SECURE CLOUD BASED VIDEO ENCODING USING HIERARCHICAL PROFILE-BASED CRYPTOGRAPHY

Ashwath S[1], Balasubramaniam T[2], S P Tamizhselvi[3]
[1,2,3]Student, Department of Computer Science & Engineering
KCG College of Technology, Karapakkam
Chennai, TN, India

Abstract

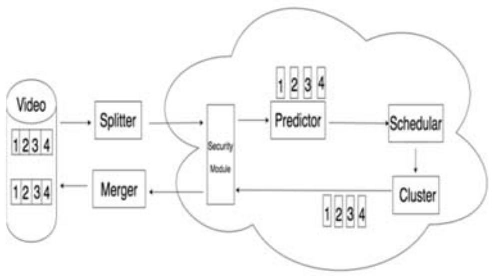
Multimedia has conquered today's world and most of the people on the planet are dependent on it either way. Video is one of the major aspects of multimedia and is being used throughout the world for various purposes. But there are several platforms for it and it faces the issue of incompatibility. Video encoding is a technique used as a solution for this issue. With video encoding we can make a video run on multiple platforms by encoding it ,thereby solving the issue of incompatibility. In this technique, the input video is divided into a bunch of segments and these segments are taken as sub-tasks and processed.. This happens in the cloud environment and is based on the map reduce concept which is used to deal with video encoding. Though it is being done world wide, the currently used techniques fail to address the security issues.The proposed model overcomes this issue by using hierarchical profile-based cryptography. This concept addresses the lack of security by using the user profile which is used as a key to secure the data in cloud. Key management is one of the biggest problems in the cloud and the proposed model overcomes this issue by using hierarchical key management technique. In this technique the key is being stored in a hierarchical manner so that it can be accessed quickly thereby reducing the processing time.

Keywords: Video encoding, Hierarchical profile based cryptography, map reduce, key management, digital profile.

Introduction:

As video has become an important from of multimedia and being used throughout the globe, it plays a very crucial role in many fields. Since several platforms are available for a video to be run on, it is necessary that a video must be compatible for all platforms. In this case, a process called video encoding is used.Video encoding can be defined as the process of encoding a video to a different format in order to run it in a different platform. It addresses the issue of incompatibility. Though it is efficient in terms of encoding, it stays behind in the aspect of security. In this process, the video fragments are highly vulnerable to attacks and needs immense security. In order to provide a solution for this issue, a model is proposed, which provides profile based cryptographic security over the video parts through out the process.

encoding architecture(Security module incl.)



The system consists of a complexity predictor, scheduler, splitter, encoding processing cluster and merger. A video is made of one or more groups of pictures . Generally, if the input sequence has a group of pictures, the encoding task can be divided into several sub-tasks. Here we assume the input to be closed group of pictures. On the arrival of a process request, the splitter divides the input into several segments. The complexities of segments are heterogeneous as these segments have different lengths. Also, they can be processed on different systems in random order. Once the video is divided into multiple segments, the predictor predicts their desired complexities. Later, all the sub-tasks are mapped to the processing cluster by the scheduler. The actual process is carried out by the processing cluster, which contains different systems with multiple cores and varying processing powers.Once all the systems in the processing cluster have finished their processes, the entire encoding process is complete. Following it, all the processed sub-tasks are sent to the merger, where they get concatenated to produce the desired output file.

Prediction of complexities :

Encoding process is defined as a process of conversion of a video file from one format to another . In order to make the predictions, the system calculates the complexities of decoding and encoding.

The complexity calculation for decoding is as follows:

$V_{decode} = \Sigma V_{mb\_decode}$ **----1.1**

The complexity calculation for encoding is as follows.

$V_{encode} = \Sigma \{K m_{len} + m_{len} + C_{other}\}$ **----1.2**

Task scheduling algorithm :

Our model is also NP Hard model. It is quite time consuming and complex to obtain solution in NP-Hard problem. Min-Min and Max-Min are the heuristic algorithms that can address these schedule problems. A combination of Max-Min and Min-Min algorithms are being used . With this, a prediction-based algorithm is being applied.The algorithm is as follows.

Profile-based cryptography :

Profile-based cryptography is a public-key based scheme that works well when there is an exchange of message between two sides and they verify each other's digital signature. In profile based cryptographic systems the user's profile which uniquely identifies a user is being used as public key for authentication purpose. Profile- based cryptography simplifies the complexity of key management. Secure distribution of public keys is not happening on the first place. Authentication can be conducted offline which stands as an added advantage to it.

In the profile-based cryptography approach, a module named PK initially generates a master public key ,followed by a corresponding master secret key. Then it will make the master public key accessible to all the interested users. Any interested user can make use of this master public key to create a public key for that user with the user's profile. A user needs to contact the PK with his profile to obtain his secret key. PK with the profile and the "master" secret key of the user, generates the secret key for that user. There are four steps for a complete profile-based cryptography system. They are setup, extract, encryption and decryption.

1. Setup: PK creates a master key along with the parameters *P* where *master key* is kept secret and used to generate

secret key for users while the parameters P are made public and can be used to generate users' public key using their identities for all users.

2. Extract: On receiving a request for secret key from a user , the PK ,using the profile of that user, parameters *P* and master key ,generates a secret key for that user.

3. Encryption: A user who wants to encrypt a message and send to another user can use the parameters *P*, receiver's profile and the message as input and generate the cipher text.

4. Decryption: On receiving a cipher text, the receiving user can use the parameters *P* and the secret key he obtained from PK to decrypt the cipher text.

In a network which uses profile-based cryptography, generation of secret keys for all the users is not the only job for PK, it must also verify the user identities and establish secure channels through which the secret keys are transmitted. In case there is only one PK in a large network, the PK will have a burdensome workload. HPBC can be a better choice in such cases. In a HPBC network, a root PK will generate and send secret keys for domain-level PKs in a distributive manner and the domain-level PKs will generate and transmit secret keys to the users in their own domain in the same manner. HPBC is suitable for a large scale network as it addresses the issue of high workload by reducing the load on root PK by distributing the work of user authentication and secret key generation to the lowlevel of PKs. It improves the security of the network by locally performing the processes of user authentication and secret key distribution. The HPBC encryption and signature algorithms comprises of the following steps.

1. Root setup: The root PK generates root PK parameters along with a root secret. The root secret is used for secret key generation for the subsequent lower-level PKs. The root parameters will be made publicly available and are used for the generation of public keys for lower-level PKs and users respectively.

2. Lower-level setup: Each lower-level PK obtains the root parameters and generates its own lower-level secret. This lower-level secret is then used to generate secret keys for the users in its domain.

3. Extract: Once a user or PK at level s with the profile requests the secret key from its upper-level PK, the upper-level PK will use it's profile and generate a secret key for that user.

4. Encryption: User who wants to encrypt a message takes the parameters, receiver's profile and the message as input and generates the cipher text.

C = Encryption (parameters, receiver profile, Message).

Decryption: On receiving a cipher text,the receiving user can use parameters and his secret key and decrypt the cipher text.

M = Decryption (parameters, Secret key, Cipher)

Signature and verification: A user generates a signature using the parameters, its secret key, and message and sends it to the receiver. The receiver receives and verifies it using the parameters, message and the sender's profile.

Signature = Signing (parameters, secret key, Message)

Verification = (parameters, sender profile, Message,Signature).

There are a few inherent demerits for profile-based cryptography. The key escrow problem is one of the major issues. As PK generates the users' secret key, it can decrypt a user's message and with no authorization it can create any user's digital signature. This indeed requires the PKs to be highly trusted. So the profile- based scheme is suitable only for a closed group of users . A big company or a university can be considered an example of it. Only under this kind of situations, PKs can be set up with users' trust.

In a HPBC system, every PK knows the users' secret keys in the domain under the PK, in the hierarchy . Another limitation of the profile-based cryptography is the issue of revocation.

Using Federated profile Management in Cloud -Federated profile Management in the Cloud

Federated profile is way ahead of the centralised profile which deals with the security problems within same networks, by adopting to deal with the problems associated with users accessing external networks or external users accessing internal networks.Federated profile is a standard mechanism for different organisations to share their identities and it enables the portability of identities across networks.One important use of federated profile is to secure single sign-in.Usage of federated profile mechanism can increase a network's security as it only requires a user authenticate for once and the profile information can be used with multiple networks. Profile federation standards help the users from different networks to trust each other.

Implementation of profile federation in the cloud is where the users from different clouds use a federated profile identification to identify themselves, which satisfies the requirement of profile based cryptography in cloud. Entities in the cloud have their own unique profiles which are hierarchical in nature. In order to access cloud services, a user needs to authenticate himself. In few cases, servers must also authenticate themselves to the users. In this paper, we add the federated profile management and HPBC in the cloud for the purpose of security. In this, cloud trusted PKs are used and unlike traditional PKs, they can allocate hierarchical identities to the users in their domains .There is a root PK present in the first level followed by other PKs and users in the subsequent levels in the low domains. The root PK will authenticate the identities for those PKs and users in the second level and so on.

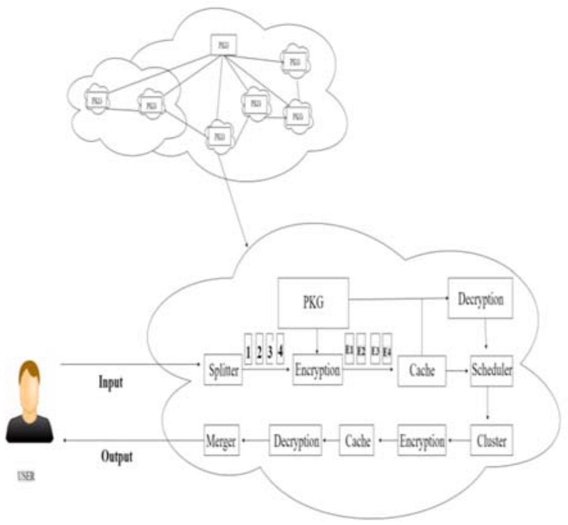
Message Encryption and Signature

In a cloud with federated profile,an entity has a unique profile and can get the profile of any user by requesting the respective PK. The public key distribution process can be made ease and simple in the cloud with the help of HPBC. The entities need not request a public key directory to obtain the public key . In case an entity needs to encrypt the data that is transmitted in cloud, the sender acquires the profile of the receiver, then encrypts the data with it.

Currently, Web Service-Security protocol that provides end-to-end message level security with the use of SOAP messages is widely implemented in cloud computing for the purpose of security of most of the cloud computing related web services. Web Service-Security makes use of SOAP header to carry security information. Since SOAP message is of XML type and generally XML message representation is larger than their equivalent binary formats, adding security information in the SOAP header results in increased costs in data communication and data parsing.On using HPBC in a cloud environment, any user can get his secret key from the PK of his domain and can calculate the public key of any other user in the cloud provided his profile is known. Later it will be easy for a sender to attach a signature with its secret key and for a receiver to verify a signature using the sender's public key. The key information may be not needed in the SOAP header, and so this will enormously reduce the amount of SOAP messages needed to be transmitted in the cloud. Thus, it will result in saving the cost.

Proposed model:

Architecture:



Work Flow:

1.The user A firstly needs to register himself to the cloud , if he is not registered.
2.Assuming that A is a registered user, A sends the video to be transcoded to the splitter.
3.The splitter splits the video into a number of chunks and sends them to the cloud. A number of video fragments is the output of splitter module.
4.The video fragments enters the encryption module, where it gets encrypted with the help of PK.

a) PK acquires the user's key request message and then checks the user's profile module.
b) If the user satisfies the requirements, the PK provides the user with public key and secret key of the user.
c) Encryption module obtains the key pair and uses the key to encrypt the data

C= Encrypt(Parameter, user ID,M)
S= signing(parameter,K,M)

5.The encrypted module is stored in the cache until the scheduler becomes free to handle the current data.
6.The scheduler requests the cache for the data.The cache sends the data to scheduler via decryption module and the scheduler gets the decrypted data to process further.
P= Decrypt(Parameter, K,C)
V=(parameter,userID,M.signature)

7.The scheduler considers these video fragments as sub-tasks and assigns them to different available machine in the processing cluster.
8.Once processed, these sub-tasks are again encrypted and sent to the cache.
9.The cache memory forwards the data to the decryption module where it gets deciphered.
10.The deciphered message comprising of encoded video fragments is sent to the merger for the purpose of concatenation.
11.The merger concatenates all the sub-tasks or video chunks and produces the properly encoded video as output to the user.
12.The encoded video obtained is now ready to run on different platforms.

Algorithm:

```
1.  Assume V as the video input size.
2.  Get the user ID and system parameters .
3.  Split V $\rightarrow$ V/2 chunks.
4.  Encryption :
    C= E(Parameter, user ID,V)
5.  Get the sub-tasks ST using for loop,
        For each $v_i$ do:
            ST=ST Ù $ST_i$
6.  Run a while loop to find the minimum of sub-tasks.
    While ST ǂ ϕ do:
      Min= infinity
        For each system do
        Calculate ET and EF values
            if ET + EF <min
               ET +EF = min;
        System m is assigned to min

    a)  Calculate $V_{min}$ and $V_{max}$ and
        Calculate $S_{curr} = V_{max} / V_{min}$
          if  $S_{curr} > ϕ$ then:
                Get $st_{x,y}$ with
          min (x,y)= max (min(i,j))
          Else:
                Get $st_{x,y}$ with
            min(x,y)=min(min(i,j))
      End while.

7.Encoding takes places by using the encoding complexities in respective system.
    Apply equations 1.1 and 1.2.

8. Decryption formula:
P= Decrypt(Parameter, K,C)
VE=(parameter,userID,M.signature)

9. Merge v/2 chunks $\rightarrow$ v chunks
```

Conclusion:

Thereby, it can be concluded that the proposed module gives reliable security over the video encoding process and also secures the video fragments throughout the process from various kinds of compromises. Also it gives a secure feeling to a user who uses this process for encoding their video content in the cloud. As security is very much important in a cloud environment, this proposed model very well satisfies the security needs.

References:

[1] Predication-Based and Locality-Aware Task Scheduling for Parallelizing video Transcoding over Heterogenous map-reduce cluster. By hui zhao, quinghua zheng,weizhan zhang, jing wang.
[2] Strengthen cloud computing security with Federal-identity Management using Hierarchical Identity-Based Cryptography by liang yan, chunming rong,Gansen zhao.
[3][Beak, J., Newmarch, J., Safavi-Naini, R., Susilo, W.: A Survey of profile-Based Cryptography. In: Proc. of the 10th Annual Conference for Australian Unix User's Group (AUUG 2004), pp. 95–102 (2004)
[4] Boneh, D., Franklin, M.: profile-based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 433–439. Springer, Heidelberg (2001)
[5] Chappell, D.: A Short Introduction to Cloud Platforms, http://www.davidchappell.com/CloudPlatforms–Chappell.pdf
[6] Cocks, C.: An profile-based Encryption Scheme Based on Quadratic Residues. In: Proceeding of 8th IMA International Conference on Cryptography and Coding (2001)