A Multi-grained Video Encryption Method Based on Spark

Yang Zhou[1, a], Yingye Cheng[2, b]

[1]School of Information Engineering, Communication University of China, Beijing 100024, China

[2]School of Information Engineering, Communication University of China, Beijing 100024, China

[a]315337155@qq.com, [b]444091417@qq.com

Keywords: multi-grained encryption, selective encryption, distributed computing, Spark

Abstract. Video encryption has the requirement of efficiency, security, discrete storage in cloud environment. In this paper, a method of multi-grained video encryption based on Spark framework is proposed. The design of distributed encryption framework and multi-grained encryption module are described. The distributed framework improves the encryption speed, and coarse-grained encryption method is suitable for the discrete storage of video, fine-grained encryption method is suitable for centralized storage of video.

I. Introduction

With the development of Electronic technology and Internet, Video shooting, production and dissemination becomes easy, but it is a threat to the copyright protection of the video. First, identity authentication and access control techniques are commonly used in video transmission, the video content is not enough emphasis on data security. And the convenience of video communication provides a channel for the spread of piracy, that seriously damaged the interests of the content producers and operators. Second, with the acceleration of video production and the continuous improvement of definition, how to encrypt video efficiently is becoming a big challenge. Third, in cloud environment video is always stored in form of splits, these splits distributed in different servers which are physically separated with each other. The storage and distribution of video splits also put forward some special requirements for video encryption. In this paper, a multi-grained encryption method based on Spark is proposed to protect the video data in the cloud environment.

A．Spark Framework

Spark[1][8] as an in-memory distributed cluster system developed in the UC Berkeley AMP Lab. Resilient Distributed Dataset (RDD) is the main abstraction in Spark. RDD is a read-only collection of objects partitioned across a set of machines. RDD can only be created through coarse-grained deterministic operations based on data in stable storage or other RDDs. RDDs support two types of operations: transformations, which create a new dataset from an existing one, and actions, which return a value to the driver program after running a computation on the dataset.

B．Video Split

The split operation of video is usually to add a header to each split, and generates an index file. Header contains flags like PID, SID, CW, MD5, length of Split. Splits and their storage locations are corresponded in the index record. Clients receive index first when play video, then get splits of video base on this index file[9].

C．Selective Encryption

Video files are always compressed into a variety of formats like MPEG-2, MPEG-4, H.264 and so on. Take MPEG-2 as an example, it provides a six-layers structure. They are Sequence, Group of Picture, Picture, Slice[5], Macro Block, Block from top to bottom. Top 4 layers have their own start code that can be used for synchronization. Video transmission process will often be packaged as transport stream. It is used in broadcast systems such as DVB, ATSC and IPTV. A packet is the basic unit of data in a transport stream, and each packet starts with a sync byte and a header. In MPEG-TS[6][7] package, we only need to find the position of payload and do encryption processing, and need not to encode MPEG. I frame, slice, DCT all can be used for selective encryption[4].

II. Distributed encryption system based on Spark

In this part, we will introduce a distributed encryption system based on Spark. In order to achieve efficient encryption, Spark framework is chosen for distributed encryption.

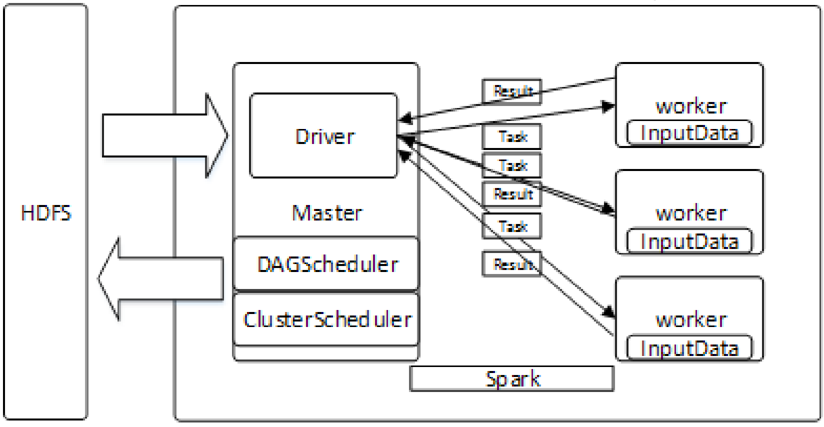

Fig.1 Distributed Processes based on Spark

As shown in Fig.1, specific procedures for distributed processing in detail will be described in the follow text.

Step1: Video files should be divided into splits, then store them in the HDFS. Spark fetch splits from HDFS[3], meanwhile resilient distributed datasets RDD should be structured for Spark.

Step2: master of Spark process each element at the base of node performance and usage through agent node cooperative algorithm for shared memory. And encryption processing of video splits is finished in Workers. Encryption module is deployed on each node. Encryption processing will start when splits distributed into Workers.

Step3: after the encryption of splits finished in Worker, the elements should be returned to master.

Step4: master stored encrypted datasets in HDFS through the interface of Hadoop. Finally the encrypted patch information should be returned into the server cluster which split video files, then enter into content distribution network.

III. Multi-grained Encryption

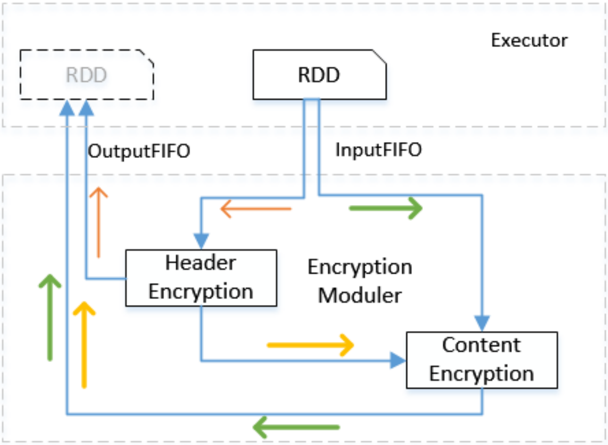

Fig.2 Multi-grained Encryption Module

In this paper, multi-grained encryption module is designed according to the characteristics of the video split, and selective encryption method is referenced in encryption module. The encryption module contains two sub modules: coarse-grained header encryption and fine-grained content encryption. Header encryption is a high speed encryption, because only head of the video split need to be encrypted, and the amount of encrypted data is less. In content encryption, I frame, slice, DCT, MV all can be selected to be encrypted. We chose slice encryption method in content encryption. Because slice information of video still contains a large amount of data, so the encryption speed is lower than the header encryption. A 128-bits AES encryption is used in this system as Encryption algorithm.

Two encryption methods are adapted to different conditions. When the size of the video splits is small and storage location of splits is discrete, the header encryption can be effectively guaranteed, although the content of the data is not encrypted. If someone stole one of the splits, he couldn't watch the whole video, because he do not know the storage location of other splits and the sequence in the video. When the size of splits is too large, the storage location is relatively concentrated, the security of header encryption is greatly reduced. It is easy to restore the whole video if illegal get splits because the video content has not been encrypted, so it is necessary to encrypt the contents of the splits. When content encryption still unable to meet the security requirements, we can do header encryption first, and then put result of header encryption into content encryption module for twice encryption. We need to get the header encryption key and content encryption key in order to successfully decrypt the video.

IV. Discussion

We will discuss two aspects in the follow text:

First, efficiency with two different encryption method, header encryption and content encryption.

Second, compare the effect of encryption before and after content encryption.

We built a 1 to 2 master-slaves mode server cluster in the lab. 512MB MPEG-TS video file is chosen for experiments. Video file is split into 64MB, 16MB, 8MB size splits and all these splits have been stored in HDFS. The cluster hardware configuration in the experiment is showed as follows [Table1]. Encryption depth is 50%, that make sure splits can be played by video player.

Table 1 Hardware Configuration per Server

Configuration per Server	
CPU	24cores
Memory	16GB
OS	Cent OS 6.5
Hadoop	Hadoop 2.7.1
Spark	Spark 1.6.0
SPARK_WORKER_INSTANCES	3
SPARK_WORKER_CORES	4
SPARK_WORKER_MEMORY	3

Results of comparison between header encryption and content encryption with Spark shown in the follow Table2.

Table 2 comparison of encryption time

Size of Split Method of Encryption	64MB	16MB	8MB
Header Encryption	4.5s	4.5s	4.3s
Content Encryption	12.3s	11.8s	10.2s
Header & Content Encryption	3.6s	13.1s	12.8s

Contrasting content encryption, header encryption has obvious advantage in the encryption efficiency. Time of encryption consumed roughly the same with different size of splits on Spark framework from the result of experiments. Intuitive results as shown below in the Fig.3. We can set a threshold for the degree of dispersion, do header encryption once the threshold is exceeded.

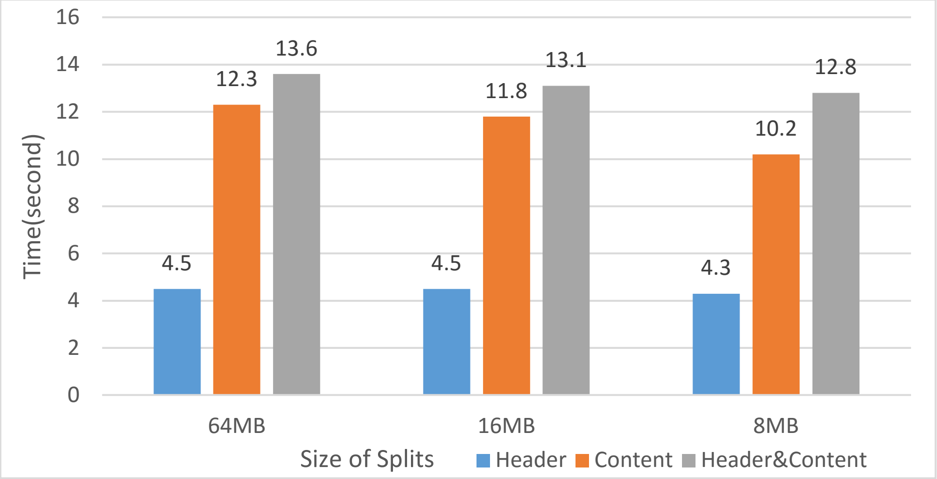

Fig.3 Comparison of Encryption Time

Since all the video data is not encrypted with header encryption, it is possible to restore whole video if we get all the splits. And every split can't be played after content encryption normally. Effect of 50% depth encryption in the Fig. 4.

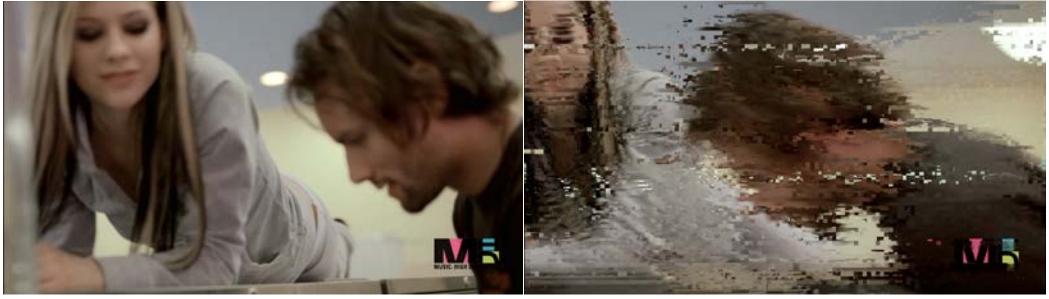

Fig.4 Effect of 50% Depth Encryption

V. Conclusion

In this paper, a method of multi-grained video encryption based on Spark framework is proposed. Because video encryption has the requirement of efficiency, security, discrete storage in cloud environment. The coarse-grained encryption method is suitable for the discrete storage of video, fine-grained encryption method is suitable for centralized storage of video. Experimental results show that the coarse-grained encryption has a great advantage in efficiency, fine-grained encryption can provide better protection for video.

Acknowledgement

The work on this paper was supported by the National Science and Technology Support Plan (2014BAH10F00、2014BAH10F02-01、2014BAH10F02-02).

References

[1] M. Zaharia, M. Chowdhury, S. S. Michael J. Franklin, and I. Stoica, Spark: Cluster computing with working sets. In HotCloud. 2010.7.

[2] Information on http://spark.apache.org

[3] Information on http://hadoop.apache.org

[4] Zhang Yue. Analysis and research on the region of interest in the video encryption. The Communication University of China.2013.5.

[5] Wang Lei. Distributed Video Scrambling System Based on Hadoop. The Communication University of China.2013.5.

[6] Qao L, Nahrs tedtK. A new algorithm for MPEG video encryption [A]. In: Proceeding of the First In Ternational Conference on Imaging Science, Systems and Technology (CISST'97)[C] , Las Vegas , Nevada, 1997: 21-29.

[7] ISO/IEC 13818 -1 ~ -2 Information technology- Generic coding of moving pictures and associated audio information

[8] Zhijie Han, Yujie Zhang.Spark: A Big Data Processing Platform Based on Memory Computing[A]. Parallel Architectures, Algorithms and Programming (PAAP),2015,172-176.

[9] Information on https://tools.ietf.org/html/draft-pantos-http-live-streaming