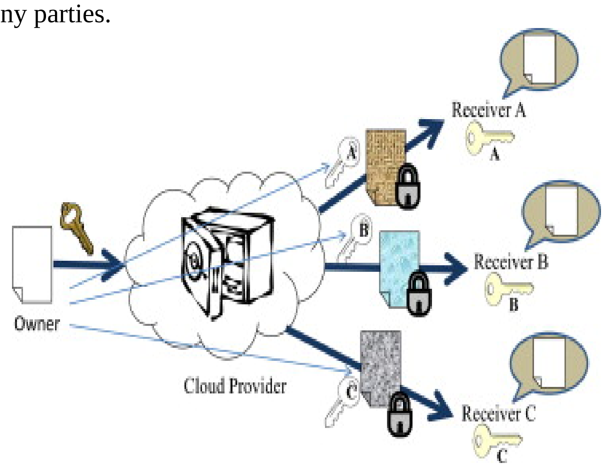

Fig.3. A cloud video leakage prevention solution

VIII Conclusion

Here, a framework for storing videos in clouds and sharing videos by constructing private agents for each mobile user to provide "non-terminating "video streaming by Scalable Video Coding technique has been proposed.This framework also provides "non-buffering "video streaming by pushing functions among VB, subVB and local VB of users and security was provided to videos in clouds using homomorphic and incremental encryption when they are stored in public clouds. This secure storage solution does not always fit as there are still a number of applications that rely on accessing videos in the cloud. It can be tried to improve the SNS-based prefetching in cloud framework in future.

Acknowledgement

This work was supported by M.Vikram, M.E., Associate Professor, Department of Computer Science and Engineering through his valuable guidance, constant encouragement, constructive criticism and keen interest.

References

[1] M. Wien, R. Cazoulat, A. Graffunder, A. Hutter, and P. Amon, "Real-Time System for Adaptive Video Streaming Based on SVC," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1227–1237, Sep. 2007.

[2] H. Schwarz and M. Wien, "The Scalable Video Coding Extension of The H. 264/AVC Standard," in *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp.135–141, 2008.

[3] P. McDonagh, C. Vallati, A. Pande, and P. Mohapatra, "Quality-Oriented Scalable Video Delivery Using H. 264 SVC on An LTE Network," in *WPMC*, 2011.

[4] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud Computing: State-of-the-art and Research Challenges," in *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, Apr. 2010.

[5] D. Niu, H. Xu, B. Li, and S. Zhao, "Quality-Assured Cloud Bandwidth Auto-Scaling for Video-on-Demand Applications," in *IEEE INFOCOM*, 2012.

[6] Z. Huang, C. Mei, L. E. Li, and T. Woo, "CloudStream : Delivering High-Quality Streaming Videos through A Cloud-based SVC Proxy," in *IEEE INFOCOM*, 2011.

[7] B. Aggarwal, N. Spring, and A. Schulman, "Stratus : Energy-Efficient Mobile Communication using Cloud Support," in *ACM SIGCOMM DEMO*, 2010.

[8] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A Survey of Mobile Cloud Computing : Architecture , Applications , and Approaches," in *Wiley Journal of Wireless Communications and Mobile Computing*, Oct. 2011.

[9] Gentry Craig. "A fully homomorphic encryption scheme".Ph.D. thesis, Stanford University; 2009. <http://crypto.stanford.edu/craig/craig-thesis.pdf>[retrieved 21.04.11].

[10] Bellare Mihir, Goldreich Oded, Goldwasser Shafi. "Incremental cryptography: the case of hashing and signing" In: *Advances in cryptology –CRYPTO'94*. Springer; 1994. p. 216–33.

[11] Bellare Mihir, Goldreich Oded, Goldwasser Shafi. "Incremental cryptography and application to virus protection" In: *Proceedings of the 27th annual ACM symposium on theory of computing*. ACM; 1995. p. 45–56.

[12] Zhao Gansen, Rong Chunming, Li Jin, Zhang Feng, Tang Yong "Trusted data sharing over untrusted cloud storage providers" In: *Proceedings of the 2nd IEEE international conference on cloud computing technology and science* (CloudCom 2010); 2010.

About Authors:

[1]**V.Manasa** (virupakshimanasa@gmail.com) is pursuing M.Tech in Computer Science and Engineering (CSE) from Sri Venkateswara College of Engineering,-Tirupati, JNTU-Anantapur. She received the B.Tech degree in the Department of Computer Science and Engineering from JNTU-Anantapur in 2012.

[2]**M.Vikram** (vikram.m@svcolleges.edu.in) is an Associate Professor in Sri Venkateswara College of Engineering-Tirupati, JNTU-Anantapur. He is pursuing Ph.D. in Computer Science from K.L.University. He received B.Tech degree from JNTU-Hyderabad. He obtained M.E degree from Anna University, Chennai. He has more than ten years of experience in teaching field.