Multimedia Contents Security by Wireless Authentication*

Jung Jae Kim[1], Kwang Hyoung Lee[2], So Yeon Min[3], Jeong Gyu Jee[4]

[1]Dept. of Computing, Soong-sil Univ., Korea
argniss@empal.com
[2]Dept. of Internet Information, Seoil College Korea
dreamace@seoil.ac.kr
[3]Dept. of Information & Telecom., Seoil College Korea
symin@seoil.ac.kr
[4] Korea Research Foundation, Korea
jgjee@krf.or.kr

Abstract. This paper proposes more various key generation algorithms than conventional encryption method, and more secure encryption method that does not keep each symmetric key of key generation algorithm in server. After implementing proposed system, we verify the system using various sizes of video data. We get the fact that proposed system can reduce the delay time of encryption and decryption at the replay of video data.

1 Introduction

The spread of Internet and interconnection make the context of digital resource distribution changeable and multimedia data such as music, picture, image, and publication need so much. Digital product can be recopied without damage so that we need Digital Rights Management (DRM) technology. External companies such as InterTrust and Microsoft, and internal company such as Digicap provide various DRM solutions [3].

However, conventional DRM solutions use private key method for encryption and this method practices encryption process when user downloads file. Accordingly, this process takes much time. Also, for decryption, all files must be decrypted first in the case of large scale product, so user cannot use file in real-time. There is another problem. If the key of encryption and decryption is exposed to other people, the security of writings may be not guaranteed farther.

* This research was supported by Seoul Future Contents Convergence (SFCC) Cluster established by Seoul Industry-Academy-Research Cooperation Project.

Proposed system proposes the solution that attacker cannot decrypt the entire writings although one symmetric key may be exposed because several symmetric keys are used, and the system encrypts not the whole but the part of movie to improve the speed of encryption and decryption. Also, we need much time to decrypt a large scale of data at the replay of movie so that proposed system uses the scheduling of compensational double buffer. Accordingly, proposed system can provide the decryption and replay service of movie to user in real-time. We implement the proposed system and verify the superiority in the respect of speed of encryption and decryption.

2 Related Work

2.1 Microsoft's DRM system

Microsoft's DRM system is end to end DRM system that distributes securely digital media file to a work provider and consumer [9]. Core control unit is WMRM (Windows Media Rights Manager) and Rights Manager in WMRM delivers a media such as secure music and video into encrypted file on the Internet. Each server or client instances receive a pair of key through the process of individualization, and the instances, which may be cracked or not secure, may be revoked using CRL. CRL is distributed through the web site of Microsoft. The key is included in a license, and license and writings are distributed separately.

However, the time to encrypt after encoding the entire file is very long at the encryption because Microsoft's DRM system can only support its WMV and WMA file format.

2.2 I-Frame DRM system

I-Frame DRM system keeps content ID (CID) and symmetric key value in a server database after selecting AES or SEED algorithm and encrypting I-Frame of movie Group of Picture (GOP) by a symmetric key [1].

When user plays an encrypted movie, sever encrypts the key used in encryption by user public key after achieving user authentication using user certificate. User can get the symmetric key value used in encryption using his private key, decrypt only movie I-Frame, and play the movie after keeping in a buffer with B, P frame. The I-Frame DRM system uses double buffer algorithm to replay the file before decrypting an entire movie. Because I-Frame DRM system encrypts only I-Frame among MPEG data, this system is a partial encryption system and the encryption and decryption speed of this method are faster than other system. Also, this system can support real-time service because this method replays a movie after decrypting one part.

However, this system still spends much time to read all GOP headers because this system computes the size of I-Frame and decrypt it after reading all headers of GOP group to extract I-Frame. Also, because this system uses only one key, encrypted

movie becomes insecure if the key is exposed and the delay time of replay to decrypt the first block occurs at the replay.

3 Proposed System Architecture

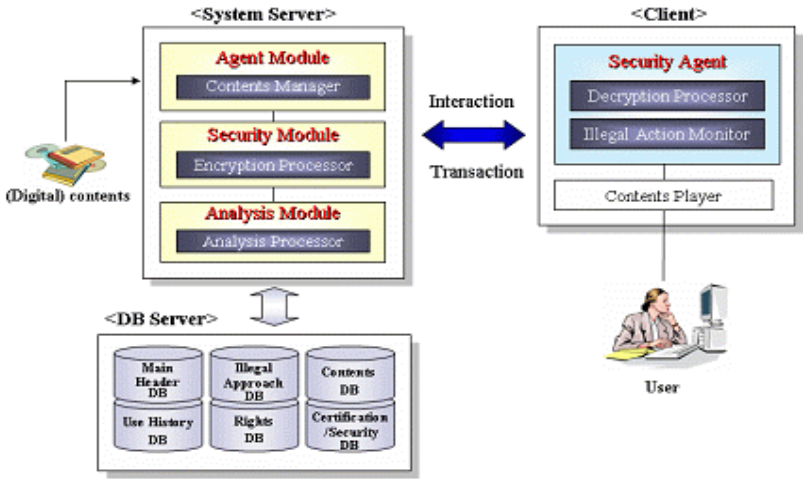

Fig. 1. Proposed DRM system architecture

Fig. 1 shows the proposed system that supports client/server structure. Server consists of agent module, encryption module, analysis module, and database. Client has a security agent that consists of decryption processor and contents player.

3.1 Server encryption module

When CP (Content Provider) receives registered contents from agent module, the security agent of server performs the job that sends it to slice layer which is preprocessing step. Slice layer gets the running time and screen size of content from server, computes the size of image file, which is time interval value(10 seconds), and this time interval is regenerated and get the 50~95% of total size that next block can be decrypted at the same time. We assume that decryption duration may be 100%. Because the former block must be decrypted on the time of regeneration, we consider the usability rate of CPU and this process make it a part of maximum decryption measure. The More CPU is excellent, the more decryption rate increases and vice versa. While next image is played using above method, the system computes repeatedly the size of an image that image file can be decrypted. Then system divides image group into n pieces and saves it. Figure 3 shows slice layer group that is divided into 4 pieces such as G1, G2, G3, and G4.

After slice layer activity, encryption needs. Encryption is applied to not G1 block but next G2 block. When user plays a movie, G1 block can be rightly operated because G1 block is not encrypted and G2 block may be decrypted at this time.

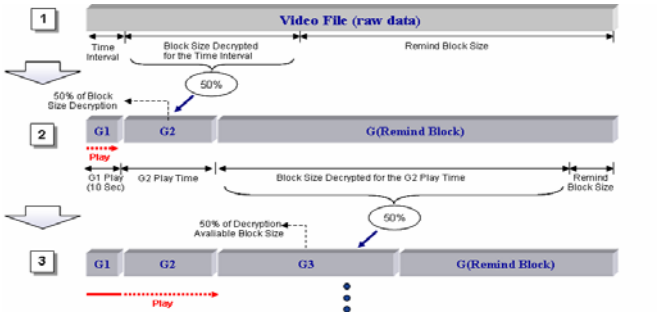

Fig.2. Proposed movie slice layer

Encryption is begun from the slice layer block of G2 and after random number from 5 to 15 is allocated to each slice layer of Gn, a movie may be divided according to this number. If random number is 7, system divides the block in same size and classifies the block with encrypted block and non-encrypted block. Encryption condition is that there is no consecutive non-encrypted block and the rate of encryption block is over 50%. The block of slice layer to encrypt is connected to "1" and block not to encrypt did to "0". The slice layer header of G2 consists of random number (n), block number and block to be encrypted (0101011), and detailed block starting byte (S_b). SH (Slice Header) cannot be opened after re-encryption using CID. Block connected to "1" among part slice layers is encrypted and encryption key is generated like theorem 1.

$$KEY = H(CID \parallel S_b \parallel n \parallel EB) \quad \text{(Theorem.1)}$$

Fig.3 shows that slice layer gets the hash value with header information (SH) and CID, and encrypts this value with slice layer part only connected to "1" using theorem 1 and symmetric encryption method.

Fig. 4. Encrypted slice layer

Hash function (H) is 128 bit MD5 and encryption method is AES. After encrypting mapping block with "1" generated from each slice layer random number, we combine all blocks of slice layer like Fig. 4.

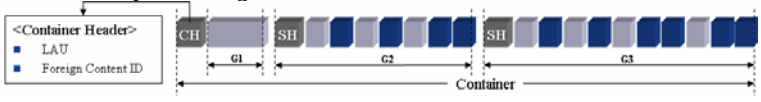

Fig. 5. Container and container header elements for an entire movie

We called encrypted slice layer as container, and Container Header (CH) consists of License Acquisition URL (LAU) and foreign content ID and user can get this contains on web site. LAU contains URL for license and when user plays encrypted content, user verifies the license in LAU using the foreign content ID of content. LAU is to transfer to web page that user can receive license when license do not exist, and container header is not encrypted.

Fig. 6. Whole movie and main header

Client needs to know the byte length of each slice layer to decrypt an image after getting license from LAU, so client must construct Main Header (MH) separately. Fig.5 shows that MH is a file to store the hash value of client DID and record the first byte from G1 to Gn.

3.2 Design of user authentication and CID transmission method

User authentication and control system issue the value of authentication related to instruction to content user.

Fig. 7. Key transmission method

To block information disclosure and verify the user, server provides user authentication number (1) via wireless network after verifying the user, and user inputs the authentication number as a key value (2) and asks the decryption key via wired network (3). After verifying the user authentication number, agent generates decryption key with OTP (One Time Password) and transmits the key to user using secure algorithm. The generated key is divided into 2 keys (CID_Keys_1, CID_Keys_2) using key partition algorithm, CID_Key_1 is hashed by session increase and user authentication value using an agent and the key is transmitted to user (4). User system hashes user authentication number and CID_Keys_1 with random value and transmits them to server (5). Server recognizes the receipt of CID_Keys_1, hashes partition key

CID_Key_2 with user authentication number and random value, and transmits it to user (6). For secure key transmission, this paper proposes key transmission protocol as Fig.6.

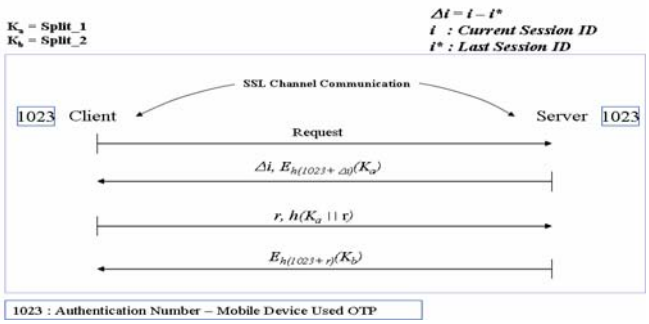

Fig. 8. CID Key transmission protocol

Ka as CID_Keys_1 and Kb as CID_Keys_2 are respectively provided. (1) User authentication number (1023) is generated by server and transmitted securely to user via mobile service of SSL channel. User can request decryption key with an authentication number provided by server. If new user requests decryption key with an authentication number, server generates the key using key partition algorithm and securely transmits it to user using CID key transmission protocol.

Proposed system distinguishes the existing user from new user. The existing user needs not to request duplicate key, because server can verify the session value i that user keeps. i is session value and Δ i is the increase value of session. Also, we define previous session as i* and present session as i. Accordingly, the existing user can verify the increase value of session and use old key. However, because new user does not have the increase value of session, he must receive a key. Fig.7 shows these processes.

3.3 Client security agent

The encryption agent of client must be installed for the decryption of encrypted content, when client user logs on server first time and the agent is downloaded from server. After client that downloads image container practices the container and transmits the hash value of client DID after verifying the license through LAU of CH to server, the agent of server sends the hash value of DID that is included in MH of container and encrypted by user public key to client. The agent of client decrypts the main header file of content based on user certificate.

When user asks system to replay a movie, new user gets a license and decrypts movie using license. User receives a decryption key for user authentication and decryption phase when he gets a key. User requests MH of digital content with content ID, and transmits hashed DID value of user and main header with user public key. Next, user decrypts main header with his private key and checks whether the hash value of main header is identical to the hash value of user computer or not. If the

verification phase is successful, user can replay a movie. MH (Main Header) is encrypted (Epu(MH)) with user public key PU, and is decrypted with user private key. System compares MH decrypted by user private key with DID value of client. If the value of DID is not valid, system stops decryption process and receives new MH. When user gets both MH and CID, user can get the size of G1~Gn in MH, divide the combined container into slice layer, decrypt slice header file (SH) with CID, and generate the key of each slice layer piece with SH. User can decrypt the encrypted block through these decryption processes.

Proposed system can play rightly content without the playing delay if system can decrypt the content header contemporarily at the playing time. The security agent of client extracts slice layer of encrypted image for the decryption, decrypts it with secret key, stores alternately it in buffer A and B, and plays it. Because first slice layer G1 may be not encrypted to acquire initial playing time, user can play G1 rightly and the decryption process of slice layer G2 is achieved contemporarily when user plays G1. In the same manner, the decryption process of slice layer G3 is achieved when G2 is played. System computes the delayed frame during the playing time of all images so that system decides initial buffer size and plays an image. Fig. 10 shows that proposed system uses compensational double buffer system, which consists of two buffers.

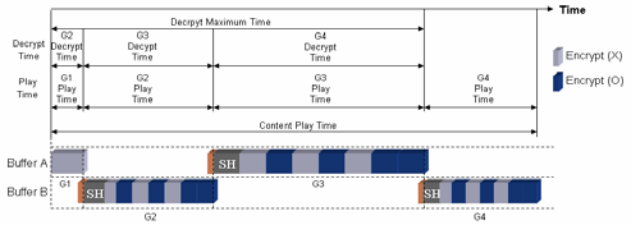

Fig. 10. Compensational double buffer system for replaying a movie

In early stage, system stores slice layer of G1 (Time interval is about 10 seconds) in buffer A to regenerate without a hitch, and decrypts the data of slice layer G2 and keeps in buffer B. When regeneration is over in buffer A, agent transfers this to buffer B to operate the data of buffer B. When data is transferred from buffer A to buffer B, the breaking phenomenon of screen may occur because the first frame of G2, G3, and G4, which is divided by random number, is incomplete. So, we need to attach the last frame value of G1, G2, and G3 to buffer B and make this frame complete.

4 Performance Analysis

4.1 Encryption / decryption analysis

Table 1 shows an encryption method, the possibility of key exposition, and movie encryption and file application as an encryption analysis. We use symmetric key as an encryption method.

Table 1. Encryption method analysis

Section	Conventional DRM system	Proposed DRM system
Encryption method	Singular symmetric key	Plural symmetric key
Key exposition possibility	High	Low
Movie encryption	Entire or section of file	File section

This method can reduce more time to download than method to encrypt content at the download because encrypting content when it is packaged. Also, one user can forward content to another user because proposed system encrypts same content by same symmetric key. Content receiver also super-distribute the content to the third and the third can use the content after getting new license from server. Accordingly, proposed system supports the content redistribution. However if DRM system encrypts content in advance, one content is encrypted by one symmetric key. Accordingly, if user exposes his symmetric key, the content become insecure, we cannot know who exposes the key, and we cannot track an expose. Also, conventional DRM system spends much time to encrypt a large scale movie because encrypting an entire movie.

Proposed system not uses singular symmetric key but plural symmetric keys analogized by decryption agent so that the risk that user exposes his key voluntarily is very low. Although user may expose one key, others cannot know other keys and decrypt the entire movie. Also, only encrypts the smaller amount data of movie because encrypting a part of movie. Accordingly, proposed system can be applied to a large scale movie.

4.2 Decryption analysis

When user plays a movie, agent verifies the license and validity of movie after accessing a server. If user is valid, agent decrypts and plays a movie. All methods of (A), (B), and (C) are same because decryption processes are achieved.

However, because conventional general DRM system (A) plays a movie after decrypting completely it, user must wait to finish the decryption process. So, conventional DRM system cannot support real-time service because a system must spend much time to decrypt a large scale movie. I-Frame DRM system (B) uses double

buffer algorithm but a system cannot immediately play a movie because all frames of movie are encrypted. Proposed DRM system (C) does not need the delay time to play a movie, so this system can support real-time service.

4.3 Performance comparison

This paper compares proposed DRM system with Microsoft's DRM system and I-Frame DRM system for performance analysis. We use movie data that includes 18 different file sizes as data sample. We use Version 1 Key ID in Microsoft's DRM system and AES encryption method that key length is modified from 128 to 256 bits in I-Frame DRM system. MD5 hash algorithm of this paper uses 128 bits key length. Conventional system uses I-Frame as an encryption method but proposed system is unrelated with I-Frame.

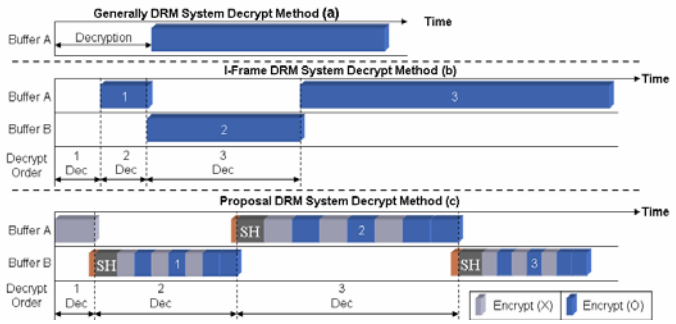

Fig. 12. Decryption method of conventional and proposed system

Proposed system is faster 1.56 times than I-Frame DRM system in respect of running time analysis to encrypt content. Microsoft's DRM system encrypts content after encoding a movie into WMV file format at the encryption. So, we exclude Microsoft's DRM system from running time analysis to encrypt movie.

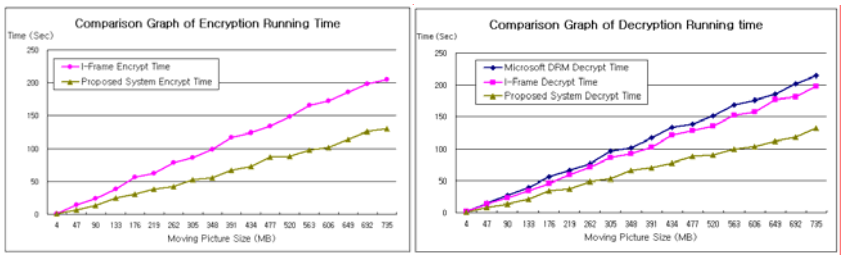

Fig. 13. Comparison graph of encryption/decryption running time

The system that encrypts only I-Frame encrypts not an entire movie but only I-Frame of movie. So, these systems is partial encryption method. However, these

system spends more time than proposed system because it reads all headers of GOP group and computes the size of I-Frame to extract I-Frame.

Proposed system is faster 1.61 times than conventional I-Frame DRM system. Microsoft's DRM system spends most much time to encrypt movie because this system encrypts an entire movie. Decryption running time of I-Frame DRM system is slower than proposed system because this system can get I-Frame after reading all headers of GOP group like an encryption method. Also, supports all movie format, and the running time to encrypt and decrypt is faster than conventional system because proposed system does not encrypt an entire movie.

5 Conclusion

This paper proposed symmetric key encryption system for multimedia data protection using plural random symmetric keys.

Proposed system provides method that sever security module uses several private keys to prevent the illegal user from catching a private key and encrypts it partially so that system can stop the decryption of whole work in advance although the one of these keys may be exposed and attacker cannot play it. As security agent of client needs to spend much time to decrypt a large scale movie, this paper provides compensational buffer control method to play smoothly a movie using streaming method and proposed system performs efficient buffer scheduling. Accordingly, user can decrypt and play a movie in real time, and proposed system can check the breaking phenomenon of screen to play a movie at the buffer scheduling. We can reduce the average 15% of replay delay time that includes decryption time when client replays a large scale movie.

References

1. J. Kim, J. Park, and M. Jun, "DRM system based on public key pool for the security of movie data," Korea Information Processing Society paper, Vol. 12-C, NO. 02, pp 0183-0190, April, 2005.
2. Brad Cox, Superdistribution : Objects As Property on the Electronic Frontier, Addision-Wesley, May 1996.
3. Sung, J Park, "Copyrights Protection Techniques," Proceedings International Digital Content Conference, Seoul Korea, Nov. 28-29, 2000.
4. V.K Gupta, "Technological Measures of Protection," Proceedings of International Conference on WIPO, Seoul Korea, October 25-27, 2000.
5. Whitfield Diffie and Martin Hellman, "New Directions in Cryptography," IEEE Transaction on Information Theory, Vol. IT-22, NO.6, pp.644-654, November 1976.
6. Intertrust : http://www.intertrust.com/main/overview/drm.html
7. Joshua Duhl and Susan Kevorkian, "Understanding DRM system: An IDC White paper," IDC, 200.
8. Joshua Duhl, "Digital Rights Management: A Definition," IDC 2001.
9. Microsoft: http://www.microsoft.com/windows/windowsmedia/drm.asp