CRYPTOGRAPHIC APPROACH FOR ACCESSING VIDEO CONTENT USING TIME-DOMAIN ATTRIBUTE BASED ACCESS CONTROL

Kousalya Selvam[1], Ilayabharathi Selvaraj[2], Sumana Manickan[3], P. Veeralakshmi Ponnuramu[4]

[1] *Student, Department of Information Technology, Prince Shri Venkateshwara Padmavathy Engineering College, Tamilnadu, India*

[2] *Student, Department of Information Technology, Prince Shri Venkateshwara Padmavathy Engineering College, Tamilnadu, India*

[3] *Assistant Professor, Department of Information Technology, Prince Shri Venkateshwara Padmavathy Engineering College, Tamilnadu, India*

[4] *Associate Professor, Department of Information Technology, Prince Shri Venkateshwara Padmavathy Engineering College, Tamilnadu, India*

ABSTRACT

In recent years, the demands on these multimedia applications, cloud computing, due to its social and efficient resources is becoming a platform to process and store large amount of video contents. In this paper, we mainly focus on how to securely share video contents to a certain group of people during a particular time period in a cloud-based multimedia system. A secure Time-domain Attribute-based Access Control (TAAC) scheme, to secure the cloud-based video content sharing. This project additionally uses a secure time-domain attribute-based encryption scheme by embedding the time into both the cipher-texts and the keys, so the user holding specific attributes can able to decrypt the video contents. In this project, we also proposed an efficient attribute updating method to achieve the dynamic change of user's attributes, and granting new attributes and re-granting previously revoked attributes. So the TAAC technique will provide more security for video contents.

Index Terms— *TAAC, Video Content Sharing, Time-domain, CP-ABE, Multimedia, Cloud Computing.*

1. INTRODUCTION

This project deals with the rapid and successful development of communication technologies and mobile devices, video applications (e.g., video, video conference, chat etc) have become more and more popular in our daily life. Meanwhile, the demands on video quality and user experience have also been increasing significantly in many video applications, such as Ultra-high definition (UHD) live streaming, 3D movies, instant high definition (HD) video messages and so on. Thus increasing demands on video processing, coding, presentation as well as communication, especially when the resources of media devices (e.g., bandwidth, strength power and computation) are limited. Cloud computing, due to its flexible [1], is a natural fit for storing, processing and sharing multimedia contents.

Fig 1. Live Video Streaming at Different Time Periods

In cloud-based multimedia systems, some contents may be time-sensitive and can only be accessed by a certain group of people during a particular time period. For example, as shown in Fig. 1, if a particular user only purchases a live streaming service for time period t2, this user may be granted to access the live streaming or the videos recorded in time period t2. However, this user does not have any permission to access the live streaming in time period t3 or videos recorded in time periods t1 and t3 when he/she does not purchase the service. Moreover, during each time period, users may purchase different types of services, e.g., online live streaming service, usual recorded video service, HD recorded video service, UD recorded video, UHD recorded video service, etc. Therefore, it is necessary to achieve fine-grained time-domain access controls for diverse video content sharing. When outsourcing video contents into the cloud sever means, it is not at all easy to achieve the fine-grained access control on

cloud and especially in time-domain, as the owners of video contents are not able to control their own video. The untrustworthy cloud servers further make this issue more challenging, because: 1) cloud servers is not fully trusted by the owners to control the access of their video contents; and 2) cloud servers may also be more curious about the stored video contents. Thus, existing server-based accessing control methods (e.g., Access Control Lists) are not applicable for cloud-based video content sharing. A possible approach is to encrypt video contents and only authorized users are given decryption keys. From above example, all the video frames in t1 are encrypted by one key, while all the video frames in t2 are encrypted by another key. Therefore, due to the large amount of video contents and the performance requirements (e.g., speed, visual quality, compression friendliness, etc.), traditional encryption methods (e.g., AES, DES, DSA, RSA, etc.) may not be suitable for data encryption. The contributions of this paper are summarized as follows. 1) We further provide the time-domain video content sharing problem in cloud-based multimedia and propose a cryptographic approach - Time-domain Attribute-based Access Control (TAAC) scheme. 2) We proposed a provably most secure time-domain attribute-based encryption scheme by embedding the time into both the cipher-text and the keys, such that only users who hold sufficient attributes in a specific time period can decrypt the data. 3) We propose an efficient attribute updating method to achieve the dynamic change of users' attributes, including granting new attributes, revoking previous attributes and re-granting previously revoked attributes. We first review some existing work that may be related to secure video sharing and time-domain access control in Section II. Section III describes the technique overview and the step-by-step construction of TAAC and Section IV shows how to solve the attribute dynamic updating problem. Section V provides security analysis and performance evaluation and Section VI provides Conclusion. Thus overall it allows user to make use of the necessary video content which is being uploaded in the cloud server and user with appropriate key can decrypt the content.

2. RELATED WORK

This Cloud-based multimedia content sharing is one of the most significant services in cloud-based multimedia systems. It allows some security and privacy issues by exploring the multimedia-oriented mobile social network. In, Attribute-based Encryption is adopted to share scalable media based on the attributes rather than the names of the consumers. In this, they focus on deal with the security issues in wireless sensor networks, which is important in multimedia data collection and transmission. Multi-authority Cipher-text Policy Attribute-based Encryption (CP-ABE) is regarded as one of the most appropriate techniques for access control of data stored in the cloud, because it allows the data owner to define and enforce the access policy over attributes from multiple attribute authorities.

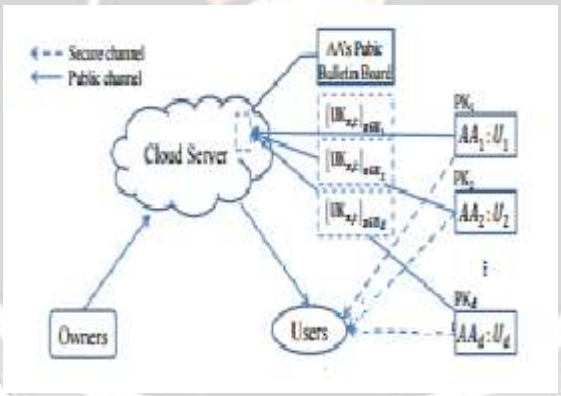

Fig 2. System Model for Public and Secure Channel.

To achieve access control in time-domain, the main challenging issue is how to embed the time into the ABAC. This method is to take time as an important attribute and embed it into the access policy by changing the access policy. The fig.2 explains about the basic distribution of necessary video content being shared in a public as well as secure channel. If video shared in a public cloud means then all user can able to access it. In case of a secure channel means only user holding specific attributes can able to decrypt the video and they cannot outsource the content.

2.1 System Model for Public and Secure Channel

The time in the system is slotted and the time space is defined such as T= {1,2..}. The system initialize its time to 0, and then increase to next time slot as 1. So, the value for the next time slot is t+1 and so on. The system model consists of necessary Attributes as an entity. Like attribute authorities (AA s), cloud server, data owner, user. Each AA is an independent authority that is responsible for revoking or re-granting attributes from/ to users. In this, each attribute is associated with single entity AA', but it is being managed by available number of attributes.

2.2 Framework for TAAC

Definition 1. *TAAC is a collection of the following algo- rithms:* Global Setup, Authority Setup, S Key Gen, U Key Gen, D Key Com, Encrypt *and* Decrypt.

- **Global Setup** (λ) $\rightarrow$ GPP. The global setup algorithm takes the security parameter λ as input. It outputs the global public parameters GPP.

- **Authority Setup** $(GPP, U_d) \rightarrow (PK_d, MSK_d)$. The authority setup algorithm is run by each *AA*. It takes as input the global public parameters GPP, an attribute domain U_d.

TABLE 1
NOTATIONS

SYMBOL	MEANING
T	Slotted time slot
D	Set of all attributes
U_d	Attribute set managed by AA_s
U	Universe attribute set element
CT	Cipher text with policy attributes
G id	Global identity
Sid	Attributes for g id
SK g, id	Secret key for attribute x
ST x	State tree of attribute x
UL (x, t)	Update list for attribute x time slot t
UK ((x, t))	Update key of attribute x time slot t
DK g, id	Decryption key of attribute time slot t

- **S Key Gen** $((g, id), x, ST_x, GPP, MSK_{\varphi(x)}) \rightarrow (SK(g\ id), x, ST_x)$. The secret key generation algorithm is run by each *AA*.

- **U Key Gen** $(t, x, ST_x, UL((x, t)), GPP, MSK_{\varphi(x)}) \rightarrow (UK((x, t)))$. At each time slot t, for each attribute $x \in U_{\varphi(x)}$, the authority $AA_{\varphi(x)}$ runs the update key generation algorithm.

- **D Key Com** $(SK_{g\_id}, , UK((x, t))) \rightarrow (DK_{g\_id,((x, t))})$ or $\perp$. For any time slot t and any attribute x, a user *g-id* can run the decryption key computation algorithm with secret key $SK_{g\_id,x}$ and update key $UK((x, t))$ as inputs.

- **Encrypt** $(M, te, A, GPP, \{PK_d\}) \rightarrow (CT)$. The encryption algorithm is run by data owners. It takes as inputs a message M, a time slot t_e, an access policy A over attributes from multiple authorities, the global public parameters GPP, and the public keys $\{PK_d\}$ related to A. It outputs a cipher-text CT which includes A and t_e.

- **Decrypt**$(CT, GPP, \{PK_d\}, \{DK_{g\text{-}id,((x, t))}\}_{x \in S_{g\text{-}id,t}}) \rightarrow (M)$ or $\perp$. The decryption algorithm is run by users. It takes as inputs a cipher-text CT which includes access policy A and time slot t_e, and the global public parameters GPP, the public keys $\{PK_d\}$ related to A, and decryption keys $\{DK_{g\text{-}id,((x, t))}\}_{x \in S_{g\text{-}id,t}}$ corresponding to a (global identity, time slot) pair $(g\text{-}id, t)$.

.

3. TAAC: TIME-DOMAIN ATTRIBUTE-BASED ACCESS CONTROL

In this section, we first go through the main ideas and techniques of TAAC.

3.1 Technique Overview

Due to the large volume of video contents and the performance requirements (e.g., speed, visual quality, compression friendliness, user-friendly, time-sharing etc.), video contents are encrypted by using video encryption methods with the help of session keys. To support time-domain access control of video contents, we control the distribution of session keys by proposing a new time-domain attribute-based encryption method, which embeds the time into both the cipher-text and the keys to the multi-authority CP-ABE.

3.2 Construction of TAAC Technique

Based on the algorithms defined, TAAC contains the following phases: System Initialization as phase [1], Key Generation by AAs as phase [2], Data Encryption by Owners as phase [3], and Data Decryption by Users as phase [4].

Phase 1: System Initialization

The system initialization consists of two steps: Global Setup and Authority Setup.

1)Global Setup

The system is initialized by running the global setup algorithm Global-Setup. Let S and ST be a bilinear group of order p, where p is a prime-2. Let s be a generator of S. The global public parameter is published as SPP = (b (C, f), where E is a bilinear pairing, and H is a hash function that maps global identities to elements of S.

2) Authority Setup

The authority setup algorithm Authority-Setup is run by each authority A Ad (d ∈ D). For each attribute x, the algorithm chooses a random of exponents exactly.

Phase 2: Key Generation by AAs

The key generation contains both Secret Key Generation as well as Update Key Generation.

1) Secret Key Generation

The secret key will be generated by the cloud server for each and every video being uploaded by the data owner, a secret key is similar to session key.

2) Update Key Generation

An update key will be provided to new user who is newly involved in creating an attribute. It can be done by using the revocation method.

$$\mathsf{UK}_{((x,\,t))} = \{(E_{v_x} = (R_{v_x})H_{\varphi(x)}(((x,\,t))),\ E_{v_x} = g)\}_{v_x \in N_{((x,\,t))}}.$$

Phase 3: Data Encryption by Owners

The owner first encrypts the video content with a session key by using video encryption algorithms. It then runs the encryption algorithm Encrypt to encrypt the session key κ. The encryption algorithm is defined.

$$\mathsf{CT} = \mathsf{h}\ (A, \rho), t_e,\ C = \kappa \cdot e(g,g)^{s},$$
$$\{C_{i,1} = e(g,g)^{\lambda_i} e(g,g)^{\alpha_{\rho(i)} r_i},$$
$$C_{i,2} = g^{\mu_i} g^{\beta_{\rho(i)} r_i},$$
$$C_{i,3} = g^{r_i},\ C^{'}\}.$$

Phase 4: Data Decryption by Users

All legal users can download any cipher-text they are interested in. But only the users who possess eligible attributes (satisfying the access policy A) at a particular time slot t can decrypt the cipher-text associated with (A, t-e). The decryption phase consists of two steps: Decryption Key Computation and Cipher-text Decryption.

1) Decryption Key Computation

At each time slot t, each user can get update keys for each attribute it possesses at this time slot from the public methods of accessing.

$$\mathsf{DK}_{g\text{-}id,((x,\,t))} = (\ D_{g\text{-}id,((x,\,t))} = K_{g\text{-}id,x,v_x},$$

$$_{g\text{-}id,((x,\,t))} = (E_{v_x})^{\,g\text{-}id,x,v_x} \cdot K_{g\text{-}id,x,v_x},$$

$$D_{g\text{-}id,((x,\,t))} = (E_{v_x})).$$

Only the users who possess x at time t can compute a valid decryption key, i.e., a user *g-id* who is entitled x at a later time slot $t' > t$ is unable to compute a valid $\mathsf{DK}_{g\text{-}id,((x,\,t))}$ even he has $\mathsf{SK}_{g\text{-}id,x}$ (issued at t') and $\mathsf{UK}_{((x,\,t))}$ (generated at t).

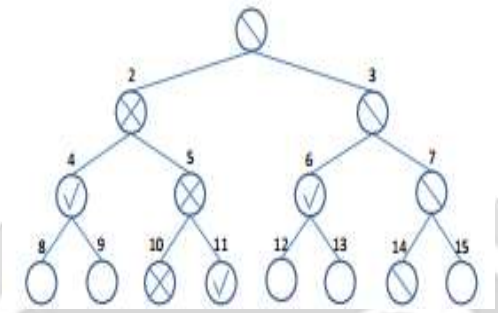

Fig 3. Example of Minimum Cover Set Selection.

2) Cipher-text Decryption

Each user can freely get the cipher-text from the server, but he can able to decrypt the cipher-text only when the attributes he possess at time slot t satisfy the access policy defined in the cipher-text. In this fig 3, it splits the given video into specific time slots and user who holding specific attributes can able to decrypt the appropriate content which is hidden in that video. So, only appropriate user holding necessary key can view the video content.

4. DYNAMIC ATTRIBUTE UPDATING IN TAAC

Phase1: Update List Determination

At the beginning of each time slot t, AA d sets the elements these UL x ,t for any attribute x ∈ U d. Attributes can be easily revoked or re-granted from or to users by using the update list.

Phase 2: Minimum Cover Set Selection

After determining the update list UL x ,t for each attribute x at time slot t, thus Fig[3] finds the minimal set of nodes for which it publishes update keys so that only the users who possess x can decrypt the corresponding attribute.

Phase3: Update Key Generation

In this phase, $AA_{\varphi(x)}$ generates the $\mathsf{UK}_{((x,\,t))}$ according to ST_x and $\mathsf{UL}_{((x,\,t))}$ so that only the users who possess x at time slot t are able to obtain valid $\mathsf{UK}_{((x,\,t))}$.

Time-domain Attribute-based Access Control

Algorithm 1 MinCSS($ST_{x,t}$, $UL_{x,t}$)

$X_e, X_r \leftarrow \emptyset$
$N_{x,t} \leftarrow \emptyset$
$u_e \leftarrow$ the most left empty leaf node
$X_e \leftarrow \text{Path}(u_e)$
for each $u_r \in UL_{x,t}$ **do**
  add $\text{Path}(u_r)$ to X_r
$X_r \leftarrow X_r \setminus X_e$
for each $v_e \in X_e$ **do**
  if v_e is not a leaf node **then**
    $v_{lc} \leftarrow$ left child of v_e
    if $v_{lc} \notin X_r \cup X_e$ **then**
      add v_{lc} to $N_{x,t}$
for each $v_r \in X_r$ **do**
  if v_r is not a leaf node **then**
    $v_{lc} \leftarrow$ left child of v_r
    if $v_{lc} \notin X_r$ **then**
      add v_{lc} to $N_{x,t}$
    $v_{rc} \leftarrow$ right child of v_r
    if $v_{rc} \notin X_r$ **then**
      add v_{rc} to $N_{x,t}$
if $N_{x,t} = \emptyset$ **then**
  add the root node x to $N_{x,t}$
Return $N_{x,t}$

5. SECURITY ANALYSIS AND PERFORMANCE EVALUATION

5.1 Security Analysis

Similar to the preceding multi-authority CP-ABE scheme we will prove TAAC is secure in the generic bilinear group model.

5.2 Performance Evaluation

We generally, analyze the performance of TAAC by comparing the metrics of Storage Overhead, Communication Cost and Computation Cost.

TABLE II
SIZE COMPARISON OF COMPONETS

Scheme	PK	SK	UK	CT
[16]	n_a	$2n_{u,a}\log n_u$	$n_{a,t}n_u \log n_u$	$2n_c(n_c+1)$
TAAC	$3n_a$	$3n_{u,a}\log n_u$	$2n_{a,t}\log n_u$	$5n_c+2$

1) Storage Overhead:

This is mainly occurring in server mainly which comes from cipher-text respectively. The public key and secret key also contribute to this storage overhead.

2) Communication cost:

The communication cost is mainly occurring from the delivery of update keys for each user's. In TAAC, the Minimum Cover Set Selection (MCC) is used for reducing the communication cost.

3) Computation Cost:

In this TAAC, not all the cipher-text need to be encrypted. Only selected cipher-text can be updated in order to simulate the computation time of encryption and decryption.

6. CONCLUSION

In this paper, we have proposed a cryptographic approach, TAAC, to achieve time-domain attribute-based access control for accessing cloud-based video content sharing. Specifically, we have proposed a provably secure time-domain attribute-based encryption scheme by embedding the time into both the cipher-texts and the keys, such that only users who are holding sufficient attributes in a specific time period can decrypt the data. To achieve this, dynamic change of users' attributes, we have also proposed an efficient attribute updating method which enables attribute authorities to grant new attributes, revoke previous attributes and re-grant previously revoked attributes to users at the beginning of each time slot. We have further discussed on how

to share the hidden content with in the video and to achieve access control of video contents that are commonly accessed in multiple time slots.

7. REFERENCES

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology, Tech. Rep., 2009..

[2] X. Wang, M. Chen, T. T. Kwon, L. Yang, and V. Leung, "AMES-Cloud: a framework of adaptive mobile video streaming and efficient social video sharing in the clouds," *IEEE Transactions on Multimedia*, vol. 15, no. 4, pp. 811–820, 2013.

[3] M. Hefeeda, T. ElGamal, K. Calagari, and A. Abdelsadek, "Cloud- based multimedia content protection system," *IEEE Transactions on Multimedia*, vol. 17, no. 3, pp. 420–433, 2015.

[4] H. Shen, L. Zhuo, and Y. Zhao, "An efficient motion reference structure based selective encryption algorithm for h. 264 videos," *IET Information Security*, vol. 8, no. 3, pp. 199–206, 2014.

[5] Z. Shahid and W. Puech, "Visual protection of hevc video by selective encryption of cabac binstrings," *IEEE Transactions on Multimedia*, vol. 16, no. 1, pp. 24–36, 2014.

[6] B. Zeng, S.-K. A. Yeung, S. Zhu, and M. Gabbouj, "Perceptual en- cryption of h. 264 videos: Embedding sign-flips into the integer-based transforms," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 2, pp. 309–320, 2014.

[7] T. Stütz and A. Uhl, "A survey of h. 264 avc/svc encryption," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 3, pp. 325–339, 2012.

[8] PKCS1, "public key cryptography standard no. 1 version 2.2," *RSA Labs*, 2012.

[9] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in *Advances in cryptology*. Springer, 1985, pp. 10–18.

[10] G. Boztok Algin and E. T. Tunali, "Scalable video encryption of h. 264 svc codec," *Journal of Visual Communication and Image Representation*, vol. 22, no. 4, pp. 353–364, 2011.

[11] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based en- cryption for fine-grained access control of encrypted data," in *Proc. of CCS'06*. New York, NY, USA: ACM, 2006, pp. 89–98.

[12] B. Waters, "Cipher-text-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in s*Proc. of PKC'11*. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 53–70.

[13] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proc. of EUROCRYPT'11*. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 568–588.