Journal of King Saud University – Computer and Information Sciences (2012) **24**, 107–116

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

ORIGINAL ARTICLE

MIKEY for keys management of H.264 scalable video coded layers

Mamoona Naveed Asghar [*], Mohammad Ghanbari

School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, United Kingdom

Received 6 September 2011; revised 26 October 2011; accepted 7 December 2011
Available online 28 December 2011

KEYWORDS

H.264/SVC;
MIKEY;
DRM;
Hierarchical key management;
AES-CTR

Abstract The paper investigates the problem of managing multiple encryption keys generation overhead issues in scalable video coding (H.264/SVC) and proposes a hierarchical top down keys generation and distribution system by using a standard key management protocol MIKEY (Multimedia Internet Keying Protocol). The research goal is two-fold; (1) prevention of information leakage by the selective encryption of network abstraction layer (NAL) units with AES-CTR block cipher algorithm, and (2) reduction of multiple layer encryption keys overhead for scalable video distribution. We combine a MIKEY with the digital rights management (DRM) techniques to derive a mechanism in which every entitled user of each layer has only one encryption key to use, but this key will transparently open the doors of all layers below. The timing results are calculated for the encryption/decryption and the key generation processes relative to encoding/decoding time of test video files, which are noticeably negligible. The scheme is enormously suitable for video distribution to users who have subscribed to various video qualities regarding their desire or constraints on their devices and helps in preventing the loss of revenue of paid services.

1. Introduction

Codecs are developed to compress the video, reduce the storage space and bandwidth requirement of transported streams. The Joints Video Team of the ITU-T VCEG and the ISO/IEC MPEG has standardized a scalable video coding (SVC) which is an extension of the H.264/AVC standard (Wiegand et al., 2003; Ostermann and Bormans, 2004). Scalable video coding (H.264/SVC) (Schwarz et al., 2007) allows the transmission and decoding of partial bit streams to provide video services at various temporal or spatial resolutions or quality, as well as preserving a reconstruction quality that is high compared to the rate of the partial bit streams. So, SVC functionalities provide improvements to transmission and storage applications. SVC has attained remarkable improvements in coding efficiency with an increased degree of supported scalability relative to the scalable profiles of the previous video coding standards.

Cryptography is a conventional technique to provide security to the multimedia content. Most of the research regarding security that has been done in the context of video content is naive and or based on selective encryption. All Cipher algorithms require the data as input but also need a unique value known as 'key' for the operation on plain text as described

[*] Corresponding author. Tel.: +44 07909391129.
E-mail addresses: masghaa@essex.ac.uk (M.N. Asghar), ghan@essex.ac.uk (M. Ghanbari).

Peer review under responsibility of King Saud University.

ELSEVIER Production and hosting by Elsevier

1319-1578 © 2011 King Saud University. Production and hosting by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.jksuci.2011.12.002

by Kerckhoff's principle which states that the rival can know the chosen cipher algorithm but not the key (Cayre et al., 2005). The key generation and distribution is the critically tackled issue to enhance the security of any cipher algorithm. Some recent work on keys generation/distribution for standard and scalable video coding is reviewed here. Recently authors (Wang et al., 2010) pointed out the idea of hierarchical key generation for the cipher algorithm to encrypt the partial H.264/AVC video content. The protection for scalable video coding has been described in (Won et al., 2006; Kim et al., 2007). Li et al. (2009) have devised a NAL level selective encryption technique for H.264/ SVC with stream cipher LEX (Leak Extraction) Algorithm. The LEX uses three keys for the three NAL units individually. The study has pointed out some future work on the key management scheme, which is a key issue in the security of any cipher algorithm. Park and Shin (2009) have designed a hierarchical key management scheme for the selective encryption of scalable video coding. The key management scheme provides the robustness against the known brute-force attack due to the different NAL unit keys.

All the reviewed researches have their own devised key management mechanisms but they don't refer to any standard key management protocol. The authors (Gregory et al., 1995) pointed out that the earlier data communication protocols/ standards have very few security features. Generally, the security is handled at system level that uses the communication protocols, but these days the communication protocols alone cannot handle the proliferating security demands in mobile devices (smart phones, ipads, notebooks and laptops) so there is a need of some key management infusion to enhance the functionality of communication security protocols.

For the scalable layers key generation/distribution, we have used the standard Multimedia Internet Keying Protocol (MIKEY) (Arkko et al., 2004) which is a significant addition to the security of specifically designed multimedia to tackle the key exchange problems in real-time networks. The worth of IETF standard protocol-MIKEY can be examined by its use in commercial applications in the past few years. A number of famous commercial applications like Erricsson (Blom and Cheng, 2009), IPWireless Inc. (Zisimopoulos, 2008), Huawei Technologies Co., Ltd. (Sun and Kong, 2009; Liu, 2007; Doerr et al., 2009) and Siemens (Horn and Kröselberg, 2004; Bücker and Horn, 2008) etc. are also using MIKEY as their key management protocol specifically for the security of multimedia applications.

With the advent of digital media, the digital rights management (DRM) becomes an issue for the digital content manufacturers and publishers. To implement DRM, many methods for digital media have been adopted (Lawrence, 2007). For DRM, cryptographic techniques alone are not enough to provide the flexible content delivery and secure usage. Much work has been done on the joint DRM security techniques, i.e. encryption along with key management (Lin et al., 2005), encryption along with finger printing (Kundur and Karthik, 2004) and encryption along with digital watermarking (Fan et al., 2009; Ju et al., 2003; Thomas et al., 2009). Zou et al. (2007) proposed a selective encryption technique for H.264 video which is adaptive to DRM. The scheme (Zou et al., 2007) claimed to be suitable for the multimedia storage as well as the transmission of digital video.

The proposed key management scheme provides the "Confidentiality Encryption", that is the complete security to the content with full encryption (Hofbauer and Uhl, 2009) which is extremely desirable in confidential defense/military applications. The presented research work incorporates the following DRM security processes.

(1) Authentication key will be derived for the authentications of sender and receiver.
(2) Encryption of data with block cipher algorithm.
(3) Key management with standard protocol.

2. Video codec H.264/SVC

With the steady growth of multimedia streaming over the Internet, the streaming applications demand a variable bandwidth over best-effort Internet. In video streaming application, the servers have to support a large number of users with different screen resolutions and network bandwidth. If the screen resolution of a user is too small and the bandwidth between the user and server is narrow to support high frame rate, resolution and quality then the need of low quality video arises. Scalable video coding (SVC) has served the server to fulfill the objective of variable frame rate, resolution and fidelity.

Scalable video coding (H.264/SVC) is the emerging model and has quickly come up to satisfy the needs of multimedia services. SVC technology permits devices to send and receive multi-layered bit streams; it allows the transmission and decoding of partial bit streams to provide video services with different frame rates, spatial resolutions (picture size) and quality on base enhancement layers. A top level view of SVC model is shown in Fig. 1 which shows the possibility of partially decodable video stream according to the requirement of the receiving device. The base layer has the minimum data which can better serve the small devices alone; the enhancement layers have more data in hierarchy from bottom to top. The top enhancement layer has maximum frame rate, picture resolution and quality to serve the need of high resolution devices (HD TV).

H.264/SVC uses two entropy coding modes, variable length coding (VLC) and binary arithmetic coding (BAC), both of these designs are used in a context adaptive (CA) way, known as CAVLC and CABAC. Syntax elements at and below the slice layer can be adaptively coded. The CAVLC requires less complexity than CABAC during encoding of H.264/SVC layers while CABAC provides an average of about 10% more compression at the cost of more complexity than CAVLC.

The SVC layers are identified by three identifiers (IDs) which are in NAL header in Fig. 2; the temporal ID (T), dependency ID (D) and quality (i.e. SNR) ID (Q), which are written as a triplet (D, T, Q). For example, the base layer NAL unit of the lowest temporal resolution and SNR scalability is identified as $(0, 0, 0)$. It is possible to combine the three concepts of temporal, spatial and SNR scalabilities together. The upper layers of scalable video are predicted on the foundation of lower layers.

H.264 has a two layer architecture; video coding layer (VCL) and network abstraction layer (NAL); VCL is used for video compression and NAL is used for VCL and the aux-

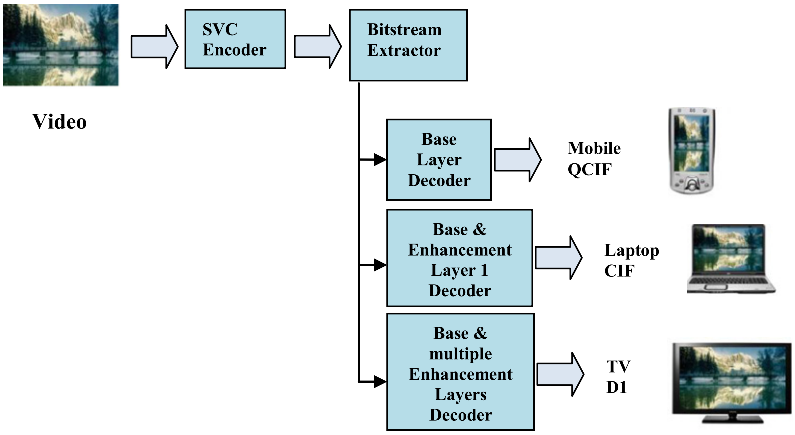

Figure 1 SVC encoder functionality.

4 bytes NAL unit Header

1	2	5	1	1	6	1	3	4	3	1	1	1	2	
F	NRI	Type	R	I	PRID	N	DID	QID	TID	U	D	O	RR	NALu Payload

Figure 2 SVC NAL unit.

iliary data (non-VCL) transportation over network. The NAL unit defines a generic format for sending video data over both packet oriented (e.g., Internet protocol/RTP systems) and bit stream oriented transport systems (e.g., H.320 and MPEG-2/H.222.0 systems). VCL NAL units contain data that represent video pictures in the form of a slice or data partition, while the non-VCL NAL units contain other additional information such as parameter sets, timing information and other supplemental data. For SVC, each NAL unit comprises of a four byte header and variable number of bytes payload holding coded symbols (Wang and Schierl, 2010). A set of NAL units has the complete decoded picture which is called access unit. SVC NAL unit is shown in Fig. 2.

Regarding the conventional technique of data security, we require encrypting the scalable layers altogether. But if that happens, the users subscribed for the different layer video may not be able to receive the desired layer data separately, hence destroying the purpose of scalable video coding. To get the benefit of multiple scalable layers, we need to encrypt them separately as per layer basis. This scheme helps the users to get the entitled data according to their bandwidth, storage and rendering devices' capabilities.

3. Cryptography

Cryptography is the mathematical science of converting readable data (plaintext) into secret codes (ciphertext). It is a primeval art to scramble or replace the known codes with the un-known codes, which began with the historical era with the advent of writing. Cryptographic methods are implemented by the use of cryptosystems. These systems comprise of two parts (Wohlmacher, 1998):

(1) Algorithms with a series of steps having a set of functions with parameters and
(2) The set of keys.

Cryptographic algorithms are never a secret; their steps are always open to everyone. The object which should be a secret and needs to be hidden from public and unauthorized access is the KEY, used by cryptographic algorithms. With the key classification, there are two types of cipher algorithms, symmetric ciphers (shared key) and asymmetric ciphers (two different keys). The symmetric key technique further categorizes itself into block ciphers and stream ciphers. The block ciphers encrypt the plaintext block by block of pre-defined size, i.e. the block of 32bytes, 64 bytes or 128 bytes. These algorithms are efficient in terms of speed and after encryption the message size is not changed while the stream ciphers use the function to encrypt the plaintext bit by bit in a stream almost like reading a file character by character and encrypting it. The message length gets changed after encryption by the stream ciphers (Kuchar, 2000). The strength of cipher algorithms are related to the key size. Even the algorithms are not so secure but with larger key size and excellent key management scheme, their performance can be enhanced (Sidek et al., 2007).

Many cryptographic algorithms have been proposed and implemented for the preceding half century. All of the algorithms have individual characteristics and usage.

3.1. Advanced encryption standard (AES)

The Advanced Encryption Standard (AES) (Schaad and Housley, 2002), also known as Rijndael, is a standardized cipher algorithm by the US government. It is a symmetric key block cipher (128-bit block size) based on modified substitution-permutation network. After the rigorous analysis of AES, it is being widely used all over the world. AES can use keys of three lengths, 128 bits, 192 bits and 256 bits. With the increase in key size the encryption round steps also increase as the total round steps with 128 bit key are 10, with 192 bit

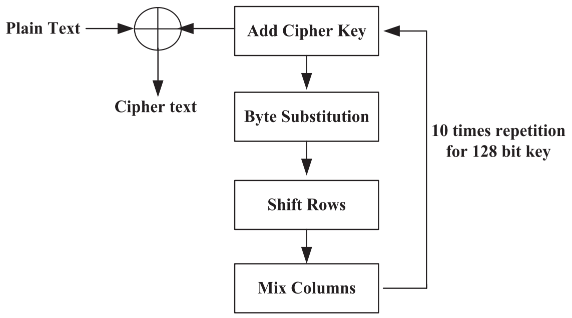

Figure 3 AES operation.

key are 12 and 256 bit key are 14. For both cipher and decipher, the AES algorithm uses a round function that is comprised of four different byte-oriented transformations shown in Fig. 3:

The reason to choose the AES in counter mode (CTR) is that, it is a block cipher, which takes the data block of 128 bits at a time for encryption; it makes the encryption process efficient for the large data rather than using any stream cipher.

4. Multimedia Internet Keying Protocol (MIKEY)

Multimedia Internet Keying Protocol (MIKEY) is an important addition to the security of multimedia specifically designed to tackle the key exchange problem especially in real-time networks. It is used for the key management of one-to-one, one-to-many and many to many small-size group communications. The key management protocol is devised to enable the end-to-end security, i.e. only the participants involved in the communication have authorized access to the generated key(s) and hence to the content. The key generation is simple and efficient.

MIKEY uses a total of Eight (08) keys. The keys will be generated on either the sender side or both sides (sender and receiver).

(1) TGK (traffic generation key).
(2) TEK (traffic encryption key).
(3) Encryption keys (total 02, one each for sender and receiver).
(4) Authentication keys (total 02, one each for sender and receiver).
(5) Salting keys (total 02, one each for sender and receiver).

MIKEY has supported five methods for transporting/establishing a TGK or to setup a common secret, for the all communication scenarios by using:

(a) Pre-shared key.
(b) Public-key encryption.
(c) Diffie–Hellman (DH) key exchange.
(d) DH-HMAC (HMAC-authenticated Diffie–Hellman).
(e) RSA-R (reverse RSA).

In all the above mentioned methods, the DH is more desirable as it provides the Perfect Forward Secrecy (PFS). This research has implemented Diffie–Hellman (Diffie and Hellman, 1976) for key establishment integrated with a keyed hash message authentication code (HMAC) (Euchner, 2006) for attaining combined authentication and message integrity of the key management messages exchange.

5. Proposed key management scheme

The paper devises a key management (generation/distribution) scheme to enhance the security of scalable video coding layers on the application level. The security to the scalable data means to provide the encryption on all layers of data from L_0 (base layer) to L_n (top enhancement layer). Let us assume user U_i is subscribed to receive the data of layer L_i, so he must have access to all the lower layer encryption keys, i.e. eK_0 to eK_i to be able to decode the subscribed layer data because the encryption is applied from layer L_0 to L_i as shown in Fig. 4.

The management of all sets of layer L_i keys (shown in Table 1) for user U_i is a huge security hazard. The handling of multiple keys for user L_i data is complicated especially when the salable data have a large number of layers. Many problems arise with the large number of keys generation especially computational cost of generating the multiple keys at once to get the L_0 to L_i layer data, memory consumption and time to save the eK_0 to eK_i keys which are sizeable as per security needs. So, the goal is to derive a mechanism in which each user needs to hold single encryption key to retrieve his subscribed layer data. Thus a number of keys will not travel over the network, hence reducing security hazard.

MIKEY generates the two major keys (TGK and TEK) which will further generate the lower keys in a hierarchical fashion. The MIKEY keys are:

i *Traffic generation key (TGK):* It is the master key and mutually generated and distributed by the sender & receiver using Diffie–Hellman (DH) key exchange agreement method after every month.
ii *Traffic encryption key (TEK):* It is generated by the TGK on the sender side and transported to the receiver after encryption by AES with TGK by using HMAC with TGK and MIKEY's specified TEK constant value

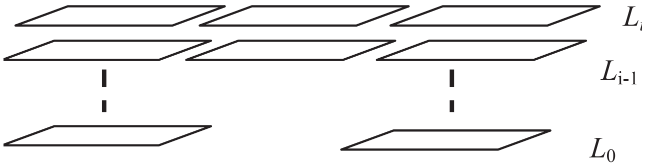

Figure 4 Scalable layers.

Table 1 Set of encryption keys should be held for each layer.

Layers	Encryption keys held for each layer
L_i	eK_0, eK_1, eK_2, eK_3, . . . , eK_{i-1}, eK$_\mathrm{i}$
L_{i-1}	eK_0, eK_1, eK_2, eK_3, . . . , eK_{i-1}
. . .	. . .
L_2	eK_0, eK_1, eK_2
L_1	eK_0, eK_1
L_0	eK_0

Table 2 Characteristics of MIKEY keys.

Keys	Key length (bits)	Generation/distribution methods and parameters	MIKEY constants	Key life time
TGK (master key)	128	Diffie–Hellman	DH prime and base values	01 month
TEK (traffic encryption key)	128	HMAC-SHA1(TGK)	*0x2AD01C64*	Daily for 12 h
Master encryption key (*eK*)	128	HMAC-SHA1(TEK)	*0x15798CEF*	For session
Authentication key (*aK*)	160	HMAC-SHA1(TEK)	*0x1B5C7973*	Unique for every user
Salt keys (*sK*)	112	HMAC-SHA1(TEK)	*0x39A2C14B*	Daily for 12 h

(see Table 2) and random number RAND. TEK will be generated on daily basis. It will be used for 12 h during the day.

iii *Master encryption key (eK):* The TEK will generate the top layer encryption key (Master) by the use of HMAC and the MIKEY's specified Encryption key constant value (see Table 2) on both sender and receiver side. It will be used for the encryption of SVC top layer contents.

iv *Authentication key (aK):* The TEK will generate the authentication key by the use of HMAC and the MIKEY's specified authentication key constant value (see Table 2) on both sender and receiver side. It will be used for the authentication purpose of parties.

v *Salt key (sK):* The TEK will also generate the salt key by the use of HMAC and the MIKEY's specified salt key constant value (see Table 2) on both sender and receiver side. It is used to alter some bytes of TEK to enhance the security. The salt keys are generated to alter some bytes of TEK to enhance the security on daily basis. The few bytes of salt key are replaced in the TEK and after 12 h use of TEK, the salted TEK will be used for the next 12 h.

The general equations for overall keys generation scheme are:

$$\text{TGK} \rightarrow g^{\text{sr}} \bmod p (\text{Diffie–Hellman}) \tag{1}$$

where p = prime no., g = generator, sr = sender and receiver RAND values

$$TEK \rightarrow \text{HMAC (TGK, MIKEY constant} \\ \|\text{RAND, TEK length)} \tag{2}$$

$$\text{Master}eK \rightarrow \text{HMAC (TEK, MIKEY } eK \text{ constant} \\ \|\text{RAND, } eK \text{ length)} \tag{3}$$

$$aK \rightarrow \text{HMAC (TEK, MIKEY } aK\text{constant} \\ \|\text{RAND, } aK \text{ length)} \tag{4}$$

$$sK \rightarrow \text{HMAC (TEK, MIKEY } sK\text{constant} \\ \|\text{RAND,}sK \text{ length)} \tag{5}$$

The TEK further generates the encryption key, authentication key and salt keys. Table 2 shows the characteristics of all MIKEY keys (key length, life time and constants) with their generation/distribution summaries.

i *SVClower layers encryption keys:* The master encryption key further generates the lower layer keys to encrypt the content of SVC lower layers by the use of self defined constants for each layer as mentioned by MIKEY specific constants. The keys are generated recursively in hierarchical fashion, i.e. top enhancement SVC layer *Ln* encryption key eK_n will generate its immediate lower L_{n-1} key eK_{n-1} by using its own value, the eK_{n-1} will generate eK_{n-2} key and so on, on the receiver side. The general equations for generation of encryption keys for lower SVC layers are:

$$eK_n \rightarrow \text{HMAC (TEK, } eK_n \text{ constant}\|\text{RAND, } eK_n\text{length)} \tag{6}$$

$$eK_{n-1} \rightarrow \text{HMAC (}eK_n, eK_{n-1} \text{ constant} \\ \|\text{RAND, } eK_{n-1} \text{ length)} \tag{7}$$

$$eK_{n-2} \rightarrow \text{HMAC (}eK_{n-1}, eK_{n-2} \text{ constant} \\ \|\text{RAND,}eK_{n-2} \text{ length)} \tag{8}$$

RAND is generated according to the PRF (a keyed pseudo-random function) in (Arkko et al., 2004). The overall key generation scheme is shown in Fig. 5.

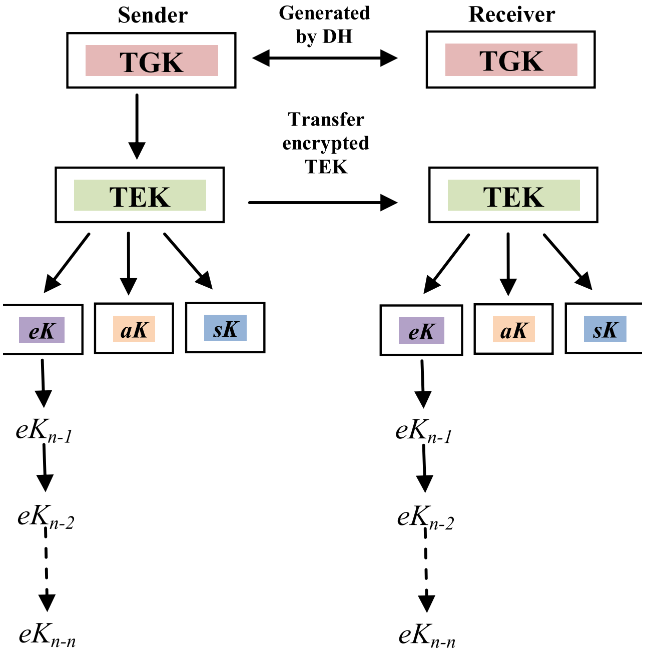

Figure 5 Key generation mechanism.

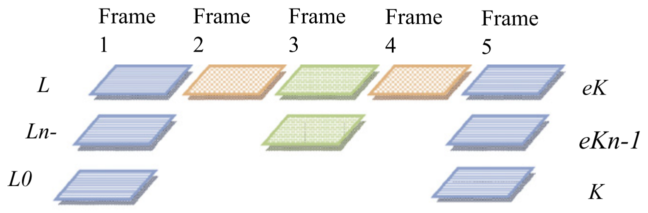

Figure 6 Keys per scalable layer.

After the keys generation and distribution, the proposed solution will work for the encryption of layers by using AES-CM cipher algorithm. The idea behind the encryption of scalable layers can be easily understood from Fig. 6.

Three ascending order scalable layers are shown in Fig. 6, lowest is the base layer and the upper two are enhancement layers 1 and 2. According to Fig. 6 the frames 1 and 5 (horizontal line pattern) are on the base layer, they will be encrypted by the key eK_0. The three frames are on the immediate upper layer of the base layer 1, 3 and 5. Frames 1 and 5 are already encrypted by the base layer key eK_0 so; only frame 3 (bricks pattern) belongs to layer L_{n-1} which will be encrypted by key eK_{n-1}. This process of encryption is continued on all the above layers and the frames which are already encrypted on the lower layers will not be re-encrypted on the upper layers. Only the respective layer frame(s) will be encrypted with the corresponding layer encryption key. The general equations for the bit streams encryption on all layers:

$$eK_n \text{ (encrypts)} \rightarrow L_n \text{ frames} - L_{n-1} \text{ Frames} \tag{9}$$

$$eK_{n-1} \rightarrow L_{n-1} \text{ frames} - L_{n-2} \text{ frames} \tag{10}$$

6. Experimental results

The performance of the proposed key management scheme has been tested with the SVC reference software (Joint Scalable Video Model) JSVM 9.19.8 version encoder. The experiments run on a machine Intel Core i3-330M (2.13GHz) processor with 4GB RAM.

For the evaluation of results; the experiments are performed over four different CIF and two 4CIF test video sequences, which were encoded into four layers with five temporal, two spatial and two SNR resolutions (levels). All four layers are encoded with variable bit-rate (VBR), GOP size is 16 frames and with the same Intra period, for both CAVLC and CABAC entropy coding modes. The proposed encryption system is applied on NAL units per scalable video layer. AES-CTR mode is used to encrypt the NAL units in independent blocks of individual layer with 128 bit encryption key and encrypts the whole NAL unit payload and leaves the initial four bytes of NAL header un-encrypted. The NAL header (Fig. 2) has some important information for the network friendly behavior as it shows the priority of NAL unit and some other information regarding the video quality of services over the network so the NAL header should be open while transferring over network. The other reason of NAL level security is that the NAL units can be routed and decrypted independently instead of collecting all of them together till the whole encrypted file is created and then the decryption will takes place, so the proposed scheme avoids the delay at the receiver end. Thus we have more control over the transmission of packets in case of independent NAL unit encryption. The evaluation results encompass the encryption/decryption and key generation timings with different number of encoded frames using CAVLC entropy coding. The encoding and decoding times varied for CABAC while the encryption/decryption times are same as with CAVLC entropy coder.

The encryption/decryption time is calculated on the whole encoded bit-stream; Table 3 shows the encoding, encryption, decryption and decoding times (in seconds) for two 4CIF Video sequences with different input frame rates. Encoding times are taken in integer values for the sake of simplicity; while encryption, decryption and decoding times are taken up to three decimal places for the clear overhead estimation of cryptography over video file. As Table 3 explains the time taken for encryption relative to encoding and decoding of video file is much smaller. Therefore, with negligible additional computational cost, we are able to achieve security and selective distribution of bit streams.

The experiments are also performed on four different CIF video sequences and the results are shown in the graph (Fig. 7), which clearly show the negligibility of encryption and decryption timings of four CIF test videos.

The graph (Fig. 8) depicts the time (microseconds) required for generating the keys. For each subscriber, three keys TEK, *aK* and a master key *eK* have to be derived. In addition, depending upon the subscriber whether he has demanded any subscribed layer key, the master encryption key is generated by the system for the subscribed layer, and given to him then he derives his own encryption keys for all lower layers. It is a hierarchical system and each layer encryption key *eK* is derived from its former layer *eK*.

Table 3 Timings (in seconds) of sample 4CIF video.

Sample 4CIF timings	30 Frames	60 Frames	90 Frames	120 Frames	150 Frames
ICE (4CIF)					
Encoding time	85	169	270	330	420
Encryption time	0.018	0.023	0.034	0.041	0.045
Decryption time	0.024	0.031	0.040	0.048	0.055
Decoding time	6.634	13.115	19.963	25.669	32.022
SOCCER (4CIF)					
Encoding time	78	162	255	342	455
Encryption time	0.017	0.022	0.032	0.042	0.047
Decryption time	0.024	0.032	0.39	0.049	0.057
Decoding time	6.983	13.672	21.796	28.151	34.021

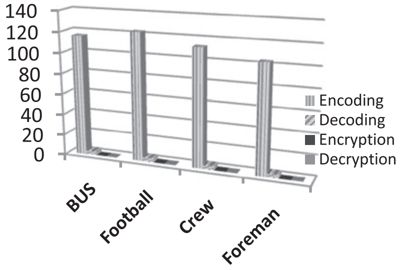

Figure 7 Timings (in microseconds) for CIF sequences encoded with 150 frames.

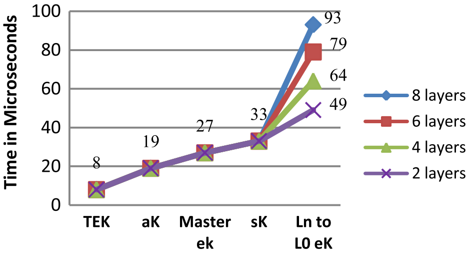

Figure 8 Keys generation timings.

The timings given in the graph are of keys for scalable layers L_8, L_6, L_4 and L_2, but for generating hierarchical encryption keys these have been derived from layer L_8 to L_0. The experiments show that the timings of generating TEK, *aK* and Master *eK* are the same whether they are being generated for layer L_2 or layer L_8. The difference is shown in key generation timings of layers L_n to L_0 *eK* which are derived from the master *eK*. If the hierarchical encryption keys will be generated for just two scalable layers (the base and enhancement layer), it will take 49 microseconds and if they are generated for eight layers (one base and seven enhancement layers) then it will take 93 microseconds. The timings of generating hierarchical encryption keys depend on the number of SVC layers from top to bottom and the layer encryption keys generation timings are included in the encryption and decryption timings which are shown in Table 3 for 4CIF and in Fig. 7 for CIF video sequences.

The graph clearly depicts that the keys generation time is fairly negligible, so to make the system robustly secure, the keys can be generated very often without any additional overhead on encryption/decryption computational cost of the proposed system.

7. Performance analysis

To evaluate the performance of the proposed DRM system in detail, we have carried out the following analysis.

7.1. Security analysis

7.1.1. Brute force attack on encrypted data and keys

The brute force attack is a strategy to find the correct value by continuously trying every possible value for data/key bits in turn, until the correct value is identified. Our proposed system is secure enough against the brute force attack on encrypted video data by considering the following countermeasures.

(1) The encryption process is much sensitive to key; the encrypted data statistics are varying with even a single key bit change, because the encryption key is XORed with the video data in every AES encryption round. As the TEK is changing after every 12 h, next time even the same data will be encrypted with a different key. This frequent key changing makes the possibility of guessing data bits impossible.

(2) The encrypted NAL units can eavesdrop during transmission. Assuming a CIF frame of 352 × 288, the number of macroblocks would be $(352 \times 288)/(8 * 8) = 1584$, thus the possibilities of attacking are 2^{1584}. This is an exceptionally large figure and cannot be handled in a reasonable finite time. Note that if the attacker will not decipher it methodically, he will use brute force and will simply try various bits. But even then, the number is so large that our video data are fairly safe against brute force attacks.

The AES-CTR with 128 bit key is used in our proposed work. The sample space in this case for brute force attack on keys is 2^{128} which is again a large figure and current day computational speeds of the computers cannot do it in a reasonably finite time. The time required to break the 128 bit key by applying all possible keys at 50 billion keys/sec is 5×10^{21} years (Esslinger, 2010).

7.1.2. Distribution of attacks

Poisson probability distribution occurs when the probability *p* of the event is small, while the number of possibilities *n* is large and the *np* is of moderate size. It is also known as probability of rare events, like customers visiting a bank, number of telephone calls received on a switchboard per minute. In our case, the number of possibilities is all the hackers launching an attack, while, probability that an attack will be launched is small. CISCO security statistics (Tesch and Abelar, 2006) show that, an attack on a host machine occurs every five minutes, translating to about 300 attacks per day. We assume that 20% of these attacks are on video. The Poisson distribution is given by:

$$P(X) = \frac{e^{-\mu}\mu^{x}}{x}$$

where e is constant $= 2.718$, μ is the average number of attacks in a given time interval. x is the number of attacks for which probability is desired.

Each attack that is launched is not necessarily successful. Let us say the attacks will need at least 12 h to succeed. Therefore we have decided to change the key after every 12 h. Another point to be noted is that even if an attack succeeds, it will be successful for a very short duration, because in the next 12 h, a new key will be used and previous successful attack will be rendered useless.

7.2. Computational overhead

There are multiple parameters that create additional overhead over the system to achieve the security. In the proposed system, the computational overhead is not only estimated with respect to key generation and encryption/decryption timings

Table 4 Comparative analysis of proposed scheme.

Proposed Schemes	Parameters used for encryption	Applicable on entropy coding	Encryption algorithms	Incorporated key management scheme	Multiple keys overhead solution	Key management protocol
Layered encryption for SVC (Li et al., 2009)	NAL units, IDR frames, PPS, SPS	CABAC/ CAVLC	LEX stream cipher	Yes	No	Not specified
Access control in SVC (Won et al., 2006)	Signs of texture, MVD and FGS	CABAC	XOR Stream Cipher	Yes	No	Not specified
Key management scheme for SVC (Park and Shin, 2009)	Intra prediction modes, signs of residual and MVD	CABAC/ CAVLC	Stream cipher	Yes	No	Not specified
Our scheme	NAL units payload	CABAC/ CAVLC	AES-CTR	Yes	Yes	MIKEY

but there is also an important subject to be noticed, which is the increase of file size after encryption when we are using a block cipher. The proposed system applies the NAL level encryption per layer rather than the whole bit-stream per layer, to maintain the identity of NAL unit during transmission which is useful to avoid the decryption delay at the receiver end. The block cipher AES takes the NAL unit payload blocks which are multiples of 16 bytes and encrypts them. But if the data block is less than 16 bytes, there is additional bit stuffing, increasing the file size. Of course, it is removed on the receiver side during decryption. It is observed during experiments that the video files with smaller resolution (QCIF) require more bit stuffing because of less payload data in NAL units, so their file size increases further from 1% to a maximum of 4%. But the influence on CIF and 4CIF which have higher resolutions is minimal as it results in 0.5% to maximum 2% file size raise.

7.3. Error tolerance

The cipher algorithms have the diffusion property, so the single bit error can cause many erroneous bits after decryption. The cipher chaining modes are not suitable for encrypting data because they are not error tolerant.

For the sake of error tolerance the AES in counter mode (CTR) is used to encrypt the NAL unit payload blocks independently on individual SVC layer. The study (Massoudi and Lefebvre, 2008) believes that the block ciphers with independent block encryption provide a good stability between security and error tolerance. It is obvious by the reviewed study that if AES-CTR is an error tolerant cipher algorithm then the scalable layers may not be much affected by the errors after decryption.

7.4. Comparative analysis

The literature shows a few studies on key management of scalable layers. For comparative analysis of our proposed DRM scheme with the existing work, we choose three, already proposed encryptions with key management methods for H.264/SVC. The chosen proposed techniques are compared on the basis of the following parameters shown in Table 4.

Li and Won, both have proposed the similar kind of key management schemes for SVC. The keys are generated for individual scalable NAL units. It means if a layer has three different NAL units according to scalability features like temporal, spatial and SNR then three different keys will be derived and distributed to decode the individual layer. Park has devised a key management scheme by creating multiple keys, i.e. layer and NAL unit keys separately. This key management scheme provides the robustness against the known brute-force attacks due to the different NAL unit keys.

To summarize, all the compared techniques are using stream ciphers which are computationally expensive than a block cipher and the security of stream ciphers is also questionable against attacks. All the discussed proposed schemes are almost the same on producing NAL level keys with complicated key generation algorithms, hence could not solve the problem of multiple keys overhead for each layer and still there is a need of some standardized key management protocol to prove the strength of proposed key management mechanism.

Our proposed DRM scheme is computationally fast with minimal overhead and specifically designed to tackle the multiple keys overhead issue of scalable layers.

8. Conclusions and future work

The paper presents the complete DRM solution for H.264 scalable layers by focusing on many critically important issues, i.e. timing results, bit-rate overhead, security analysis and error tolerance. After the detailed analysis of key management protocol, the strength of cipher algorithm for layer wise encryption, it is expected that the proposed DRM system will be a desirable contribution for the security of scalable video coding. The timing results for SVC stream encryption/decryption and hierarchical keys generation proves the efficiency of the proposed scheme. The keys generation timings are quite negligible; even the very often hierarchical keys generation does not affect the system performance and robustness. The proposed DRM system including key management scheme is fully functional with CAVLC and CABAC encoded bit-streams. The significance of the proposed method is that, the subscriber of each layer has only one encryption key to use, but this key can open the doors of all the layers below. It is suitable for video distribution to users who have subscribed to a different video regarding bandwidth and different quality streams.

The same hierarchical key management scheme can be implemented on partial encryption of scalable layers (Won et al., 2006; Park and Shin, 2009) without any modification. Although research shows that the AES-CTR mode provides a good support for error tolerance, but error recovery issues can be investigated more in transmission scenarios of scalable layers as future work.

References

Arkko, J., Carrara, E., Lindholm, F., Naslund, M., Norrman, K., 2004. MIKEY: Multimedia Internet KEYing, RFC 3830. Internet Engineering Task Force. <http://www.ietf.org/rfc/rfc3830.txt>.

Blom, R., Cheng, Y., 2009. Key Management for Secure Communication, Wo Patent Wo/2009/070,075.

Bücker, W., Horn, G., 2008. Method for Providing a Symmetric Key for Protecting a Key Management Protocol, Wo Patent Wo/2008/037,670.

Fan, C.-I., Chen, M.-T., Sun, W.-Z., 2009. Buyer–seller watermarking protocols with off-line trusted third parties. Int. J. Ad Hoc Ubiquitous Comput. 4 (1), 36–43.

Cayre, F., Fontaine, C., Furon, T., 2005. Watermarking security: theory and practice. IEEE Trans. Signal Process. 53, 3976–3987.

Doerr, M.B., Nysen, P.J., et al., 2009. Mobile Television Broadcast System, Wo Patent Wo/2009/006,593.

Esslinger, B., 2010. The CrypTool Script: Cryptography, Mathematics and More, Background reading for CrypTool the free e-learning program (with number theory code samples for Sage) (distributed with CrypTool version 1.4.30), 10th ed., Frankfurt am Main.

Gregory, B. White, Gregory, W. White, Eric, A. Fisch, Udo, W. Pooch, 1995. Computer system and network security. CRC Press Comput. Eng.

Ju, H.S., Kim, H.J., Lee, D.H., Lim, J.I., 2003. An anonymous buyerseller watermarking protocol with anonymity control. in Information Security Cryptology (ICISC), Seoul, Korea 2587/2003, 421–432.

Hofbauer, H., Uhl, A., 2009. Selective Encryption of the MC EZBC Bitstream for DRM Scenarios, MM&Sec '09 Proceedings of the 11th ACM Workshop on Multimedia and Security.

Horn, G., Kröselberg, D., 2004. Method for Creating and Distributing Cryptographic Keys in a Mobile Radio System, and Corresponding Mobile Radio System, Wo Patent Wo/2004/075,584.(Siemens).

Kim, Y., Jin, S.H., Bae, T.M., Ro, Y.M., 2007. A selective video encryption for the region of interest in scalable video coding. In: IEEE Region 10 Conference, pp. 1–4.

Kuchar, M., 2000. Dispelling the myths of cryptography. Database and Network Journal 30 (2), 3.

Kundur, D., Karthik, K., 2004. Video fingerprinting and encryption principles for digital rights management. Proc. IEEE 92 (6), 918–932.

Lawrence, Harte, 2007. Introduction to Digital Rights Management, ISBN: 1932813403.

Li, C., Zhou, X., Zhong, Y., 2009. "Layered encryption for scalable video coding", image and signal processing, CISP '09. IEEE, 1–4.

Lin, E., Eskicioglu, A., Langendijk, L., Delp, E., 2005. Advances in digital video content protection. Proc. IEEE 93 (1), 171–183.

Liu, Y., 2007. Method, system and device for realizing multi-party communication security, US Patent App. 20,090/271,612.

Massoudi, A., Lefebvre, F., et al., 2008. Overview on selective encryption of image and video: challenges and perspectives. EURASIP J. Inf. Secur. 2008, 23–34, Hindawi Publishing Corp., New York, NY, United States.

Euchner, M., 2006. HMAC-Authenticated Diffie–Hellman for Multimedia Internet KEYing (MIKEY), RFC 4650. <http://www.ietf.org/rfc/rfc4650.txt>.

Ostermann, J., Bormans, J., 2004. Video coding with H.264/AVC. IEEE Circuits Syst. Mag.

Park, S., Shin, S., 2009. An efficient encryption and key management scheme for layered access control of H.264/scalable video coding. IEICE Trans. Inf. Syst. 92 (5), 851–858.

Schaad, J., Housley, R., 2002. Advanced Encryption Standard (AES) Key Wrap Algorithm, RFC 3394.

Schwarz, H., Marpe, D., Wiegand, T., 2007. Overview of the scalable video coding extension of the H.264/AVC standard. IEEE Trans. Circuits Syst. Video Technol. 17 (9), 1103–1120.

Sidek, M., Rahim, A., Sha'ameri, Z.A., 2007. Comparison analysis of stream cipher algorithms for digital communication. Jurnal Teknologi 46D, 1–16.

Sun, K., Kong, T., 2009. Method and System for Distributing Key of Media Stream, US Patent App. 20,090/279,705.

Tesch, D., Abelar, G., 2006. Security Threat Mitigation and Response: Understanding Cisco Security MARS. Cisco Press ©2006 ISBN: 1587052601.

Thomas, T., Emmanuel, S., Subramanyam, A.V., Kankanhalli, M., 2009. Joint watermarking scheme for multiparty multilevel DRM architecture. IEEE Trans. Inf. Forensics Secur. 4 (4), 758–767.

Diffie, W., Hellman, M., 1976. New directions in cryptography. IEEE Trans. Inf. Theory 22 (6), 644–654.

Wang, X., Zheng, N., Tian, L., 2010. Hash key-based video encryption scheme for H.264/AVC. Signal Process. Image Commun. (25/6), 427–437.

Wang, Y.K., Schierl, T., 2010. RTP Payload Format for SVC Video. draft-ietf-avt-rtp-svc-21.txt.

Wiegand, T., Sullivan, G., Sullivan, J., Bjøntegaard, Gisle, Luthra, Ajay, 2003. Overview of the H.264/AVC video coding standard. IEEE Trans. Circuits Syst. Video Techn. 13 (7), 560–576.

Wohlmacher, P., 1998. Requirements and Mechanisms of IT-Security Including Aspects of Multimedia Security, Citeseer. Workshop at ACM Multimedia '98 Bristol, U.K.

Won, Y.G., Bae, T.M., Ro, Y.M., 2006. Scalable protection and access control in full scalable video coding. In: Proceedings on the 5th International Workshop on Digital Watermarking, IWDW '06, volume 4283 of Lecture Notes in Computer Science, Springer, pp. 407–421, Korea, November 2006.

Zou, Y., Huang, T., Gao, W., Huo, L., 2007. H.264 video encryption scheme adaptive to DRM. IEEE Trans. Consum. Electron. 52 (4), 1289–1297.

Zisimopoulos, H., 2008. Method and Apparatus for Providing Multimedia Broadcasting Multicasting Services, Wo Patent Wo/2008/138,764.