Multimed Tools Appl (2014) 71:1577–1597
DOI 10.1007/s11042-012-1287-6

The study of selective encryption of motion vector based on the S-Box for the security improvement in the process of video

Sung-Sam Hong · Myung-Mook Han

Published online: 18 December 2012

Abstract The Selective Encryption method encrypts the important and requisite parts of data. Since the method does not encrypt the whole of data, the amount of computation is small,which makes it faster and the resources can be used efficiently. The existing selective algorithms have vulnerabilities to the plain text attack and the image restoration attack using the motion vector. They are also vulnerable to the attack in storing and transmitting of the random table data using in encrypt and scramble. In this paper, we propose the selective encryption algorithm of motion vector based on S-Box to remove the vulnerabilities of the existing selective algorithms. The motion vectors generated by the end of motion estimation function of video encoding/decoding xored with S-Box table, are replaced to certain location by using mapping table. The S-Box and mapping table are generated by the secret key through the Rivest Cipher 4 (RC4) encryption algorithm. The proposed algorithm enhances the resistance against attacks through the reinforcement of video security, and thus, reduces the vulnerabilities of the existing algorithms such as I-Frame selective encryption and MVEA. Even though the level of security of the proposed algorithm is higher than the bit scrambling algorithms, it has much better security and higher processing rate than others selective algorithms.

Keywords Encryption · Videos · Information security · Multimedia security

1 Introduction

Recently, the usage of multimedia contents has been rapidly increased due to fast-growing digital content-related service, and it increased the concern to protect such

S.-S. Hong · M.-M. Han (✉)
Department of Computer Science, Gachon University, Seongnam, Korea
e-mail: mmhan@gachon.ac.kr

S.-S. Hong
e-mail: sungsamhong@gmail.com

data. In general, encryption, scaramble and watermarking have been used to protect data against attackers [4, 20]. S-Box is a set of results created by a certain expression or algorithm, which substitutes and transforms data based on these results.

S-Box used to increase confidentiality in substitution stage of some encryption algorithm. In particular, in the AES and DES, the substitution process by S-Box plays an important role to ensure confidentiality.

The selective encryption method is an algorithm that selects only important or required part of the data to encrypt it. The method does not encrypt all the data, so it can use resources quickly because of the little computation. The method which selectively encrypts images is primarily associated with a compression technology, And that is classified into pre-compression, in-compression, and post-compression.

This paper implemented a system to selectively encrypt motion vectors of images by using a substitution method based on S-BOX. Each element in the motion vector table is substituted by S-BOX, which is created by carrying out the RC4 algorithm with an entered key. Data is hidden by matching the substituted value with a unique mapping table created by the key, and by shuffling each elements position in the motion vector table.

This papers selective encryption method increased the security strength by compensating vulnerability of the existing scrambling schemes, which store and send the pseudo random sequence or the spatial mapping table used in scrambling and use simple operations.

The proposed algorithm showed excellent processing speed that processing time is improved as about 2.5 times, from 37.614 ms to 15.516 ms, compared to the existing I-frame selective encryption algorithm, and it improved the processing speed about 1.07 % compared to 15.683 ms, which is processing time of the MVEA method. In addition, since its bit overhead is 3.14 %, which is lower than 39.1 % of I-frame and 4.476 % of MVEA, it could be considered that it has a high connectivity with resource efficiency or compression technology.

This paper is organized as follows. Section 2 introduces the selective encryption algorithm, which is background of this paper, the existing selective encryption algorithms, and analyzes their problems. Section 3 describes the proposed method that selectively encrypts motion vectors based on S-BOX. Section 4 explains experiments and the results, and Section 5 draws a conclusion.

2 Related work

2.1 Selective encryption of video

The selective encryption has been studied for the purpose of reducing the load of encrypting multimedia data. The selective encryption is a technique that encrypts or transforms part of multimedia data to make the quality of multimedia data unsuitable for being watched if played without proper decoding method [12, 21]. When images are selectively encrypted, it is encrypted using various elements of images. It should be considered the positional association with an image compression codec and which part of images is extracted to encrypt. At present, the selective encryption methods are divided into pre, in, and post compression method according to the relation with a position that compression is carried out [14]. (Or, they are also divided into

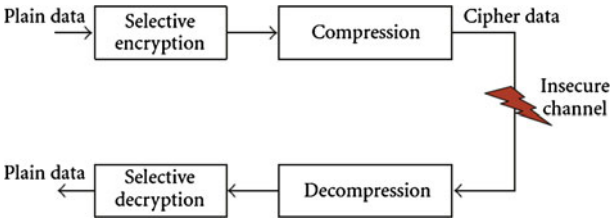

Fig. 1 The pre-compression method

joint-compression and compression-independent encryption method based on the relation with compression [10].) As show in Fig. 1, the pre-compression method encrypts images before compression and decrypts them after restoration, which is suitable for applying to the standard codec. However, the pre-compression selective encryption algorithm often decreases efficiency of compression because it usually has poor relationship with a compression codec, so it has been hardly used, and there have not been a lot of studies.

The in-compression method carries out compression and encryption simultaneously, which requires understanding detailed compression technologies in Fig. 2. In addition, it is difficult to implement because the video encoder/decoder should be often modified. However, it has good efficiency or scalability for encryption, and has little effect on the compression rate. The post-compression method, show is Fig. 3, encrypts after compression and decrypts before decompression. It has the closest relationship with a compression algorithm because there is little participation for compression methods, and the encoder/decoder should not be modified. But, it may be required the diversity following video formats [8].

2.2 I-frame selective encryption

Maples and Spanos [13] suggested a selective encryption method that applies DES encryption into intra frames to decrease overall data volume to be encrypted. This method applied a fundamental idea that P, B frames, which correspond to predicted frames, are meaningless if intra frames are not normally restored. I. Agi and L. Gong, however, showed that there is a problem that the encryption scheme only with intra frames has a poor encryption effect because of intra blocks in P, B frames [2]. However, it has an advantage that it is simple to implement. As shown in Fig. 4, the sequence of I-,P-,B- Frames. I-frame plays an important role for predicting frames, and it is an important element exploiting a starting point of resynchronization, fast

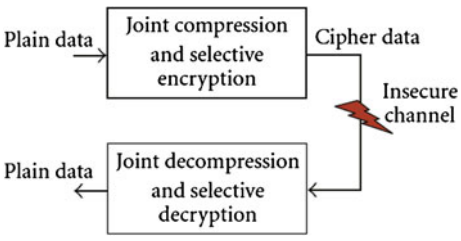

Fig. 2 The in-compression method

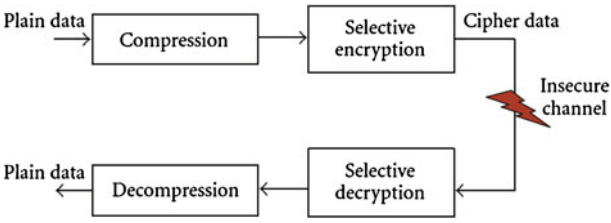

Fig. 3 The post-compression method

rewind, random play, and etc. Therefore, if selecting I-frame for encryption, P- and B-frames, which are predicted after, are significantly influenced, and images may be distorted. Since error on the prediction is accumulated as progressed frames, images could be protected by distorting them. In addition, I-frame occupies about 30–60 % of video.

Weakness and issue I-frame encryption has an advantage that it is simple to implement and not reduces efficiency for compression. However, since it uses DES or AES etc. to encrypt entire frame, it is relatively slower than other selective encryption or scrambling methods also requires much more computations. In addition, since motion vector values are excluded while encrypting I-frames, there are parts vulnerable to an image restoration attack with motion vectors. Some literature has discussed the possibility of using motion vectors to restore video information. Agi and Gong [2] have mentioned that when played with traditional MPEG player, I frame encrypted data stream may still be apprehensible with respect to some special video images, such as American Lady. On the other hand, some video segmentation algorithms using motion vectors may also be considered as kind of video restoration approaches [3]. The image restoration attack with motion vectors extracts motion vector arrays from P, B frames to predict statistical values of motion vectors by the data mining and clustering algorithm to reversely restore images, which 100 % of restoration is difficult but about 80 % could be restored [11].

2.3 Motion vector encryption algorithm (MVEA)

MVEA is a method to encrypt motion vectors for preventing the image restoration attack through motion vectors [1, 11]. The image restoration attack through motion vectors was explained in the above section. It could be said that images are vulnerable to the corresponding attack as the ratio of P, B frames is increased. MVEA is

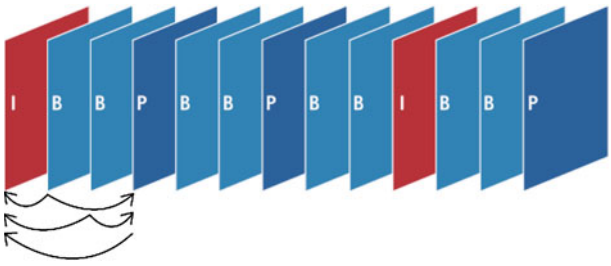

Fig. 4 I-,P-,B-frame sequence

composed of two stages: conceal stage and spatial distancing stage. First, the conceal stage is a process that makes prediction difficult by hiding statistical features of motion vectors, which creates a random list table to carry out XOR operations with the motion vector array. Since the random list table created here is compiled as the Gaussian distribution form so that the result XOR operated with the motion vector array is also created as the Gaussian distribution form, the statistical features of motion vector array could be removed. Second, the spatial distancing stage shuffles spatial association of motion vectors, which uses the regular scrambling method to shuffle the motion vector array [15]. A random number generator creates a mapping table, and the motion vector array is scrambled by the mapping table.

Weakness and issue In the conceal stage, a random number sequence which is created by generationg random numbers as the Gaussian Distribution form of encryption is the method to scramble motion vectors with pseudo random numbers [11]. In the spatial distancing stage, the method uses a random number table when shuffling positions of motion vectors, which a waste of memory and data is additionally arisen for storing and sending the pseudo random number sequence. It is because the method stores and sends the corresponding pseudo random number sequence for using these random numbers when decoding. In addition, if such a random number sequence is not protected, it can be easily exposed to attackers and intercepted during transmission. Therefore, addtional protection is additionally required for the random number sequence data and this can raise the undesirable waste of resources.

2.4 Sign bit scrambling

Discrete Cosine Transform (DCT) based scrambling method, is a scrambling method in the frequency domain, shuffles transformed coefficients to distort images like the wavelet based scrambling method [22]. For example, DCT collects only DC coefficients in a frame to shuffle the DC coefficients through a certain table, or collects coefficients located at the identical frequency to shuffle the coefficients within the identical frequency by making a frequency layer like the sub-band concept in the wavelet transformation. Motion vectors are also divided into blocks in the sub-band to distort images by selectively shuffles the sign-bit of motion vectors to change the vectors direction. Such scrambling method is simple to compute so that it is fast to process and simple to implement. This motion vector scrambling method can provide high level of security if used together with the I-frame encryption.

Weakness and issue The sign-bit scrambling method has simple implementation and algorithm so that it is suitable to use when high performance is required. Because it does not influenced from the compression time, in terms of security, the scrambling method that changes only the sign of motion vectors is simple to compute so that attackers could easily predict and restore images by a brute force attack [22].

3 Proposal system

We propose to the novel selective encryption system using the motion vector (MV). In video compression, the motion vector is the key element in the motion estimation

process. It is used to represent a macroblock in a picture based on the position of this macroblock (or a similar one) in another picture, called the reference picture. Motion estimation is the process of determining motion vectors that describe the transformation from one single 2D image to another; usually from adjacent frames in a video sequence. It is an ill-posed problem as the motion is in three dimensions but the images are a projection of the 3D scene onto a 2D plane. The motion vectors may relate to the whole image (global motion estimation) or specific parts, such as rectangular blocks, arbitrary shaped patches or even per pixel. The motion vectors may be represented by a translational model or many other models that can approximate the motion of a real video camera, such as rotation and translation in all three dimensions and zoom [17]. Closely related to motion estimation is optical flow, where the vectors correspond to the perceived movement of pixels. In motion estimation an exact 1:1 correspondence of pixel positions is not a requirement. Applying the motion vectors to an image to synthesize the transformation to the next image is called motion compensation. The combination of motion estimation and motion compensation is a key part of video compression as used by MPEG 1, 2 and 4 as well as many other video codecs.

This system is implemented by the in-compression method that is carried out by joining the compression codec technology. The entire system structure receives motion vectors, which created after motion estimation in the encoder, and the key value, which is used in the encryption, to encrypt them through the proposed encryption algorithm, and then transmits the bit stream (compressed images), which is created after the entropy stage, to the network as shown in Fig. 5. The used encryption key is sent to the decoder through the security protocol, the decoder carries out the decryption algorithm with a received key to restore the original motion vector after restoring the encrypted motion vector values from compressed images, and then the existing image data is normally outputted. If the key used for encoding is not identical to the key used for decoding, output-images are distorted and original images can be protected. The key for encryption could be used up to 256 bits, and the key stream is generated by a stream cipher scheme to be used in the selective encryption algorithm.

The following section explains the algorithm used for encryption in detail.

3.1 S-Box generated by RC4 and the mapping table

The stream cipher is widely used in wireless and streaming service due to its faster computational speed compared to block cipher. RC4 is one of the stream encryption methods and shares all the characteristics of stream cipher. Therefore, RC4 is selected as most suitable method for encryption in this paper. The stream cipher is relatively used less frequently than the block cipher. However, the former is much used in the environment such as wireless or streaming service because it is lighter and faster than the latter. RC4 is also one of the stream encryption methods, which computation speed is so fast that it is suitable to apply into this system.

This system exploits the RC4 to generate the S-Box used in the first stage. The reason using the RC4 is to create a random number table through the encryption algorithm depending on specified key values rather than a pseudo random number table by a usual random number generation, and use the result as the S-Box.

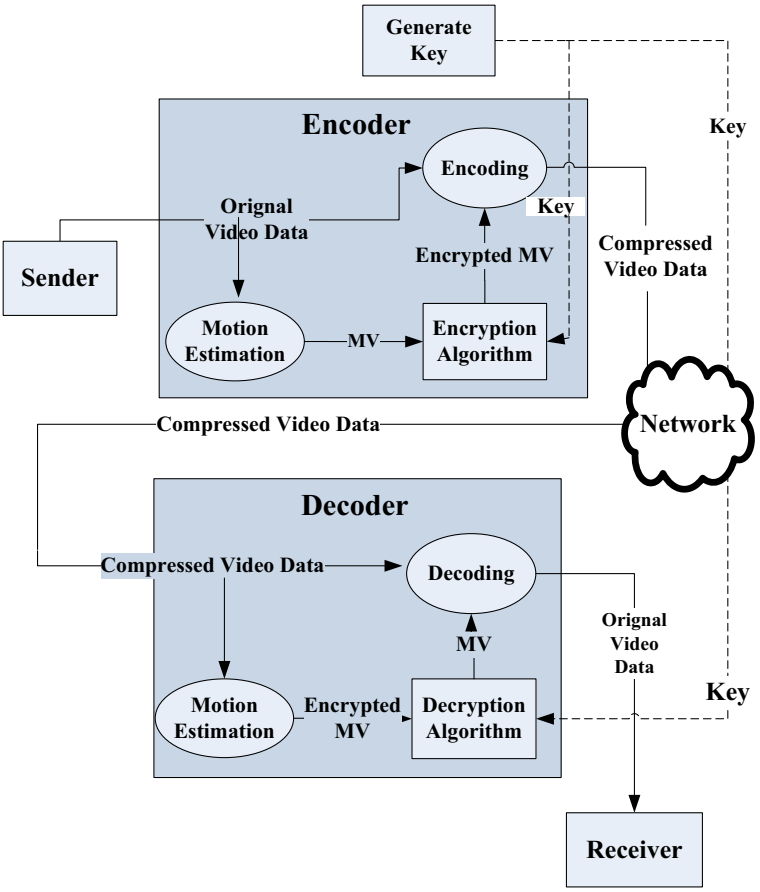

Fig. 5 Proposal system architecture

The process of creating S-Box is to initially generate it as a size, which multiplies macro block column (MBC) value by macro block row (MBR) value, and to initialize it by inserting values sequentially in ascending order. Macroblock is an image compression component and technique based on discrete cosine transform used on for still images and video frames, where macroblocks is a set of two or more blocks of pixels [18]. The MBR and MBC are values divided by the size of a macro block of pixel values in the image. The initialized S-Box plays a role of plain text in the RC4. The secret key used in the system is key[i], (i = 0, 1, 2 .... N, $0 < N \leq 256$) the S-Box and key values are delivered into the RC4 encryption module to create cipher text by encrypting S-Box, and the values are stored in the S-Box table again. As the length of key becomes longer, the key value is more difficult to predict so that the strength of security becomes higher. The pseudo code to create S-Box is described in Table 1. Since such created S-Box table is generated by a key generated randomly or a key entered by users, it is difficult to derive S-Box values unless the key value is known. The created S-Box table changes values of motion vectors by carrying out XOR operations with the motion vector table.

Table 1 Generator of S-Box

Procedure sBoxGeneratorByRC4(key)
Input: The encryption key.
Output: The S-Box is generated to RC4,
Size of S-Box is size of $MBR \times MBC$.
FOR i = 0 to $MBR \times MBC$
Sbox[i] = i + 1
END FOR
Buffer = RC4(Sbox,key)
Sbox = buffer
RETURN Sbox

The second stage of encryption creates a mapping table to shuffle positions of motion vectors depending on the position value in the mapping table. The mapping table used in this stage is created from S-Box, and values in the mapping table are also changed depending on the key entered because S-Box is created by the key. If the mapping is represented as M[i] (i = 0, 1, 2 .... N), where N should be within the range of $0 < N \leq (MBC \times MBR)$. M is initialized by inserting each element value of M sequentially from the first value to the Nth value. Such entered permutation is used as position values in the mapping process. Next, values in the initialized M are shuffled by exploiting S-Box and a sorting algorithm.

The shuffling method delivers S-Box and M into the sorting module, and aligns S-Box by an alignment algorithm. In the process aligning S-Box, the position of each element is changed, if jth positions of S-Box and k th position of S-BOx are exchanged, the corresponding M[j] and M[k] are also exchanged. For example, if the first element value is moved to the (N-5)th position and vice versa in S-Box by alignment, M also exchanges the first and (N-5)th elements in the same way. The process in this example is represented in Fig. 6.

If the alignment is completed like this, values in S-Box are aligned in non-decreasing order, and values in M are shuffled. The alignment algorithm used here exploited the fastest quick sort algorithm for performance.

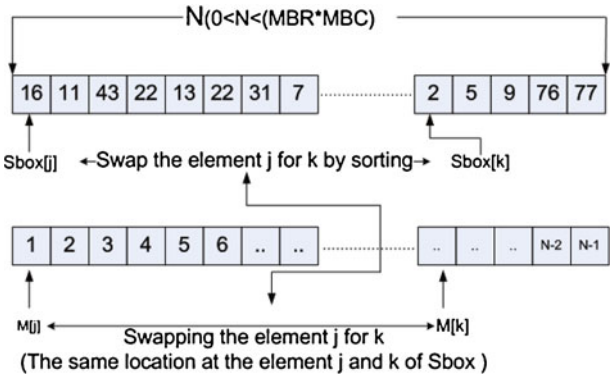

Fig. 6 Shuffle the mapping table

3.2 Encryption

The encryption process receives the motion vector map table data, which is stored in a macro block size unit after completing the motion estimation process in P-frame (Inter-frame) generated firstly after the intra coding, to carry out the encryption. The encryption process is composed of two stages. S-Box stage, Stage to shuffle the position vlaues. The encryption process is described in Fig. 7.

S-Box stage This stage can be explained as follows:

1. A key value is created. The key could be entered by users, or generated as a random number by the system. The length of key should be between 40 to 256 bytes and the strength of security becomes higher as its length is increased.

$$key[i],\ 40 \leq i \leq 256 \tag{1}$$

2. The S-Box table of $MBR \times MBC$ size is created through the RC4 encryption module with the entered key. If the key value is changed, values in the S-Box table are also changed. The size of S-Box table is the same as the size of motion vector table.

$$sBox[i],\ 0 < i \leq (MBR \times MBC) \tag{2}$$

3. XOR operation is applied sequentially to each value in the created S-Box table and the motion vector table, and the result is inserted into the motion vector table again. The motion vector is Motion Vector(MV)

$$MV' = MV \oplus sBox \tag{3}$$

Stage to shuffle the position values The stage to shuffle the position values exchanges the motion vector position of each macro block with the motion vector of macro block at other position by matching the motion vector table (MV) encrypted after the S-Box process with the mapping table created by the mapping table-generating algorithm. The index value of macro block is stored in each element in the mapping table M, and the position of the corresponding macro block is exchanged

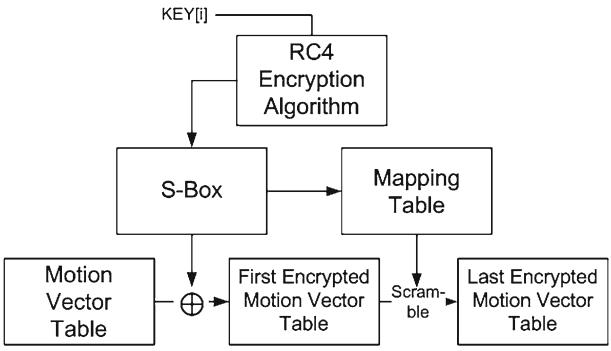

Fig. 7 Encryption process

with the motion vector value of macro block at the position indicated by the element value in table M. This process could be explained as follows:

1. A key is entered to create an S-Box table through the RC4 algorithm.
2. A table M used for mapping is created depending on the S-Box table value. At this time, the size of S-Box, M, and MV tables are identical.
3. The created M and the primary encryption motion vector table created in the S-Box stage are sequentially searched from 0 (the position of the first macro block) to MBR*MBC position (the position of the last macro block,) and the element value (the index value of macro block) of the mapping table M, which corresponds as a one to one at the position of each macro block, is received to exchange with the motion vector value of macro block at the position indicated by the element value (the index value of macro block) of M. The ith position of original macro block is MV(i) = (r,c), M(i) = v The process above is represented in Fig. 8.

3.3 Decryption

The process to decode motion vectors can be executed in reverse order of the encryption process. The decoding module carries out decoding with the key used for encryption after obtaining the motion vector table from the bit stream at the decoder. If the key is different, it is not decoded normally. The decoding process is represented in Fig. 9.

It should be noted that the inverse shuffling process of decoding does not directly use the mapping table created from S-Box, but it is executed through the mapping

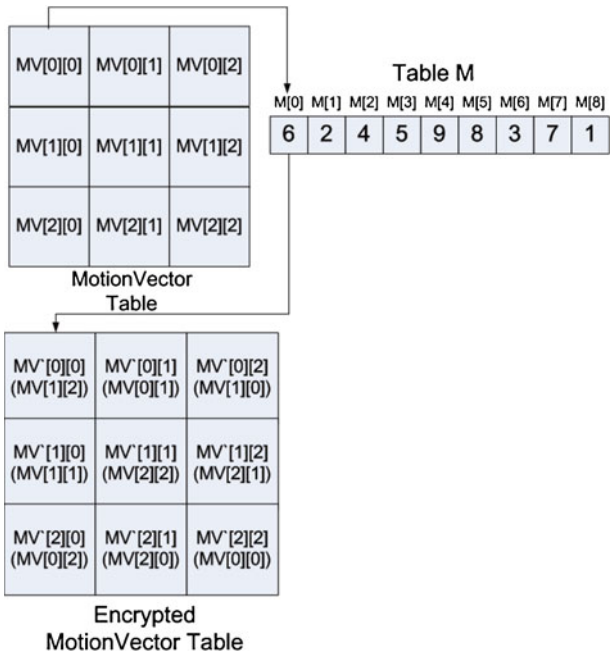

Fig. 8 Stage to shuffle

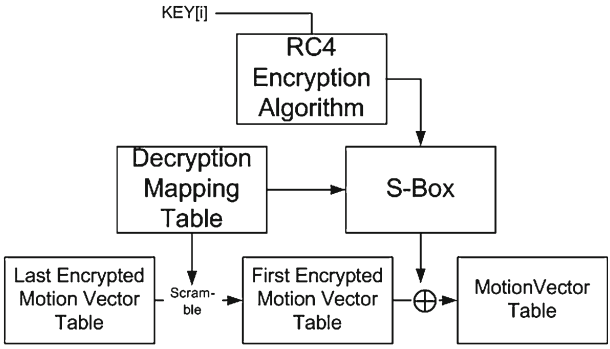

Fig. 9 Decryption process

table for decoding after creating it again from the mapping table through the mapping table-generating algorithm.

1. The key used in encryption is sent to the decoder, and the decoder receives the key.
2. The S-Box table of $MBR \times MBC$ size is created with the received key through the RC4 encryption module.
3. The mapping table M is created from S-Box.
4. The mapping table M′ for decoding, which relocates elements of the motion vector table through the shuffling algorithm, is created from M. In order to initialize elements in M′, the numbers are sequentially inserted by ones from 1 to N ($0 < N \leq (MBC \times MBR)$). Initialized M and the M created from S-box are applied to the shuffling algorithm to generate the mapping table M′ for decoding.

$$M' = shuffle(M, M', 0, MBC \times MBR - 1) \tag{4}$$

   The process written above is represented in Fig. 10.
5. The created M′ and the encrypted motion vector table are sequentially searched from 0 (the position of the first macro block) to $MBR \times MBC$ position (the position of the last macro block), and the element value (the index value of macro

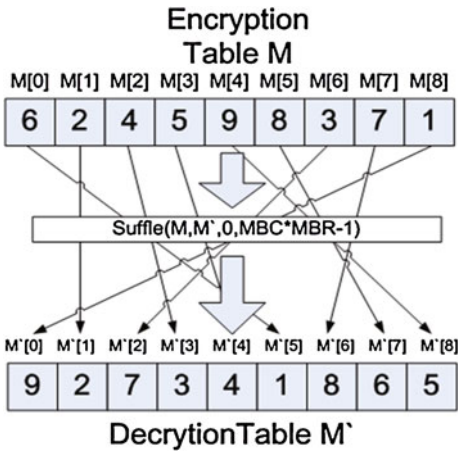

Fig. 10 The decryption table M′ to generate

block) of the mapping table M′, which corresponds as a one to one at the position of each macro block, is received to exchange with the motion vector value of macro block at the position indicated by the element value (the index value of macro block) of M′.

6. XOR operation is sequentially applied to the primary decoding motion vector table, which mapping is completed, and the S-Box value, which is created by applying the identical key used in encryption, to restore the existing motion vector value.

$$MV = FirstdecryptedMV \oplus sBox \tag{5}$$

7. If the motion vector values are restored, images are normally decoded.

4 Experiment

4.1 Experimental environment and method

The environment used for experiment exploited a general PC with Intel Core2Duo CPU 2.26 GHz, RAM 2 GB, and a video source were used for experimenting, which the experiment was conducted with foreman (CIF, 352288, 300 frame) stefan (CIF, 352288, 90 frame), and news (CIF, 352288, 300 frame) videos in Fig. 11. The video compression codec used in the experiment was H.263, which was executed byused the

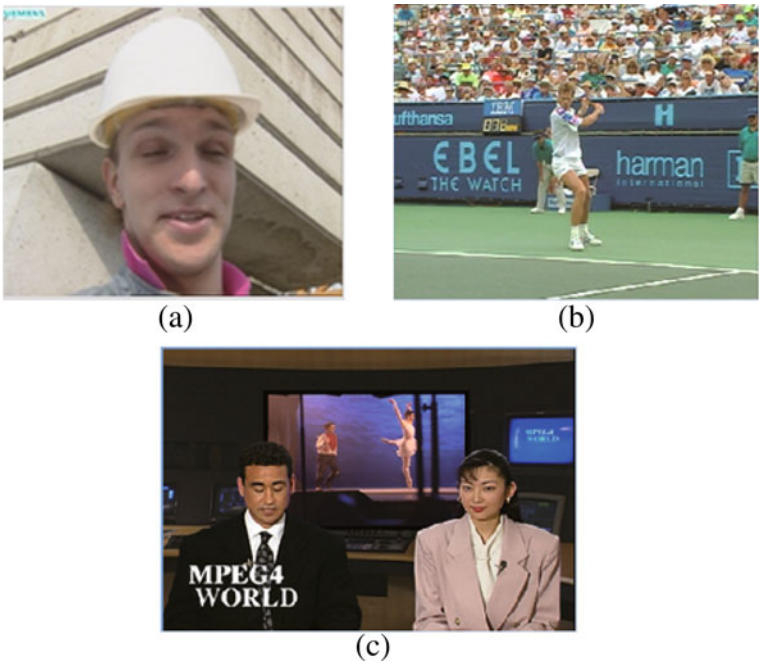

Fig. 11 The videos used for experiment : **a** foreman, **b** stefan, **c** news

TMN Encoder/Decoder software that is an open source [16]. The video compression codec used in the experiment was H.263, which used the TMN Encoder/Decoder that is an open source. The experimental method conducted the selective encryption with I-frame, the motion vector sign bit scrambling, MVEA method of the existing selective encryption methods, and the processing speed and overhead were measured to compare them, and analyze for the strength of security.

4.2 Performance analysis

First, it was investigated the case of restoring the video, which was encrypted through a certain key, without decryption when decoding. As shown in Fig. 12, each samples show encrypted video. In case of foreman and stepan it could be found that the images is are distorted to make difficult to identify. In particular, it they could be found that image distortion is more severe as the area has more motion. On the other hand, the news video with minimal motion in the video has less distortion that also represents less protection.

(a) Original (b) Encrypted Video

Fig. 12 A result of visual quality with encryption

Performance analysis measures the processing time required when applying encryption during encoding for each encryption methodology, and the overhead generating in bit streams after encoding. The encryption was performed 100 times for each algorithm, and for the I-frame encryption, it encrypted I-frames by the AES algorithm after a 128 bit key was entered.

We assume that we have the number of n motion vectors. The time complexity of the sign bit scrambling algorithm is $O(n)$ because the sign bit scrambling algorithm change the sign of n the motion vector. In encryption process, MVEA is expected to be $O(n)$ in both of conceal and spatial distance phase, and to be $O(n^2)$ in generating a Gaussian Random Table. So finally, we can conjecture the fact that MVEA is $O(n^2)$. We expect that the proposed algorithm is $O(n)$ in XOR and shuffle phase, and the creating S-Box and a mapping table phase is $O(nlogn)$ because it uses the quick-sort algorithm(The time complexity of quick-sort is $O(nlogn)$ when creating S-Box and a mapping table by using RC4 encryption(The time complexity of RC is $O(n)$), creating S-Box and a mapping table phase is $O(nlogn + n)$. In conclusion, we assume that sign bit scrambling method has the best performance and the proposed algorithm has better performance than MVEA.

As shown in Table 2, in the result of experiment using foreman cif, it could be found that the motion vector sign bit scrambling method is the fastest as 0.474 ms, and I-frame encryption method is the slowest as 37.614 ms in terms of the speed. As a result, it could be known that the sign bit scrambling is faster than other algorithms because it relatively has less number of computations. In addition, for I-frame encryption, its processing time required for encryption is 37.614 ms, which is slower than 0.474 ms of the motion vector sign bit scrambling method exploiting P-frame, 4.476 ms of MVEA method, 15.516 ms of the proposed algorithm, and its bit stream overhead is 39.1 %, which is also higher than 0.884 % of the sign bit scrambling method, 4.476 % of MVEA method, 3.14 % of the proposed algorithm. Therefore, it could be considered that I-frame encryption has the most significant effect on efficiency for compression. It is considered that the selective encryption, which exploits elements of P-frame, has better performance than the selective encryption exploiting I-frames. The proposed algorithm had the second fastest processing speed, 15.516 ms, which showed excellent data processing ability. In the experiment, MVEA algorithm unexpectedly showed 15.683 ms of processing time, which was slower than the proposed algorithm. The data throughput is total amount of bytes processed per second by encoding and encryption modules, which the sign bit scrambling algorithm showed the highest throughput as 424,849.2 bps, the proposed algorithm showed the second highest throughput as 407,244.2 bps, therefore, it also has a high level in terms of throughput. In addition, as a result of measuring the bit stream overhead, the motion vector sign bit scrambling method is 0.88 %, MVEA is 4.476 %, and the proposed algorithm is 3.14 %. It shows that the proposed algorithm also has relatively less effect on the compression rate. I-frame encryption has overhead of 39.1 %, which is considerably high, so that it is shown that it has much effect on the compression efficiency. Likewise, the experiment using Stepan and News is same the result.

As shown as Table 3, Additionally from the PSNR result, each I-Frame Encryption, sign bit scrambling, and MVEA had 20.72 dB, 20.79 dB, and 13.70 dB, where the proposed algorithm got 13.53 dB. From this result, sign bit scrambling method has minimum security and the proposing algorithm is the most secured method among others.

Table 2 The result of experiment

	Encryption time (ms)	Bitstream overhead (%)	Throughput(bps) (encode + encryption)
– Experiment of foreman cif			
I-Frame			
Encryption	37.614	39.1	383068.4
MotionVector			
Sign bit scrambling	0.474	0.884	424849.2
MVEA	15.683	4.476	400756.0
Proposal Algorithm	**15.516**	**3.14**	**407244.2**
– Experiment of stefan cif			
I-Frame			
Encryption	24.77	44.76	–
MotionVector			
Sign bit scrambling	0.165	6.49	–
MVEA	3.465	9.74	–
Proposal Algorithm	**3.391**	**7.94**	–
– Experiment of news cif			
I-Frame			
Encryption	39.181	49.1	–
MotionVector			
Sign bit scrambling	0.552	0.0	–
MVEA	10.02	13.40	–
Proposal Algorithm	**9.92**	**11.71**	–

Bold items highlight the proposal algorithm and distinguish between the proposed algorithm and the other algorithms

If encrypt to all data of video, it will reduce the QoS because need to many computation and many time. This approach are not appropriate for multimedia services. Therefore, the Selective Encryption Algorithms have been studied. Although some of the video data to be encrypted, it is difficult to recover the original video data. Because, In Compress algorithm, One of frame predict to the other frame. If it is difficult to recognize video, we will consider to provide security.

In the result of performance analysis, the proposed algorithm has the bit stream overhead of 3.14 %, which does not significantly reduce performance of the compression codec. The processing time and bit throughput are also 112.019 s and

Table 3 The result of experiment

	PSNR(db)
– Experiment of foreman cif	
I-Frame Encryption	20.72
MotionVector Sign bit scrambling	20.79
MVEA	13.70
Proposal Algorithm	**13.53**

Bold items highlight the proposal algorithm and distinguish between the proposed algorithm and the other algorithms

407,244.2 bps, respectively, which show higher than other algorithms. Therefore, it could be evaluated that the proposed algorithm shows a high level of performance in terms of processing time, data throughput, and overhead.

4.3 Security analysis

This encryption system encrypts motion vectors through two stages of process. The first stage carries out XOR operations with S-Box generated from RC4. The RC4 algorithm is an encryption algorithm developed by RSA in 1987, which is a stream encryption method that could use a key of various sizes by working in a byte unit. It is evaluated that it is almost impossible to predict cipher texts unless the key is known because it is an algorithm based on the random substitution algorithm, which the period of cipher is larger than 10,100. In addition, there is no successful case for attacking RC4 with a key of more than 128 bits [15].

Therefore, since values of S-Box, which are created in the proposed system, are generated from the key through the RC4 algorithm, attackers could not predict S-Box unless the key is known, so it is evaluated that security of the proposed algorithm is as high as the RC4 algorithm. However, even though XOR calculation is vulnerable to security in XOR calculation stage for S-Box and motion vectors, it could compensate vulnerability for XOR calculation because the second stage shuffles the values again, which XOR calculation is applied, through the table created by the RC algorithm. An advantage of two-stage encryption is that the first XOR stage hides statistical values of vectors created through the motion estimation stage by XOR calculations, and the second shuffling stage hides the position values to prevent exposing the original positions by replacing positions of motion vectors. Confidentiality could be improved for motion vectors through repeating encryption process twice.

If attackers, who do not know the key, try to find values of the motion vectors or mapping table through a brute force attack, they should carry out the following attempt [11]. m and n is the number of rows and columns in the macro block, respectively.

$$P_{mn}^{mn} \approx (m \times n)! \tag{6}$$

$$P_{10\times 8}^{10\times 8} = (10 \times 8)! \approx 7.1569 \times 10^{118} \tag{7}$$

If an image with 160×128 pixels is divided into 16×16 macro block unit, calculations of $(10 \times 8)!$ should be carried out for a brute force attack on a frame. If the number of pixels is 352×288, calculations of $(18 \times 22)!$ would be required. In addition, if the number of encrypted frames is n, the number of times being calculated would be increased by n times according to the number of frames. The chance of a brute force attacks success is usually decided as about a half of total number of attempts, and it is decided that a brute force attack is impossible for this algorithm.

$$P = n(18 \times 22)! \tag{8}$$

Beside, as a result of PSNR value(Table 3), it could be found that the proposal algorithm is the lowest PSNR value as 15.53 db, and the motion vector sign scrambling algorithm is the highest PSNR value as 20.79 db. Therefore, the proposal algorithm is provided to better confidentiality than the other algorithms.

The S-Box on our proposal algorithm was generated by RC4 encryption algorithm using the secret key(symmetric key). Even if an attacker had a ciphertext, he can't decrypt to a ciphertext because he don't know the secret key. If the secret key is changed, an chiphertext is changed also. If an attacker don't know the key value even if the encryption algorithm input data is same, can't find the input/output from the association. Also, an attacker can't analyze the S-Box original data from the past data set because the S-Box is generated disposable by key. We don't consider that the secret key is exposed, because we assumed that the secret key is protected by safety security system(in security channel, in security storage). For this reason, we can provide to the non-linearity of the S-Box by using RC4 algorithm and the secret key.

For example, we assume to have 4*4 Motion Vectors Table which use an input data in encryption alghorithm. It is the original(plaintext) Motion Vector Table(MVT).

$$\mathbf{MVT(input)} = \begin{pmatrix} 41 & 67 & 34 & 0 \\ 69 & 24 & 78 & 58 \\ 62 & 64 & 5 & 45 \\ 81 & 27 & 61 & 91 \end{pmatrix}$$

In case 1, the S-Box is generated by RC4 algorithm using case1_key.

$$case1\_key = \{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\}.$$

The mapping table of S-Box of case 1 is

$$\mathbf{MappingTable\ of\ sBox(case1)} = \begin{pmatrix} 1 & 9 & 5 & 7 \\ 10 & 6 & 11 & 3 \\ 15 & 13 & 12 & 2 \\ 8 & 14 & 4 & 0 \end{pmatrix}$$

Then,the MVT is encrypted by the process of two stage encrpytion. Firstly, the MVT operates exclusive-operation with S-Box. Secondly, it is shuffled by the

mapping table of S-Box. In the result, a ciphertext motion vector table of case 1 has been generated by proposal algorithm.

$$\textbf{cipher MVT(output of case1)} = \begin{pmatrix} 89 & 51 & 10 & 136 \\ 108 & 5 & 96 & 135 \\ 59 & 118 & 103 & 240 \\ 61 & 28 & 191 & 247 \end{pmatrix}$$

Also, in case 2, the S-Box is generated by RC4 algorithm using case2_key.

$$case2\_key = \{1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\}.$$

The mapping table of S-Box of case 2 is

$$\textbf{MappingTable of sBox(case2)} = \begin{pmatrix} 0 & 15 & 10 & 6 \\ 3 & 4 & 13 & 11 \\ 7 & 1 & 9 & 5 \\ 2 & 8 & 14 & 12 \end{pmatrix}$$

Then, we encrypt MVT(same the process of encrpytion of case 1). In the result, a ciphertext motion vector table of case 2 is generated by proposal algorithm.

$$\textbf{cipher MVT(output of case2)} = \begin{pmatrix} 40 & 52 & 158 & 167 \\ 77 & 40 & 192 & 225 \\ 49 & 73 & 200 & 224 \\ 24 & 188 & 143 & 247 \end{pmatrix}$$

We don′t discover similarity or correlation between ciphertext motion vector table of case 1 and case 2. And we discover that chipertext are changed by key value. In other words, the ciphertext of case1 and case2 are non-linearity. Also, the correlation between input and output data is non-linearity. If key don′t expose to an attacker, he/she isn′t able to get a plaintext from a ciphertext. Thus, it is noted that the proposal algorithm can provide from the confidentiality from known-ciphertext attack.

The proposed algorithm protects images by encrypting motion vectors of images to distort them so that makes attackers could not simply decode unless they know the key even if they intercept images. In addition, this algorithm protects images from the restoration attacks for images by analyzing motion vectors, which is vulnerable in the existing I-frame encryption. The algorithm also has improved security for the pseudo random table by compensating possibility of exposing data in storage and transmission for the pseudo random list and mapping table, which is a vulnerability of MVEA algorithm, to generate S-Box and a mapping table through the RC4 algorithm. Since the same values of S-Box and a mapping table are always created if the same key is entered, there is no need to prepare additional storage space or transmit them for S-Box or the mapping table, it is also reduced the waste of resources such as memory.

5 Conclusion

This paper proposed a method to selectively encrypt motion vectors based on S-Box for processing and protecting images effectively. The selective encryption is applied because encryption of a whole image is computationally intensive and slow, and it

makes unauthorized users, who do not know the key, could not output normal images by encrypting motion vectors, which play an important role in image identification and image compression processing, to distort images.

Its resistance to attacks is high because S-Box and the mapping table are created through the RC4 encryption algorithm that reliability is already proved, and the vulnerability for data exposure is compensated because S-Box and the mapping table are generated by a key so that there is no need to store and transmit the tables unnecessarily. Likewise, it increases security for images to strengthen resistance to attacks by removing vulnerability of other selective encryption algorithms such as I-frame selective encryption, MVEA, and etc. Even though processing speed is slower than simple scrambling methods for the sake of improving security, it could be evaluated that performance of this encryption algorithm is excellent because it shows higher speed and throughput compared to other selective encryption methods, and the overhead of bit streams is also small. However, since distortions of images is low when using only motion vectors for images with little motion, the security should be improved by selectively encrypting other image elements together or combining other selective methods if the user wants high security.

For the future study, even though this papers experiment used H.263 codec, the encryption algorithm would be applied also to the compression codec exploiting motion estimation of MPEG or H.26x series so that performance of the algorithm could be evaluated, and it should be reduced the unnecessary memory use and the overhead for bit streams, and the processing speed problem should be improved by optimizing the selective encryption and RC4 algorithm. In addition, it would be improved the security for images by combining other image elements to selectively encrypt them.

Acknowledgement This work was supported by the Gachon University research fund of 2012.(GCU-2012-R237).

References

1. Ahn J, Jeon B (2005) Digital video scrambling methods using motion vector and intra prediction mode. Journal on Korea Electronic Science 42(4):133–142
2. Agi I, Gong L (1996) An empirical study of secure MPEG video transmissions. The Internet Society Symposium on Network and Distributed System Security, p 137
3. Baraldi A, Blonda P, Parmiggiani F, Satalino G (1998) Image segmentation through contextual clustering. In: International computer science institute
4. Chang FC, Huang HC, Hang HM (2007) Layered access control schemes on watermarked scalable media journal of VLSI signal processing systems for signal, image, and video technology. IEEE Int Symp Circuits Syst, (ISCAS 2005) 5:4983–4986
5. Cheng H, Li X (1996) On the application of image decomposition to image compression and encryption. Commun Multimedia Security II 116–127
6. Forouzan BA (2008) Cryptography and network security. McGraw-Hill, New York
7. Han M-M, Kim G-S (2009) A Study of selective encryption for images using tree structures. Korea Society for Internet Information
8. Hiroshi H (1995) MPEG. KyoboBook, Seoul
9. Kankanhalli MS, Guan TT (2002) Compressed-domainscrambler/descrambler for digital video. IEEE Trans Consum Electron 48(2):356–365
10. Liu F, Koenig H (2009) A survey of video encryption algorithms. Computer Security 29(1):3–15
11. Liu Z, Li X (2004) Motion vector encryption in multimedia streaming. In: Multimedia Modelling Conference, Proceedings. 10th International (MMM'4), pp 64–71

12. Macq B, Wuisquater J-J (1995) Cryptology for DigitalTV broadcasting. Proc IEEE 83(6):944–957
13. Maples TB, Spanos GA (1995) Performance study of a selective encryption scheme for the security of networked. Real-time video. In: Proc. ICCCN in Las Vegas
14. Massoudi A, Lefebvre F, De Vleeschouwer C, Macq B, Quisquater J-J (2008) Overview on selective encryption of image and video: challenges and perspectives. EURASIP Journal on Information Security 2008:179290. doi:10.1155/2008/179290
15. Stallings W (2008) Network security essential (Application and Standard). Prentice Hall, New Jersey
16. TMN-En/Decoder (H.263) Copyright (C) 1995, 1996 Telenor R&D, Norway
17. Torr PHS, Zisserman A (1999) Feature based methods for structure and motion estimation. ICCV Workshop on Vision Algorithms
18. Watinson J (2004) The MPEG handbook, 2nd edn. Focal Press
19. Wen J, Severa M, Zeng W, Luttrell M, Jin W (2002) A format compliantconfigurable encryption framework for access control of video. IEEE Trans Circuits Syst Video Technol 435–440
20. Wohlgemuth S, Echizen I, Sonehara N, Muller G (2011) On privacy-compliant disclosure of personal data to third parties using digital watermarking. Journal of Information Hiding and Multimedia Signal Processing 2(3):270–281
21. Wu CP, Kuo J (2005) Design of integrated multimedia compression and encryption systems. IEEE Trans Multimedia 7(5):828–839
22. Zeng W, Lei S (2002) Efficient frequency domain selective scrambling of digital video. IEEE Trans Multimedia 5(1):118–129

Sung-Sam Hong was born in Seoul, Korea, in 1983. He received the Bachelor degree in Computer Science from Kyungwon University, Korea in 2009 and Master degree Computer Science from Kyungwon University, Korea in 2011. He is currently a researcher for Multimedia Encryption and Information Security in Information Security Lab of Gachon University. His research interests include Multimedia Security, Information Security, Mobile Security, Cryptology, Reverse Engineering.

Myung-Mook Han received MS degree in computer science from New York Institute of Technology in 1987 and Ph.D. degree in information engineering from Osaka City University in 1997, respectively. From 2004 to 2005, he was a visiting professor at Georgia Tech Information Security Center(GTISC), Georgia Institute of Technology. Currently, he is a professor in the Department of Computer Engineering, Gachon Univ., Korea. His research interests include Information Security, Intelligent System, Mobile Computing. He is a member of IEEE and IEICE.