

## Chapter 2

# Advances in Multimedia Encryption

**Abstract** Rapid advances in embedded systems and mobile communications have flooded the market with a large volume of multimedia data. In this chapter, we present a summary of multimedia compression and encryption schemes, and the way they have evolved over the decades.

Chapter goals:

- Familiarize with existing deal of research approaches in the area of multimedia encryption.
- Summary of key/popular schemes proposed in research literature.

### 2.1 Introduction

Security is becoming an escalating concern in an increasingly multimedia defined world. The recent emergence of embedded multimedia applications such as mobile-TV, video messaging, and telemedicine have increased the impact of multimedia and its security on our personal lives. For example, a significant increase in the application of distributed video surveillance technology to monitor traffic and public places has raised concerns regarding the privacy and security of the targeted subjects.

Multimedia content encryption has attracted more and more researchers and engineers owing to the challenging nature of the problem and its interdisciplinary nature in light of challenges faced with the requirements of multimedia communications, multimedia retrieval, multimedia compression and hardware resource usage.

With the continuing development of network communications (wired and wireless), easily capturing videos and rapid advances in Internet technology and embedded computing systems multimedia data (images, videos, audios, etc.) are of importance for use more and more widely, in applications such as video-on-demand, video conferencing, broadcasting, etc. Now, it is closely related to many aspects of daily life, including education, commerce, defense, entertainment and politics. In order to maintain privacy or security, sensitive data need to be protected before transmission or distribution. The advancements in ubiquitous network environment, and rapid developments in cloud computing have promoted the rapid delivery of digital multimedia data to the users.

Users are eager to not only enjoy the convenience of real-time video streaming but also share various media information in a rather cheap way without awareness of possibly violating copyrights. In view of these, encryption and watermarking technologies have been recognized as a helpful way in dealing with the copyright protection problem in the past decade. Encryption allows secure end–end communication of data while digital watermarking allowing still faces some challenging difficulties for practical uses; there are no other techniques that are ready to substitute it.

Within the signal processing and multimedia communities, many schemes have been proposed for protecting sensitive information while allowing certain legitimate operations to be performed. These schemes typically lack a rigorous model of privacy, and their protection becomes questionable when scaled to large datasets. The cryptography community has long developed rigorous privacy models and provably secure procedures for data manipulations. However, these procedures are primarily designed for generic data. As a result, they usually lead to a blow up in computational costs and overheads when applied to real-life multimedia applications.

There has been a great deal of effort to design algorithms and architectures for multimedia security (particularly encryption) suitable for mobile and embedded devices which have tighter constraints on computational resources.

## 2.2 Multimedia Encryption Problem

Multimedia encryption involves changing the multimedia datastream itself to ensure secure transmission of video data between client and server (or two nodes). It can be accomplished by means of standard symmetric key cryptography where multimedia bitstream is treated as a binary sequence and the whole data can be encrypted using conventional cryptosystem such as AES or DES [18].

In general, when the application requirements are not dynamic (not a real-time streaming) we can treat bitstream as a regular binary data stream and use the conventional encryption techniques. Encrypting the entire multimedia stream using standard encryption methods is referred to as the naive algorithm. There are many practical constraints in case of mobile multimedia which make such a scheme not practical in real-life scenario. First there are issues with available computational resources in mobile devices which combined with low battery life and limited device area limit the application of AES or DES like ciphers. Unlike desktop processors, dedicated AES co-processor will cause high power and area requirements. This can be understood with the example of GSM mobile phones which use a much lighter cryptographic cipher for data encryption. A5 is the stream cipher used to provide over-the-air communication privacy in the GSM cellular telephone standard and is used in various variants. A5/0 utilizes no encryption while A5/1 is the original A5 algorithm used in Europe. A deliberate weakening of the algorithm was proposed as A5/2, but it was cryptanalyzed the same month as it was published. The A5 algorithm is much simpler in implementation than AES, and is implemented using stream ciphers. A5/3, also known as KASUMI is a stronger encryption algorithm

created as part of the 3rd Generation partnership Project (3GPP). The Secure Real-time Transport Protocol, or for short SRTP [22RFC3711?], is also an application of naive approach, where multimedia data are packetized and each packet is individually encrypted using AES. The HDTV encryption standard also uses a similar approach, allowing one to choose from AES or the lightweight M6 cipher.

Communication encryption of multimedia content is a problem beyond the application of established encryption algorithms, such as DES or AES, to its binary sequence [20]. This is primarily due to the way multimedia is used commercially. Unlike data encryption, where we want to encrypt a complete bitstream, mobile multimedia encryption introduces several challenges. Firstly, the content providers want to ensure real-time streaming of videos. The mobile phone users will not wait for authentication and encryption of downloaded videos if they need to wait for long times. Real-time streaming of secure bitstream is a serious challenge for mobile multimedia delivery, because the wireless environment (in which mobile phones are operating) already pose serious bandwidth restrictions. First of all, the user may search (in real time) for a particular video at run-time from a digital library. Further, video compression is done in a scalable way, to allow a single compressed copy at server to be downloaded at multiple bit rates. Transcoding may be required at times. We need encryption schemes which can maintain format compatibility and not slow any of these operations.

Further, it involves careful analysis to determine and identify the optimal encryption method when dealing with audio and video media. To identify an optimal security level, we have to carefully compare the cost of the multimedia information to be protected and the cost of the protection itself. If the multimedia to be protected is not that valuable in the first place, it is sufficient to choose relatively light levels of encryption. On the other hand, if the multimedia content is highly valuable or represents a government or military secrets, the cryptographic security level must be the highest possible. For many real-world applications such as pay-per-view, the content data rate should be very high, but the monetary value of the content may not be high at all. Thus, very expensive attacks are not attractive to adversaries, and light encryption may be sufficient for distributing these videos. On the other hand, applications such as video-conferencing or videophone require much higher level of confidentiality. Maintaining such high level of security and still keeping a real-time and limited-bandwidth constraints is not easy to accomplish.

## 2.3 Common Approaches to Video Encryption

### 2.3.1 Scrambling

Scrambling is one of the simplest form of encryption that can be applied to multimedia data. It usually refers to encryption methods which perform random permutations to video data using some scheme. The histogram of image generally remains the same except for the fact that the individual positions are shuffled. Early work on

signal scrambling was based on using an analog device to permute the signal in the time domain or distort the signal in the frequency domain by applying filter banks or frequency converters [22]. However, these schemes are extremely easy to crack using modern computers. With the popularization of DSP (Digital Signal Processing), in the digital signal domain focus was placed on scrambling in the domain of orthogonal transforms (DFT, DCT, wavelet transform, Hadamard transform, etc.) [22]. The security provided by scrambling alone is low. It also decreases the compression efficiency of video bitstream leading to compression losses and increased size of video file.

Scrambling is often used as an easy way to encrypt live analog/digital video signals such as surveillance camera feeds where heavy ciphers are ruled out because of computational delay. Some most common techniques include:

1. **Line Inversion Video scrambling:** In this method whole or some parts of the signal scan lines are inverted. This scheme is relatively cheap and simple to implement but the security level achieved is low.
2. **Sync Suppression Video scrambling:** The horizontal/vertical line syncs are hidden or entirely removed in this method. This provides a low-cost solution to Encryption and provides good quality video decoding. A typical disadvantage is that the level of obscurity reached by this scheme depends on video content.
3. **Line Shuffle Video scrambling:** In this scheme each signal line is re-ordered on the screen. Although this scheme provides reasonable security, it requires a lot of storage to re-order the screen.
4. **Cut and Rotate Video scrambling:** In this method, each scan line is cut into pieces and then re-assembled in a permuted manner. This scheme provides a compatible video signal, gives an excellent amount of obscurity and good decode quality and stability. However, it requires specialized scrambling equipments.

Compression algorithms have been designed for the unscrambled signals and they use the statistical characteristics of raw data. Once the signal is scrambled, these characteristics will change and the performance of the compression filter will be degraded.

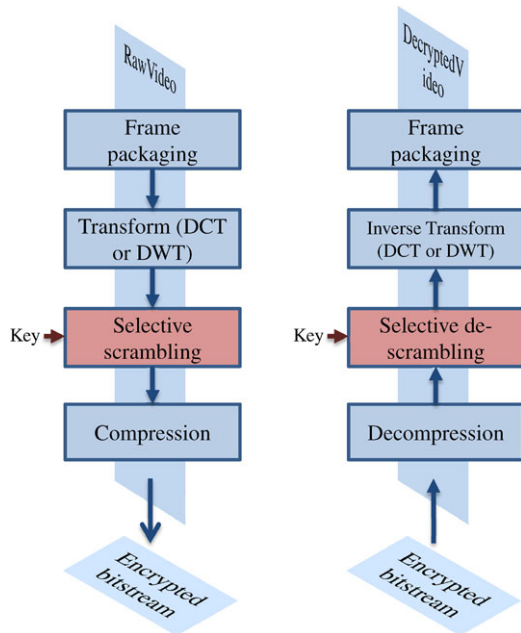
In Zig-Zag permutation [19], instead of mapping the  $8 \times 8$  block (used in DCT compression stage of video coder) to  $1 \times 64$  vector in “Zig-Zag” order, it maps individual  $8 \times 8$  block to  $1 \times 64$  vector by using a random permutation list (secret key). This algorithm consists of three steps.

1. Generation of a permutation list with cardinality 64.
2. Splitting of coefficients according to permutation list, and passing the result to the entropy coding procedure.

However, this method decreases the video compression rate because the random permutation distorts the probability distribution of Discrete Cosine Transform (DCT) coefficients.

A digital image-scrambling scheme should have a relatively simple implementation, amenable to low-cost decoding and low-delay operation for real-time interactive applications. IT should be independent of compression algorithm and should

**Fig. 2.1** Joint scrambling and compression framework proposed by Zeng and Liu [23]



not incur any loss to the compression operation. We present a case study of the technique presented by Zeng and Liu [23] to better understand the scrambling operations. An overview of their approach is presented in Fig. 2.1.

The authors first transform the input signal into the frequency domain using Discrete Cosine Transform (DCT) or Discrete Wavelet Transform (DWT). The transform coefficients are then divided into subsequent operations which permute their values within the image. The motion vectors are also subject to random sign changes and shuffling. A cryptographic key will be used to control the scrambling. The scrambled coefficients and motion vectors are then passed through compression block to obtain compressed bitstream. Authorized users can obtain the original content back using the same key. The scrambling operation is performed prior to compression allowing preservation of multimedia-specific compression properties such as transcoding and scalability. Frequency domain scrambling makes it easier to control transparency (i.e., what part of the video data is allowed to be freely accessed).

The encryption/decryption operations are designed to preserve, as much as possible, the transformed image properties that allow entropy coders to efficiently compress an image.

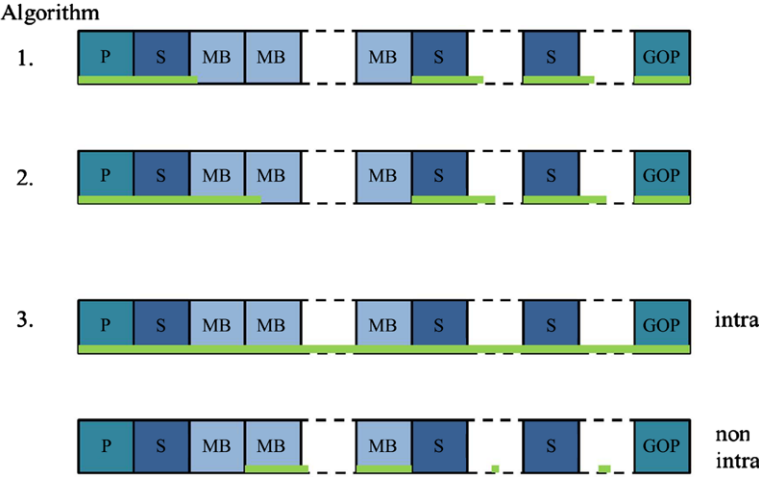
Aside from easy and secure transcoding, the joint scrambling-compression framework proposed by [23] provides some other advantages over those that perform scrambling on the compressed bitstreams.

1. **Flexibility to perform selective encryption:** In the frequency domain, it is easier to identify what parts of the data are critical for security purpose. This allows providing different levels of security and transparency.

2. **Encrypting incompressible segments:** It is also easy to identify what parts of the data are not compressible. For example, the sign bits of the coefficients are usually difficult to compress; yet they are critical for security purposes. This incompressible data segment can be selected to scramble without affecting the overall compression efficiency. Some other data segments, such as the motion vector information, are usually losslessly compressed. They therefore can be selected to encrypt without the need to consider the transcoding issue, since it does not make much sense for the transcoder to recode this part of the compressed data. Notice that the selected data can be easily located in the frequency domain without incurring any processing overhead. On the contrary, since the compressed bitstream is usually variable length coded, it is generally difficult to perform fine-scale selective encryption on the compressed bitstream without incurring processing and bit overheads.
3. **Less vulnerability to channel errors:** Encryption after compression such as using AES over MPEG is more vulnerable to channel errors because a block of 128 bits in AES are bound together so that one single bit error in a block will cause the synchronization word/bits contained in that block to be erroneous. Since the synchronization information is hidden in the encrypted video stream; it will be harder to recover from transmission errors in the network. On the other hand, spatial scrambling in the frequency domain has no adverse impact on the error resiliency.
4. **Compatibility to transform domain signal processing:** Scrambling involves changing the spatial positions of individual frequency coefficients. Watermarking and other transform domain tasks can be performed without requiring cryptographic key.

Some of the techniques for scrambling by the author are as follows:

1. *Selective Bit Scrambling:* The first basic approach scrambles selected bits in the transform coefficients to encrypt an image. Each bit of a coefficient can be viewed as one of three types. Significance bits for a coefficient are the most significant bit with a value of 1, and any preceding bits with a value of 0. These bits limit the magnitude of the coefficient to a known range. Refinement bits are the remaining magnitude bits, used to refine the coefficient within the known range. The sign bit determines whether the known range is positive or negative.
2. *Block Shuffling:* To increase the level of security, block shuffling is proposed. We divide each subband into a number of blocks of equal size. The size of the block can vary for different subbands. Within each subband, blocks of coefficients will be shuffled according to a shuffling table generated using a key. The shuffling table generally will be different for different subbands, and can vary from frame to frame.
3. *Block Rotation:* To further improve security with little impact on statistical coding, each block of coefficients can be rotated to form an encrypted block.



**Fig. 2.2** Different levels of security offered by the SECmpeg algorithm (Meyer and Gadegast [10]). P, S, MB and GOP refer to Primary Coding Unit (Individual Image), Slice Layer (restart points within a frame), Macro block layer (Motion Compensation Unit) and Group of Picture

### 2.3.2 Post-compression Encryption Algorithm

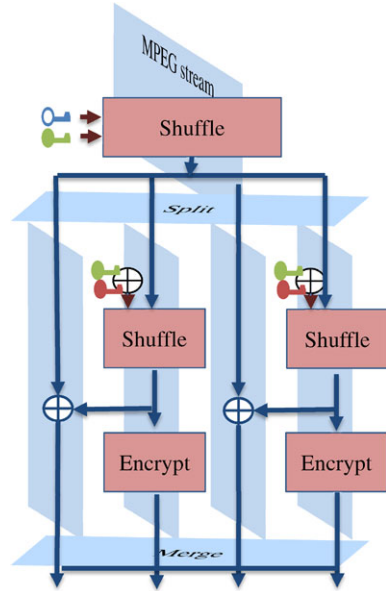
The Secure Real-time Transport Protocol, or the naive approach encrypts the compressed bitstream by packetizing multimedia data and individually encrypting every packet using AES. Although it is secure, it has huge computational overheads and it is not conducive to different desired properties of compressed bitstreams in general, owing to encryption of compressed data.

Many different algorithms have been proposed—which are format compliant, or have low computational requirements. Meyer and Gadegast [10] proposed a selective video encryption scheme called Secure MPEG or SECmpeg for the MPEG-1 video coding standard. See Fig. 2.2 for details. It offers different levels of security by encoding different parts of compressed bitstream:

- Algorithm 1: It encrypts only the headers from the sequence layer to the slice layer.
- Algorithm 2: It encrypts additionally low frequency DCT coefficients of all blocks in the I-frames.
- Algorithm 3: It encrypts all I-frames and all I-blocks in the P- and B-frames.
- Algorithm 4: It encrypts the whole MPEG-1 sequence with the naive algorithm.

The approach has some notable limitations: computations savings are not significant because I-frames constitute 30–60 % of an MPEG video. Moreover Agi and Gong [22agi96??] demonstrated that some scene contents are still discernible by directly playing back the selectively encrypted video stream on a conventional decoder. Maples and Spanos [17] presented a similar approach called AEGIS. All

**Fig. 2.3** The Video Encryption Algorithm proposed by Qiao and Nahrstedt [14]. MPEG packets are shuffled using key information for fast, efficient encryption



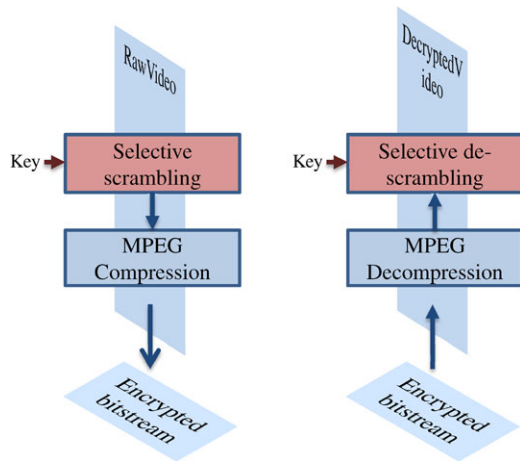
I-frames in an MPEG-video stream are encrypted, while P- and B-frames are left unencrypted. The AEGIS algorithm is almost same as SECmpeg level 2.

Qiao and Nahrstedt [14] introduced the Video Encryption Algorithm (VEA) which reduces the computational complexity to almost half. This video encryption algorithm is detailed in Fig. 2.3. Half of the bitstream is encrypted with a conventional encryption algorithm such as AES and is then used as key to XOR with the other half stream. The basic VEA algorithm is vulnerable to plaintext-attacks as an attacker can recover the whole frame from knowledge of either the odd or the even list. A  $2n$ -bits random key (KeyM) is used to split the  $2n$ -byte chunk randomly into two lists instead of the fixed odd-even pattern in the basic VEA. Thus, VEA also leads to increased key management issues.

Pure permutation or scrambling algorithms scramble bytes within a frame of MPEG stream by permutation. Adam J. Glagell demonstrates that pure permutation algorithm is vulnerable to known-plaintext attack, and hence its use should be carefully considered [slagell04??]. By comparing the ciphertext with the known frames, the adversary can recover the secret permutation list.

Chiaraluce et al. [5], Li et al. [6], Pareek et al. [11] and others propose a chaotic scheme for video encryption. Chaotic schemes are mostly based on encrypting image/ videos using chaotic maps. Logistic map is the simplest of them all and is popular choice to chaotic encryption scheme. However, simple cryptanalysis has been performed against these schemes.

**Fig. 2.4** Pre-compression encryption scheme proposed by Pazarci and Diplin [12]. The scrambler allows unauthorized user to have an arbitrarily degraded view of program, yet is totally transparent to MPEG-2 compression



### 2.3.3 Pre-compression Encryption Algorithm

Although it is possible to encrypt the video content before compression it has some serious limitations which are crucial for mobile devices:

1. Pre-compression encryption implies encrypting raw or uncompressed bits which will waste lot of computational resources.
2. Encryption output is generally a random bitstream with lack of redundancy, making compression operation highly inefficient for general case. For example, consider encrypting a HD video at bare resolution of 480p ( $852 \times 480$ ) with AES. It would require 2.3 Million AES cycles per second to encode (and to decode) that video on a mobile device (or any device)! Moreover, the compression performance will be mostly lost as the AES output bits will be nearly random with no possibility of lossless compression!

One known example is the work of Pazarci and Diplin [12]. The scrambler, shown in Fig. 2.4, is transparent to MPEG-2 compression. They encrypt the video in the RGB (red, green, blue) color space using four secret linear transformations before video coding. This scheme maintains the compression efficiency of the video codec but has been found unsafe against brute-force attacks.

### 2.3.4 Selective Encryption

The idea of selective encryption overlaps with post-compression approaches in some cases but it can also be applied during the compression process. A lot of research on integrating encryption with multimedia compression standards to reduce the overall computation cost is focused on using some form of selective encryption. For example, since most of the image energy in DCT domain is concentrated in the dc

coefficient, Tang [19] proposed a system that encrypts dc coefficients with DES and scrambles the ac coefficients using a block-based permutation. However, the energy concentration is often unrelated to the degree of intelligibility (Wu and Kuo [21]). It was proven that the semantic content of the image is almost unaffected by removing the dc information. Therefore, the security level of Tang's system is reduced to that of the block-based permutation, making it vulnerable to various attacks. Wu and Kuo show that even encrypting some ac coefficients does not solve the problem. Shi, Wang and Bhargava [15] had proposed to encrypt every sign of DCT coefficients, but that effort was also refuted by Wu and Kuo. Pommer and Uhl [13] present a wavelet based selective encryption approach by using wavelet packet based compression instead of pyramidal compression schemes. Header information of a wavelet packet transform based image coding scheme is protected as AES is used to encrypt the subband decomposition structure. Lian et al. [8] uses Exp-Golomb codes for the encryption operation. Cheng and Li [4] propose a DWT-based partial encryption scheme which encrypts only a part of compressed data. Only 13–27 % of the output from quadtree compression algorithms is encrypted for typical images. A good summary of efforts in selective or partial encryption of images can be found in Liu and Eskicioglu [9].

A syntax compliant encryption algorithm is proposed for H.264/AVC [3]. The authors allow any decoder to decode the encrypted video (although incorrectly) achieving up to 25–30 dB gains. Using the proposed method allows insertion of the encryption mechanism inside the video encoder, providing a secure transmission which does not alter the transmission process. The bits “selected for encryption” are chosen with respect to the considered video standard such that each of their encrypted configurations gives a non-desynchronized and fully standard compliant bitstream. This can in particular be done by encrypting only parts of the bitstream which have no or a negligible impact on evolution of the decoding process, and whose impact is consequently purely a visual one. For example, an encryption operation which leads to interpreting a given codeword instead of another of same size is suitable for such scheme. About 25 % of I-slices and 10–15 % of P-slices are encrypted. Since intracoded slices can represent 30–60 %, the encryption ratio is expected to be relatively high. The main drawback of this scheme is the lack of cryptographic security. Indeed, the security of the encrypted bitstream does not depend more on the AES cipher.

Lian et al. [7] use a mixture of these schemes for encrypting H.264 AVC bitstreams. During AVC encoding, such sensitive data as intra prediction mode, residue data and motion vector are encrypted partially. Among them, intra prediction mode is encrypted based on exp-Golomb entropy coding, the intra macroblocks DCs are encrypted based on context based adaptive variable length coding, and intra macroblocks ACs and the inter macroblocks MVDs are sign encrypted with a stream cipher followed with variable length coding. The encryption scheme is of high key sensitivity, which means that slight difference in the key causes great distortions in cipher video and that makes statistical or differential attack difficult. It is difficult to apply known-plaintext attack. In this encryption scheme, each slice is encrypted under the control of a 128 bit sub-key. Thus, for each slice, the brute-force

space is  $2^{128}$ ; for the whole video, the brute-force space is  $2^{256}$  (the user key is of 256 bit). According to the encryption scheme proposed here, both the texture information and the motion information are encrypted, making it difficult to recognize the texture and motion information in the video frames.

### ***2.3.5 Joint Video Compression and Encryption (JVCE) Approaches***

The main idea behind joint coding is to integrate encryption into compression operation by parameterization of the compression blocks, and (in general) not modifying the compressed bits. Two main compression blocks where these techniques have been applied are Wavelet Transform and Entropy Coding. We will present a brief summary of entropy coding-based approaches followed by a discussion of Wavelet Transform based approach proposed by the authors. Next, we will present the general rules of thumb in designing a new JVCE scheme.

**Advantages** JVCE approaches compression and encryption into a single operation making it feasible for mobile and embedded devices to ensure multimedia security with their low power budgets. By integrating compression and encryption operations into one, JVCE approaches reduce the latency of encryption operation which is useful for real-time video delivery. JVCE approaches typically do not change the compressed bit streams themselves but change the way compressed bitstream is obtained. This integration allows exploiting the hierarchical signal representation in a transform domain, as used by most image and video compression techniques, in order to provide the advanced functionalities required by many modern applications. The ISO/IEC JPEG 2000 Part 9 (JPSEC) standard is an example of how compression and security can coexist and take advantage of each other.

### ***2.3.6 Future of JVCE Schemes***

JVCE schemes have opened an entirely new paradigm of encryption without explicit-encryption of video content which gives them advantages in terms of computations, mobility, and friendliness to post-compression operations. However, many such schemes have been broken especially against known-plaintext attacks. To design an efficient encryption key for mobile applications, we propose the following directions: Development of JVCE algorithms for different video coding blocks and efficient integration into a common framework. An efficient integration will refute most of the cryptanalysis and the combined system will give a much greater degree of security than existing ciphers. Including some efficient scrambling operations into the design is meant to obfuscate input-output relationships at different levels.

## References

1. Agi, I., Gong, L.: An empirical study of secure mpeg video transmissions. In: Proceedings of the Symposium on Network and Distributed System Security, pp. 137–144. IEEE Press, New York (1996)
2. Baugher, M., McGrew, D., Naslund, M., Carrara, E., Norrman, K.: The secure real-time transport protocol (SRTP) (2004)
3. Bergeron, C., Lamy-Bergot, C.: Complaint selective encryption for h.264/avc video streams. In: IEEE 7th Workshop on Multimedia Signal Processing, pp. 1–4 (2005). doi:[10.1109/MMSP.2005.248641](https://doi.org/10.1109/MMSP.2005.248641)
4. Cheng, H., Li, X.: Partial encryption of compressed images and videos. *IEEE Trans. Signal Process.* **48**(8), 2439–2451 (2000). doi:[10.1109/78.852023](https://doi.org/10.1109/78.852023)
5. Chiaraluce, F., Ciccarelli, L., Gambi, E., Pierleoni, P., Reginelli, M.: A new chaotic algorithm for video encryption. *IEEE Trans. Consum. Electron.* **48**(4), 838–844 (2002)
6. Li, S., Zheng, X., Mou, X., Cai, Y.: Chaotic encryption scheme for real-time digital video. In: Real-Time Imaging VI. Proceedings of SPIE, vol. 4666, pp. 149–160 (2002)
7. Lian, S., Liu, Z., Ren, Z., Wang, H.: Secure advanced video coding based on selective encryption algorithms. *IEEE Trans. Consum. Electron.* **52**(2), 621–629 (2006)
8. Lian, S., Liu, Z., Ren, Z., Wang, H.: Commutative encryption and watermarking in video compression. *IEEE Trans. Circuits Syst. Video Technol.* **17**(6), 774–778 (2007)
9. Liu, X., Eskicioglu, A.M.: Selective encryption of multimedia content in distribution networks: challenges and new directions. In: Communications, Internet, and Information Technology (CIIT 2003), pp. 276–285 (2003)
10. Meyer, J., Gadgetast, F.: Security mechanisms for multimedia data with the example mpeg-1 video. Project Description of SECmpeg, Technical University of Berlin, Germany (1995)
11. Pareek, N.K., Patidar, V., Sud, K.K.: Image encryption using chaotic logistic map. *Image Vis. Comput.* **24**(9), 926–934 (2006)
12. Pazarci, M., Dipcin, V.: A mpeg2-transparent scrambling technique. *IEEE Trans. Consum. Electron.* **48**(2), 345–355 (2002)
13. Pommer, A., Uhl, A.: Selective encryption of wavelet-packet encoded image data: efficiency and security. *Multimed. Syst.* **9**(3), 279–287 (2003)
14. Qiao, L., Nahrstedt, K.: A new algorithm for mpeg video encryption. In: Proc. of First International Conference on Imaging Science System and Technology, pp. 21–29 (1997)
15. Shi, C., Wang, S.Y., Bhargava, B.: Mpeg video encryption in real-time using secret key cryptography. In: Proc. Int. Conf. Parallel and Distributed Processing Techniques and Applications (1999)
16. Slagell, A.J.: Known-plaintext attack against a permutation based video encryption algorithm (2004)
17. Spanos, G.A., Maples, T.B.: Performance study of a selective encryption scheme for the security of networked, real-time video. In: ICCCN, pp. 2–10. IEEE Comput. Soc., Los Alamitos (1995)
18. Stinson, D.R.: Cryptography: Theory and Practice. CRC Press, Boca Raton (2006)
19. Tang, L.: Methods for encrypting and decrypting mpeg video data efficiently. In: Proceedings of the Fourth ACM International Conference on Multimedia, pp. 219–229. ACM Press, New York (1997)
20. Wen, J., Luttrell, M., Severa, M.: Access control of standard video bitstreams. In: Proc. IEEE Intl. Conf. Media Future (2001)
21. Wu, C.-P., Kuo, C.-C.J.: Design of integrated multimedia compression and encryption systems. *IEEE Trans. Multimed.* **7**(5), 828–839 (2005). doi:[10.1109/TMM.2005.854469](https://doi.org/10.1109/TMM.2005.854469)
22. Wu, C.P., Kuo, C.C.J.: Efficient multimedia encryption via entropy codec design. In: Proceedings of SPIE, vol. 4314, p. 128 (2001)
23. Zeng, W., Lei, S.: Efficient frequency domain selective scrambling of digital video. *IEEE Trans. Multimed.* **5**(1), 118–129 (2003). doi:[10.1109/TMM.2003.808817](https://doi.org/10.1109/TMM.2003.808817)



<http://www.springer.com/978-1-4471-4458-8>

Embedded Multimedia Security Systems  
Algorithms and Architectures

Pande, A.; Zambreno, J.

2013, XVIII, 146 p., Hardcover

ISBN: 978-1-4471-4458-8