UNIVERSITÄT SALZBURG

A Survey of H.264 AVC/SVC Encryption

Thomas Stütz Andreas Uhl

Technical Report 2010-10 December 2010

Department of Computer Sciences

Jakob-Haringer-Straße 2
5020 Salzburg
Austria
www.cosy.sbg.ac.at

Technical Report Series

A Survey of H.264 AVC/SVC Encryption

Thomas Stütz and Andreas Uhl

***Abstract*—Video encryption has been heavily researched in the recent years. This survey summarizes the latest research results on video encryption with a special focus on applicability and on the most widely-deployed video format H.264 including its scalable extension SVC. The survey intends to give researchers and practitioners an analytic and critical overview of the state-of-the-art of video encryption narrowed down to its joint application with the H.264 standard suite and associated protocols (packaging / streaming) and processes (transcoding / watermarking).**

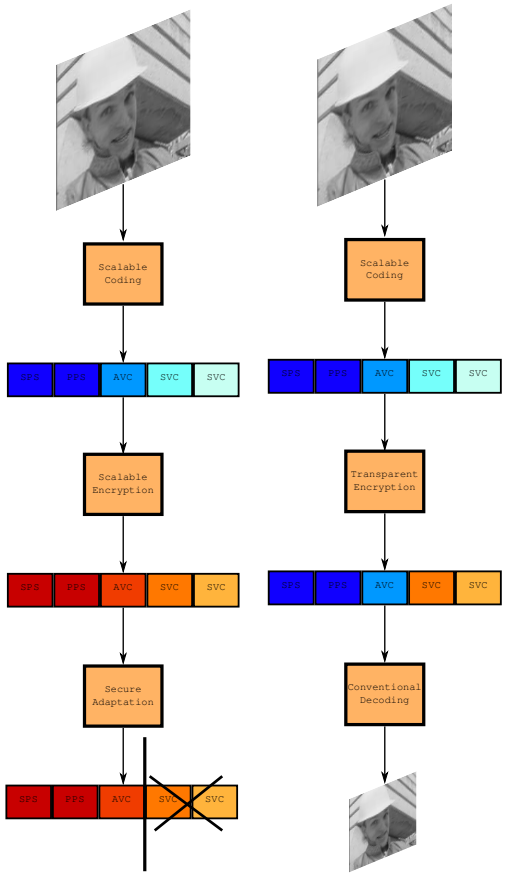
I. Introduction

H.264 is the most widely-deployed video compression system and has gained a dominance comparable only to JPEG for image compression. The H.264 standard has also been extended to allow scalable video coding (as specified in Annex G [27], referred to as SVC within this work) with a backwards compatible non-scalable base layer (non-scalable H.264 bitstreams referred to as AVC in this work). This extension enables the implementation of advanced application scenarios with H.264, such as scalable streaming and universal multimedia access [69]. Given the dominant application of H.264 as video compression system, the necessity of practical security tools for H.264 is unquestionable. In this survey we present an overview, classification and evaluation of the state-of-the-art of H.264 encryption, a topic to which numerous proposals that have been made. The survey focuses solely on H.264 AVC/SVC encryption and intends to give researchers a brief, yet comprehensive survey and to aid practitioners in the selection of H.264 encryption algorithms for their specific application context. Furthermore, the survey identifies the most relevant research questions in the area of video encryption, that still need to be answered in order to leverage the deployment of H.264 encryption.

A secure approach to encrypt H.264, also referred to as "naive" encryption approach, is to encrypt the entire compressed H.264 bitstream with a secure cipher, e.g., AES [49], in a secure mode, e.g., CBC (cipher block chaining mode). There are well-founded reasons not to stick to this approach, but to apply specifically designed encryption routines:

- The implementation of advanced application scenarios, such as secure adaptation, transparent / perceptual encryption and privacy preserving encryption.
- The preservation of properties and functionalities of the bitstream, such as format-compliance, scalability, streaming / packetization, fast forward, extraction of subsequences, transcodablity, watermarking, and error resilience.
- The reduction of computational complexity (especially in the context of mobile computing).

Secure adaptation requires a scalable bitstream and specific encryption routines that preserve the scalability in the encrypted domain (see figure 1(a)). Secure adaptation is the basis for secure scalable streaming [70], where secure adaptation is employed in a multimedia streaming scenario. A secure stream for a mobile phone (low bandwidth, low resolution display, low computing power) and a personal computer (high bandwidth, high resolution display, high computing power) can be generated from the same secure source stream (by secure adaptation) without the necessity of the secret key, thus enabling creator-to-consumer security. Transparent encryption denotes encryption schemes where a low quality can be decoded from the ciphertext; this functionality can be implemented with scalable bitstreams (see figure 1(b)) by encryption of the enhancement layers. Privacy preserving encryption should conceal the identify of persons, an exemplary implementation is shown in figure 2.



(a) Secure adaptation (b) Transparent encryption

Fig. 1. Secure adaptation and transparent encryption

The remainder of the paper is structured in the following way: H.264 is briefly summarized in section II. In section III application scenarios of video encryption are discussed and

Fig. 2. Privacy preserving encryption: DCT coefficients permutation (figure taken from [13], figure 2 (b), p.1171)

their corresponding different notions of security are motivated and defined. The application context also requires that the video encryption scheme preserves functionality of the video bitstream; details are discussed in this section as well, which ends with the presentation of our evaluation criteria for a video encryption scheme. In section IV H.264 compression and encryption are discussed jointly in detail. This approach of presentation was chosen to keep the level of redundancy low. Having discussed implementation and technical issues of H.264 video encryption schemes, section V proposes solutions and discusses the proposed schemes with respect to the security and application scenarios. Further research directions are discussed in section VI and finally we conclude in section VII.

II. Overview of H.264 AVC / SVC

The H.264 standard specifies the syntax and semantics of the bitstream together with a normative decoding process [27]. However, it is often and especially in the context of H.264 encryption more convenient to consider the encoding process. The raw video data is input to the encoder, the output is the bitstream in the NAL (network abstraction layer) format, i.e., a series of NAL units (see figure 3). The NAL units have a plaintext header indicating the type of data in AVC as shown in figure 4 in which the entire H.264 NAL header is outlined. The NAL header consists of the forbidden zero bit (F), a 3-bit field signalling importance of the NALU (NRI), and the NAL unit type (NUT). The most common NUTs are summarized in table I, NALUs with a unspecified NUT have to be discarded by the decoder.

These NAL units are commonly packaged in a container format for transmission and storage. A typical H.264 encoder has the structure as outlined in figure 6. Important parts are motion estimation (ME in figure 6) and motion compensation (MC in figure 6). Novelties in H.264 compared to previous video coding standards are intra prediction (Intra pred in figure 6) and in-loop deblocking filtering, i.e., reference pictures are filtered to reduce blocking artifacts prior to motion estimation and compensation. A 4x4 DCT-based transform is applied (T in figure 6), followed by quantization (Q in figure 6). There are two types of entropy encoding in H.264, namely CAVLC (context adaptive variable length coding) and CABAC (context adaptive binary arithmetic coding).

NUT	Description	AVC class	SVC class
0	Unspecified	Non-VCL	Non-VCL
1	Non-IDR slice	VCL	VCL
5	IDR slice	VCL	VCL
6	SEI	Non-VCL	Non-VCL
12	Filler data	Non-VCL	Non-VCL
14	Prefix NAL	Non-VCL	Variable
16 . . . 18	Reserved	Non-VCL	Non-VCL
20	SVC slice	Non-VCL	VCL
21 . . . 23	Reserved	Non-VCL	Non-VCL
24 . . . 31	Unspecified	Non-VCL	Non-VCL

TABLE I
Selected NAL unit types.

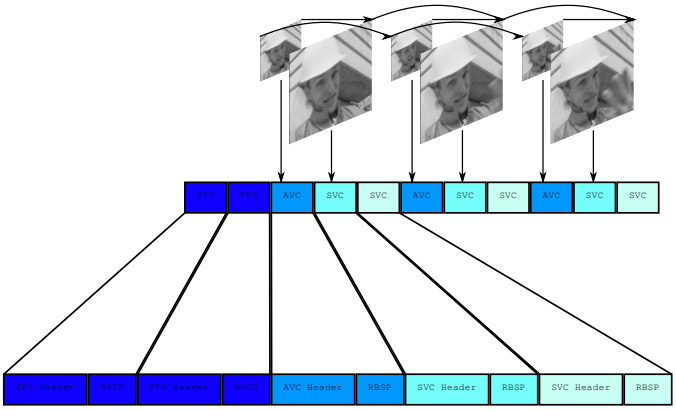


Fig. 3. A mapping of video data to H.264 SVC NALUs

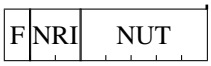


Fig. 4. NAL unit header structure.

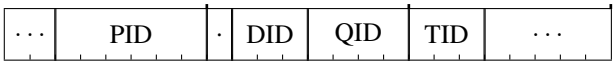


Fig. 5. NAL unit header SVC extension structure.

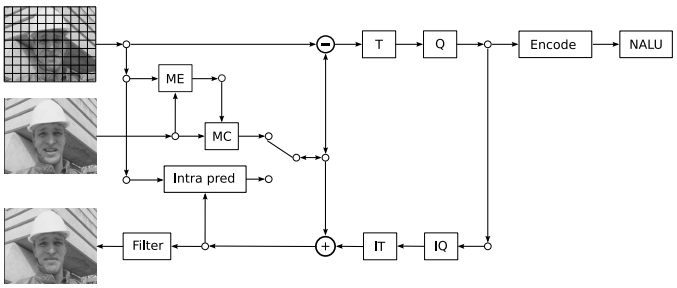


Fig. 6. H.264 compression overview

The scalable extension of H.264, referred to as SVC, employs most of the tools defined in the non-scalable H.264, referred to as AVC. An SVC bitstream consists of a base layer and enhancement layers; each enhancement layer improves the video in one of three "scalability dimensions", namely resolution, quality, and time. Therefore each scalable NALU belongs to a certain dependency layer (most commonly for resolution-scalability), a certain quality layer (to enable SNR-scalability), and a certain temporal layer (to enable different frame rates). The scalability information for SVC is contained in an SVC extension header succeeding the AVC NALU header, as shown in figure 5. Most important are the fields DID (dependency id), which indicates that the NALU belongs to a certain resolution, QID (quality id), which indicates that the NALU belongs to a certain quality and TID, which indicates that the NALU belongs to a certain temporal layer, i.e., commonly a certain frame rate. The value of the PID (priority id) is not standardized and can be used to enable very simple adaptation by only taking this field into account. An exemplary mapping between raw video data and NAL units is illustrated in figure 3. A resolution and quality scalable bitstream is shown, the higher resolution is coded with two quality layers. The NALU header of the base layer has a DID of 0, a QID of 0 and a TID of 0, while the NALU headers of the two enhancement layers have a DID of 1, a QID of 0 or 1 and a TID of 0.

The following references give further details on H.264 AVC [73], [52] and SVC [54]. Of course the standard itself should be considered as ultimate reference [27] for both formats.

III. Multimedia Encryption

The potential application scenarios for multimedia encryption are diverse and often require specific functionality of the video stream, e.g., scalability, to be preserved by the encryption scheme, such that associated processes, e.g., rate adaption can be conducted in the encrypted domain. The classic application scenario of video encryption is in DRM (digital rights management), more precisely copyright protection, in which content providers aim to secure their business value, i.e., they want to prevent uncompensated redistribution of their content, very frequently videos.

It is also common practise, that content providers, e.g., as frequently applied in pay-TV, offer free public access to parts of their content to attract potential customers. In the application scenario of transparent encryption (also referred to as perceptual encryption in literature [34], [41]) the availability of a public low quality version is a requirement and the threat is that an attacker is able to compute a reconstruction of the original content with higher quality then the available public version (see figure 1(b)).

Privacy preservation is also a concern in the context of video encryption, e.g., a commonly referred application is privacy preserving video surveillance [13]; here the privacy of the people and objects in the video should be preserved; analogous problems for still images is currently facing Google with its StreetView application. The security threat in privacy preserving surveillance is the identification of a human person or object, e.g., a license plate, in the video, which thus has to be prevented (see figure 2).

Video conferences are another prominent application scenario in which video data is encrypted [24].

The secure adaptation of compressed video streams to network conditions, often referred to as secure scalable streaming (SSS), is a frequently discussed application scenario in the context of video encryption [4], [3], [77], [5], [19].

Though not a distinct application scenario, mobile computing is often referred to in the context of multimedia encryption [24], as the lower performance of mobile devices imposes strict constraints on the computational complexity, which is an argument for low-complexity encryption approaches.

A. Security / Quality / Intelligibility

The security notions for video encryption are application-context dependent. E.g., in a commercial content-distribution scenario the security notion for video cryptosystems is different to the conventional cryptographic security notion for cryptosystems. While conventional cryptographic security notions are built upon the notion of message privacy (referred to as MP-security), i.e., nothing of the plaintext message can be learnt / computed from the ciphertext, the privacy of the message (video) is of limited concern for the content providers, but a redistribution of a sufficient quality version poses the threat to their business model. The security of the video cryptosystem has to be defined with respect to this threat, i.e., the reconstruction of a sufficient quality version on the basis of the ciphertext, which leads to a specific security notion for multimedia encryption [53], [45], [60], which we refer to as MQ-security (message quality security)[60] in this paper. The security requirements of many application scenarios in the context of video encryption can be pinned down to this definition: A video is encrypted and an adversary must not be capable to compute a reconstruction of the plaintext with higher quality than allowed in the application scenario. It is sufficient for DRM scenarios that the quality is severely reduced, such that a redistribution is prevented. In the context of privacy-preservation the quality / intelligibility of a video is measured in terms of recognizability of faces and persons [14]. The MQ-security notion has often been implicitly employed in literature and similar concepts have been put forward explicitly [53], [45].

It has to be highlighted that application scenarios require different quality levels to be protected, the leakage of an edge image might not be considered a security threat in a commercial content distribution scenario, but renders the application in a video conferencing scenario or a privacy-protection scenario impossible.

Although the privacy of the content is not an objective of the content provider in the commercial content-distribution scenario, the customers may have privacy concerns if the content, e.g., the movie being watched in a VoD scenario (video on demand), can be identified especially if the content is incriminated with social taboos. In the conventional cryptographic security notion, MP-security, no information about the plaintext has to be (efficiently) computable from the

ciphertext. If one considers the raw video data as plaintexts the preservation of any information, even the preservation of the length of the compressed video stream or the length of units that comprise the compressed video stream constitutes a security violation, as even the compressibility of the raw video data leaks information on the raw video data. If encryption is conducted after compression the compressibility information is contained in the length of the compressed video data. This security notion appropriately models the security requirements of highly confidential video communications, e.g., video conferences in politics and economy. Thus in this very strict interpretation of MP-security for video cryptosystems (the raw video data is the plaintext space), the length of the video data (or packets) must not depend on the raw video data itself. This has major implications for the compression settings and video packetization, e.g., in a video conferencing or VoD scenario. In order to ensure that absolutely no information about the visual content is computable from the transmitted encrypted and compressed video data, even the length of the packets must not depend on the raw video data [19]. This implies that for this interpretation of MP-security the video has to be transmitted at a source-independent rate (e.g., constant) and in a source-independent fashion (e.g., constant packet lengths). The issue that video streams can be identified is present even in "secure" state-of-the-art technology, e.g., in SRTP [19] and IPsec, as the secure encryption of packets does not conceal the packet lengths. This strict security notion, which we refer to as MPV-security, for video cryptosystems conflicts with rate-distortion optimal packetization and an optimal secure adaption, also referred to as secure scalable streaming (SSS) in the context of network adaptation [4], [3]. MPV-security does not at all model the security requirements and threats of many application scenarios at all, e.g., of perceptual / transparent encryption and privacy preserving video surveillance.

The preservation of format-compliance could be assumed to compromise security, recent contributions from the cryptographic community [6] discuss the topic in depth and define a concise formal framework and re-formulate the MP-security notion for format-preserving encryption (MP-security is defined for equal length format-compliant datums) and also analyze format-preserving encryption algorithms, which are proved to be secure. However, there still is a gap between the theoretic availability and the practical applicability of format-preserving and secure encryption algorithms [60] and this security notion is also not applicable for many application scenarios in the context of multimedia encryption.

Lightweight / Soft / Partial / Selective Encryption: Some contributions to multimedia encryption propose the application of less secure but more efficient encryption algorithms (soft encryption), i.e., the computational complexity to break the employed cryptosystem with respect to MP-security is limited. E.g., in [16] it is proposed to employ a weaker cipher (an AES derivative with fewer rounds) for the less important parts of the bitstream. Often obviously insecure algorithms are employed (e.g., adding constants to the coefficients [31]) which also fall into that category.

Another approach to reduce the computational complexity of encryption is selective / partial encryption of the bitstream with a secure cipher [46].

In this paper, we will discuss the schemes in a cipher-independent fashion, i.e., all encryption proposals will be considered to employ the same secure cipher as single source of pseudo-randomness.

B. Preserved Functionality

H.264 bitstreams are associated with functionalities, i.e., there are specified protocols and processes how to store, transmit, and process H.264 bitstreams (e.g., extract parts, adapt the rate, ...). Non-scalable and scalable H.264 bitstreams are accessible via a network abstraction layer (NAL), i.e., the coded video data is a sequence of separate NAL units (see figure 3 for a possible mapping of raw video data to NAL units).

H.264 bitstreams are embedded into container formats for transmission and storage [72], [25] and depending on the encoding settings bitstreams allow certain operations, such as extraction of IDR-picture (comparable to an I-frame in previous standards), extraction of a subset of the frames, cropping and in the case of SVC the extraction of sub-streams with different spatial resolution, temporal resolution and SNR-quality. A wide range of watermarking algorithms specifically tailored to H.264 have been proposed, e.g., [38], [50], [24], [80], [79], [47] and a joint application of encryption and watermarking, especially watermarking encrypted content, is often desirable [38], [10], [80], [79].

1) Format-compliance: A bitstream is denoted format-compliant or H.264-compliant, if it suffices the syntax's and semantics's requirements of the H.264 standard [27]. A format-compliant H.264 bitstream has to be accepted by every H.264-compliant decoder without any undefined decoder behaviour. It is necessary to distinguish whether a functionality is preserved format-compliantly, i.e., standard processing can be applied and no modified software is necessary. E.g., functionality is not preserved if the encrypted parts of the video are signalled as supplementary data, which has no semantics in the H.264 decoding process comparable to commentary in programming languages. The encrypted H.264 bitstreams where the encrypted data is signalled as supplementary data (e.g., using SEI messages, see table I) are still format-compliant, but the encrypted video data is treated completely different compared to plaintext video data. Thus the application of conventional tools to process the video data may lead to unexpected and undesired behaviour, e.g., rate adaptation algorithms may not perform optimal on the encrypted bitstreams. Thus we say that a functionality is preserved in a compliant fashion, if exactly the same processes as for an H.264 bitstream are applicable.

2) Packetization: NAL syntax / structure: The preservation of the NAL structure and syntax requirements enables the transparent application of standard container formats and tools for H.264.

3) Fast Forward / Extraction of Subsequences / Scalability: NAL semantics: The additional preservation of the NAL semantics, i.e., the NAL unit type (NUT, see figure 4) enables more sophisticated processing of the encrypted bitstream, such

as fast forward, e.g., to the 100^{th} IDR-picture of a coded video sequence, the extraction of subsequences, e.g., the last 10 minutes of a football match. In case of SVC this enables the preservation of scalability in the encrypted domain. The preservation of scalability in the encrypted domain allows to even adapt the encrypted video. In SVC the scalability information is contained in the NALU headers, this information has to be preserved in the encrypted domain. In the context of DRM-systems the preservation of scalability allows to adapt the encrypted content to diverse hardware platforms efficiently without the need for the encryption key, e.g., a single content representation for mobile devices and home cinema systems.

4) Transcodability: Even if the bitstreams are not coded with SVC, the bitstreams can be transcoded via coefficient requantization and it is beneficial if this property is preserved in the encrypted domain [66].

5) Robust Watermarking: Many watermarking algorithms are compression-format independent and require the raw video data as starting point. In the scope of this paper we only consider proposals that have been explicitly designed for the joint application with H.264. Furthermore we do not discuss the watermarking algorithms in detail, but limit the discussion to the marking space and whether embedding could be conducted in the encrypted domain. The embedder may be required to have precomputed metadata for embedding [80], [79]. We consider two basic approaches for watermarking compressed H.264 bitstreams, watermarking of DCT coefficients [38], [50] and watermarking via modifications to the intra-prediction modes [80], [79]. DCT coefficient watermarking requires a partial decoding, i.e., entropy decoding of the coefficients and afterwards the watermark is embedded via modifications of the coefficients. The approach of [50] requires access to the entire coefficients, while [38] proposes a quantization-based watermarking scheme, which preserves the signs of the coefficients and thus can be combined with coefficient sign encryption. The inter-prediction mode watermarking approach [80], [79] can be applied by simple substitutions of bits in the compressed bitstream. Thus two basic watermarking approaches for H.264 are considered in this work:

- DCT coefficient watermarking
- Stream substitution watermarking

C. Advanced Application and Security Scenarios

Apart from the goal of severely reducing the quality or even completely hiding all (visual) information, video encryption is also often designed to meet different application requirements. In the context of Pay-TV partial access schemes are commonly employed, e.g., frequently only the start of a video is made publicly available (to attract customers). Efficient integration into existing distribution frameworks can be achieved by format-compliant encryption schemes, as no modification of the existing infrastructure is required.

Transparent encryption can be considered a special case of partial access to the video, where only the low quality version can be accessed unconditionally. Transparent / perceptual encryption requires that a high quality version of the content shall be protected, while a low quality version shall be publicly available. Tunability of a perceptual encryption scheme, i.e., whether quality can be efficiently adjusted, is an important aspect. ROI encryption is useful for privacy preservation, i.e., the image areas in which persons and privacy sensitive objects appear have to be rendered unintelligible (encrypted). Privacy preserving encryption has gained significant interest [55], [12], [14] and our discussion in this work is deliberately only focused on the H.264 encryption details.

In conclusion we distinguish the following distinct security and application scenarios:

- Highest level security (MP security on the raw video data):
  E.g., source independent length packets and MP-secure encryption schemes, i.e., practically AES encryption in a secure mode, can meet the imposed requirements.
- Content confidentiality / visual semantic security (MQ security with a security metric that can identify a security breach for the visual data):
  E.g., encryption of source dependent length packets can meet these requirements. Content confidentiality lowers the security requirements, in order to allow functionality, such as optimal adaptation, to be preserved (which leads to a security breach for highest level security).
- Sufficient encryption (MQ security with a quality metric that can reliably determine the subjectively perceived quality):
  E.g., many encryption schemes may offer this property, but the computational hardness of the problem (to recover a higher quality then targeted) has not yet been proved for any proposal. However, for most encryption schemes no practical attacks exist. Sufficient encryption even more than content confidentiality lowers the security requirements and even more functionality can possibly be preserved, such as transcodablity and watermarking.
- Transparent / perceptual encryption (MQ security with a quality metric):
  Similar to sufficient encryption, the goal of transparent encryption is to reduce the quality, but transparent encryption also requires that a certain quality is preserved, i.e., that the ciphertext can be decoded with a certain quality in order to provide a public low quality version.
- ROI encryption / privacy preserving encryption (MQ security with an intelligibility metric):
  The security of a privacy preserving encryption scheme can be defined on the basis of intelligibility, i.e., does (automatic) face recognition work on the encrypted video [14].

D. Evaluation Criteria

Given a video encryption scheme suitable for a security and application scenario, security has to be assessed in terms of the computational complexity to break the scheme with respect to the appropriate security notion. A video encryption scheme can have a negative impact on the compression performance, which is a major evaluation criterion for the practical applicability of an approach.

Online / Offline Scenario: The computational complexity of an encryption scheme also is a decisive factor for its applicability. Depending on the application context, the video data is available in a raw, uncompressed format and H.264 compression has to be conducted anyway (this is referred to as online scenario [68], e.g., video conferencing) or the video data is already available as compressed H.264 bitstream (this is referred to as offline scenario [68]). This distinction is necessary for the assessment of the computational complexity of a video encryption scheme, as most compression-integrated schemes are only feasible in an online scenario (for complexity reasons), as they require computationally demanding parts of the compression pipeline to be performed.

The application context may further require that certain properties and associated functionalities of the video bitstream are preserved in the encrypted domain. Thus the main assessable properties of a video encryption scheme in a security and application scenario are:

- Security (in a certain application scenario with respect to the specific security notion)
- Compression efficiency
- Computational complexity
- Preserved functionality (format-compliance, packetization, scalability, transcodability, and watermarking)

IV. An Overview of H.264 [Encryption]

A. Encryption before Compression

If encryption is conducted before compression, the most important issue is the influence on H.264 compression performance. Thus if the entire visual information should be concealed, these approaches are not the method of choice, but if smaller areas need to be concealed, encryption before compression has been proposed. For privacy preservation the straight-forward solution would be to cut out the privacy-endangering areas and code them independently and encrypt them afterwards. Another solution is to encrypt image areas and encode the modified image, e.g., a permutation of positions of the pixels in the privacy-endangering areas is proposed in [9]. With respect to H.264 the following aspects need to be considered: Is decryption after lossy H.264 compression still possible (it is possible with pixel position permutations) and how can the influence on compression performance be minimized, which is more easily achieved when encryption is performed in the compression pipeline [12], [13], [67].

B. Compression [Integrated Encryption]

In H.264 a picture is processed in blocks, starting with 16x16 macroblocks, which can be further sub-divided in a hierarchical tree fashion down to 4x4 blocks. The macroblocks can be grouped in slices, but most commonly a slice consists of the macroblocks of an entire picture (default configuration in most encoders). In an IDR picture (instantaneous decode refresh picture, similar to I-frames in previous standards) only intra-coding is permitted and all previously decoded reference pictures will not be used in the further decoding process.

Index	Codeword	mb_qp_delta
0	1	1
1	01 0	1
2	01 1	-1
3	001 00	2
4	001 01	-2
5	001 10	3
6	001 11	-3
7	0001 000	4
8	0001 001	-4
...	...	...

TABLE II
Exponential Golomb code encryption

1) Intra-Prediction [Mode Encryption]: Intra-prediction may only take advantage of the previously coded data of the current slice. In intra-coding the pixel data of a block can be either predicted on the basis of previous block data (in raster scan order) or transmitted directly (I_PCM mode). The prediction mode has to be signalled in the bitstream, modification of the intra-prediction modes for encryption has been proposed in [1], [36], [7], [39], [37], [38], [40], [56], [62]. A visual example of a encryption of the intra-prediction modes is shown in figure 10(a). The encryption of the intra-prediction modes is very similar in all contributions. The intra-prediction modes are encoded with exponential Golomb codes, the format and length preserving encryption of exponential Golomb codes is illustrated in table II. Table II shows exponential Golomb code word and the associated value for the syntax element mb_qp_delta, which changes the quantization parameter on a macroblock basis. Exponential Golomb codes have a prefix, the leading zeros until the first one, which signals the length of the suffix. The suffix is an arbitrary bitstring of the signalled length. This suffix can be conventionally encrypted without compromising the format-compliance, i.e., the resulting ciphertexts are still valid exponential Golomb codes. The suffix are the bits contained in the boxes in table II.

The intra-prediction produces residual data (the difference to the intra-prediction), macroblock partition information, and intra-prediction mode information.

2) Inter-Prediction [Mode Encryption]: In inter-prediction blocks are predicted on the basis of previously decoded reference pictures. For that end motion estimation and compensation is conducted, a novelty of H.264 is the tree-structured motion estimation and compensation, which employs variable block sizes. Some inter frame macroblock subdivisions result in the same number of motion vectors. In [36] it is proposed to perform permutations on the set of inter frame macroblock subdivisions with the same number of motion vectors.

The results of inter-prediction are residual data (the difference to the inter-prediction), the macroblock partition information, the inter-prediction mode and the motion vector data.

3) Motion Vector [Difference Encryption]: Motion estimation and compensation works on a macroblock basis (16x16 blocks), which can be further decomposed to 4x4 blocks. For

each of the blocks of a macroblock a motion vector is calculated. The motion vector data are subject to further processing before entropy coding, i.e., motion vector prediction, which yields motion vector differences. The modification of motion vectors and motion vector difference data for encryption has been proposed frequently [36], [39], [30], [44], [74], [18], [28], [38], [42], [66], [37], [40], [62]. The schemes to modify the motion vectors and motion vector differences are diverse, e.g., many propose sign encryption [39], [74], [66], [62], while other encrypt the suffix of the exponential Golomb code [38], [40]. In [40] the encryption of the exponential Golomb codes is proposed as illustrated in table II.

In order to control the distortion (for perceptual / transparent encryption), it is proposed in [44], [18] to modify the motion vectors mv in the following way: $mv = mv + \mathrm{round}(\alpha * Z)$, where Z is uniformly distributed on [-1,1] and α is used to adjust the quality degradation (the larger α, the higher the distortion). A lightweight encryption scheme for motion vector data is proposed in [42], only half the motion vectors are securely encrypted in [36]. The motion vectors of a slice are permuted in [30].

4) [Secret] Transform: The residual data either obtained by intra or inter prediction is subject to transformation and quantization. All residual data is subject to 4x4 DCT-based transform. The chroma DC coefficients are further transformed with 2x2 Hadamard transform (in all macroblocks) and the luma DC coefficients are only further transformed with a 4x4 Hadamard transform in case of intra-prediction and 16x16 mode. The transform coefficients are subject to scalar quantization. It has been proposed to employ different 4x4 transforms with similar properties as the 4x4 DCT-based transform [76]. If the coefficients of these different 4x4 transforms are input to the standard inverse DCT transform, a reduced quality version is decoded as shown in figure 10(b).

In [57] it is proposed to encrypt the quantization parameter.

5) DCT Coefficient [Encryption]: Many proposals modify the signs of the coefficients [39], [74], [32], [28], [12], [13], [33], [37], [38], [40], [56], [67]. If CAVLC encoding is applied, the context-adaptive coding makes it complex to integrate format-compliant and length-preserving encryption of the coefficients other than sign encryption. In [37] it is proposed to encrypt the level of intra-macroblock DC coefficients as well. The proposal of [56] encrypts non-zero coefficients that have been mapped to exponential Golomb codes prior to CABAC coding, the encryption process is illustrated in figure 7, the obtained reconstruction is shown in figure 10(c). The non-zero coefficients NZ, are split into sign and level: the level part is encoded in two parts, smaller level values (up to a value of 14) are encoded only with a truncated unary binarization process [27, 9.3.2.2], while higher level values additionally require an exponential Golomb code, which are format-compliantly encrypted in [56] by only encrypting the suffix bits of the exponential Golomb code.

The proposals of [44], [18] encrypt only the least significant bits of a coefficient (for transparent encryption), which significantly reduces the compression efficiency. As a solution the authors propose to code these data independently and encrypt it separately.

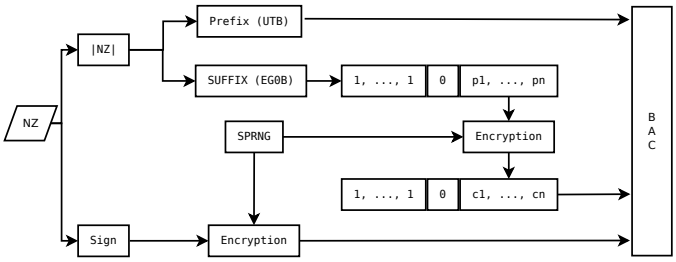


Fig. 7. CABAC coefficient encryption [56]

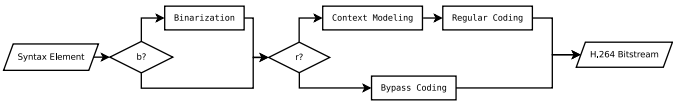


Fig. 8. H.264 CABAC

6) [Secret] Scan Orders: Prior to entropy coding in each 4x4 array of coefficients, the non-zero coefficients are mapped to a sequence by the zig-zag scan. A modification of the scan-order has been proposed in [12], [62], i.e., the instead of the zig-zag scan a random permutation is performed.

In H.264 two entropy coding modes are available, CAVLC (context adaptive variable length coding) and CABAC (context adaptive binary arithmetic coding).

7) [Joint Encryption and] CAVLC: In CAVLC all data but the residual data is encoded with fixed codewords (not context-adaptive). Only the residual data is encoded context-adaptively. Many of the codewords in CAVLC are coded with exponential Golomb codes, which can be encrypted format-compliantly (see table II). The exponential Golomb code encryption scheme is employed frequently [7], [39], [40]. In [40] the intra prediction modes and the motion vector differences are encrypted in this fashion. In [48] the code words of the syntax element run_before are permuted in an almost length-preserving fashion, namely the entries of the code word table are split into two partitions and within these partitions the codewords are randomly chosen.

8) [Joint Encryption and] CABAC: CABAC is employed for coding slice data and macroblock data [27, sect. 7]. Syntax elements are mapped to binary codes, e.g., exponential Golomb codes, in the binarization process prior to entropy encoding (see figure 8). For a minimal impact on the compression performance, only bits encoded in bypass mode should be encrypted (see figure 8), these include the suffix of the exponential Golomb coded MVD and coefficient levels. In the case of the offline scenario (the compressed video bitstream is already available, see section III-D) it is necessary to arithmetically decode the binarized syntax elements (MVD and coefficient levels) and encrypt the suffixes, in case of online scenario (the raw video is available, see section III-D) this approach can be implemented in the compression stage. Performing format-compliant encryption / bit replacement directly on the compressed bitstream is extremely complex (practically infeasible) as internal states of the coder have to be preserved [79], otherwise the remaining data is interpreted falsely which may easily lead to format violations.

Compression-oriented encryption schemes can be combined (see table III for possible combinations) and figure 10(d) for a visual example of the quality reduction for a combination of MVD, DCT coefficient and inter-PM encryption.

C. Bitstream [Oriented Encryption]

The output of H.264 encoding is a bitstream (coded video sequence) which is given as a sequence of NALUs (network abstraction layer units).

1) NALU [Encryption]: The fully format-compliant encryption of the NALUs with simple encryption algorithms, e.g., comparable to packet body encryption in JPEG2000 [15], is not possible due to the violation of syntactical and semantical requirements. In JPEG2000, the encryption of packet body data is possible as only fractional bitplane data is coded, i.e., every decoded bit sequence can be interpreted as fractional bitplane data, while H.264 arithmetically encodes a multitude of syntax elements, not all syntax element values are always possible. The decoding of a ill-suited syntax element value (by decoding encrypted data) leads to violations of semantical requirements after arithmetic decoding, e.g., a value out of range.

However, given the appropriate encryption schemes which avoid the generation of marker sequences [19], regular NALU processing can be conducted on the compressed and encrypted bitstream. The preservation of the NAL structure and syntax requirements enable the transparent application of packaging methods [59], [19], which is a prerequisite for efficient transmission. The encryption of NALUs and preservation of the NALU header has been proposed in [81], [5], [19]. In [59], [19] it is highlighted that encrypted data can be transparently and efficiently signalled by the application of unspecified NALU types (NUTs) for encrypted NALUs, which then have to be ignored by a H.264 decoder [27, sect. 7.4.1].

2) Container-Formats: Several proposals [24], [23], [22] employ container formats and encrypt the H.264 bitstream or fractions of the H.264 bitstream with conventional algorithms, e.g., AES in counter mode. The basic setup is illustrated in figure 9. In the compression process, additionally an XML description of the bitstream is produced. This XML description can be used to adapt the bitstream (see [29] for an evaluation for H.264). The same description can also be used to efficiently identify parts of the bitstream suitable for encryption.

There is a standard for secure streaming of RTP data, including H.264:AVC/SVC [21]. If container formats are employed functionality can be preserved through explicit signalling in the container format, e.g., all NAL based functionality can be preserved.

3) Partial / Selective Encryption: Partial encryption of H.264 bitstreams can reduce the amount of data to encrypt and can also lead to performance improvements in a streaming system [19]: results are presented for base layer encryption and IDR picture encryption (of an SVC bitstream), which corresponds to the encryption of a certain subset of the NALUs of the bitstream. Figure 10(e) shows a possible reconstruction if IDR picture encryption is applied, we have replaced the IDR picture by a picture of zero values (a replacement attack) and

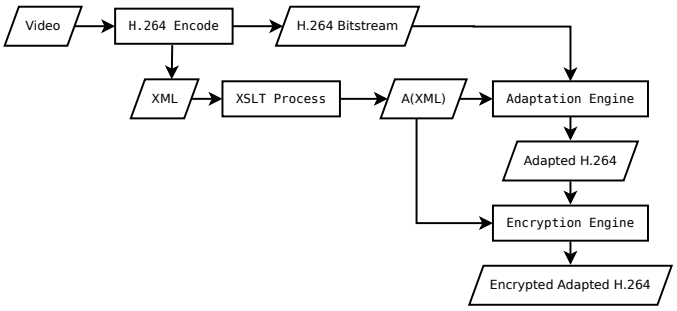


Fig. 9. MPEG-21 encryption and adaptation

figure 10(f) and 10(g) show the result of a replacement attack against base layer encryption (we have replaced the base layer picture by a picture of zero values). Another approach is to partially encrypt NALUs [24], i.e., only parts of a NALU are encrypted. The encryption of the leading fraction of a NALU renders the entire NALU not decodeable by standard decoders, which could be employed to effectively implement content confidentiality. The idea of selective encryption could also be applied to container format data, e.g., it has been proposed to only encrypt parts of the RTP header [2]. In [61] this RTP header encryption scheme, which is also applicable to H.264 RTP transmission, is analyzed. It is concluded that RTP header encryption of RTP streams containing H.264 data is insufficient to ensure security (with respect to all discussed security notions, i.e., the entire video can be reconstructed).

D. An Overview of SVC [Encryption]

1) Encryption before Compression: There are no dedicated encryption proposals that take SVC-specifics into account.

2) Compression [Integrated Encryption]: The base layer is encoded similar to AVC, thus all encryption schemes for AVC can be basically employed in the base layer. The enhancement layers can employ inter-layer prediction, but not necessarily have to, e.g., if inter-layer prediction does not result in better compression. The compression integrated encryption approaches for AVC can be applied as well for SVC, e.g., the approaches targeting the coefficient data can also be applied for SVC.

3) Bitstream [Oriented Encryption]: The approach of [59] takes advantage of SVC to implement transparent encryption after compression. The following approaches have been proposed for SVC encryption [3], [5], [19], [65], which all preserve the NALU structure and encrypt almost all of the NALU payload. As the NALU structure is preserved, scalability is preserved in the encrypted domain.

V. Solutions, Analysis and Discussion

In the following we discuss suitable encryption algorithms for the distinct security and application scenarios on the basis of our presented evaluation criteria (see section III-D).

A. Highest Level Security

This application scenario is hardly discussed in literature and none of the discussed approaches is capable to meet its requirements.

(a) Inter-prediction mode encryption: IDR picture (figure taken from [1], figure 1(a), p.392)

(b) Secret DCT transforms (figure taken from [76], figure 6, p.896)

(c) Sign encryption and coefficient encryption (CABAC) (figure taken from [56], figure 5(c), p.1040)

(d) MVD, DCT and Inter-PM encryption (figure taken from [39], figure 4, p.286)

(e) IDR picture encryption (a reconstruction of a P picture is shown)

(f) SVC base layer encryption (replacement attack)

(g) SVC base layer encryption (replacement attack and level adjustment)

Fig. 10. Visual examples of H.264 encryption

1) Suitable Video Encryption Schemes and Proposed Solution: Video compression and video encryption schemes can be designed to meet the requirements of highest level security and preserve scalability. The preserved (scalability) information must not depend on the raw data.

Solutions can target the encoding process, i.e., compress the data to certain source-independent bitrate and / or stuff bytes afterwards.

A practical solution for SVC defines target bitrates for the base and enhancement layers, and packetizes base and enhancement layers into fixed length packets and additionally applies byte stuffing if there is too few video data.

2) Security: The proposed solution (see section V-A1) is as secure as the employed encryption primitive, e.g., AES in counter mode, with respect to MP security on the raw video data.

3) Compression: Depending on the packetization strategy there can be a negligible to significant negative influence on the compression performance.

4) Performance: Encryption is applied on the compressed bitstream. Thus runtime performance is as good as the runtime performance of the applied encryption primitive, which is very good in the case of AES.

5) Preserved Functionality: Format-compliance can not be preserved, optimal packetization leads to a violation of the MP security notion on raw video data. Thus there is a trade-off between frame structure or scalability preservation and compression complexity and efficiency (i.e., it depends on the packetization strategy which needs to packetize the video bitstream in a source independent fashion, which requires special care in the compression and afterwards requires stuffing of the packets to source independent lengths). Transcoding and DCT watermarking can not be conducted in the encrypted domain, but stream substitution watermarking is still possible. For stream substitution watermarking we propose the application of a stream cipher mode with no cipher feedback, e.g., counter mode. The substitution watermark in the encrypted domain w_e is then $w_e = w_p \oplus k$, where w_p is the plaintext watermark and k are the corresponding key stream bits.

B. Content Confidentiality

To securely implement content confidentiality it is necessary to encrypt most of the video data or to prevent (partial) decoding at all.

1) Suitable Video Encryption Schemes and Proposed Solutions: For this application scenario we propose to encrypt the NALU payload, because each of the compression-integrated encryption schemes leaks some visual information. A combination of compression-integrated schemes heavily distorts the videos (see figure 10) but a human observer can guess the content of the video. Furthermore, sophisticated attacks are likely to reveal even more of the visual content.

The encryption algorithm can preserve the NALU syntax and semantics or format-compliance. If the semantics are preserved (i.e., the NUT is preserved) the format-compliance is lost, as the decoding of the encrypted data will lead to violations of the semantical requirements. If the format-compliance through explicit signalling of the encrypted data via unspecified NUTs is preserved the semantics are not compliantly preserved, as unspecified NUT only indicates that a decoder has to ignore the NALU data. Another approach is the application of container formats, which more clearly separate encryption and compression, but introduce certain overheads in terms of runtime and compression efficiency (e.g., results for adaptation for a NAL-based system and a MPEG-21-based system are presented [29], that can serve as reference for the expected overhead if container formats are employed). In both cases (NAL and container format) partial encryption could be applied, e.g., as proposed in [24].

Partial encryption schemes that encrypt a subset of NALUs are not applicable for content confidentiality as edge images can be reconstructed (see figures 10(e), 10(f), and 10(g)). Partial encryption schemes that encrypt the start of NALUs may offer some security with respect to content confidentiality if parts of the slice data are encrypted as well, as data is coded context adaptively (both for CAVLC and CABAC). In case of CAVLC only the coefficient data is coded context-adaptively, and thus at least some coded coefficient data has to be encrypted. For CAVLC the security relies on the uncertainty of the number of non-zero coefficients in neighbouring blocks, at most 17x17 combinations (0 to 16 non-zero coefficients) have to be considered for single 4x4 DCT transformed residual block, not an easy, but solvable task, using back tracking algorithms that try to correctly decode the partial coefficient data. For CABAC the security relies on the uncertainty of the state of arithmetic decoding, which means that the current state in decoding (which syntax elements to decode) and the state of each of the approximately 400 contexts [52] have to be guessed, as well as the state of the arithmetic decoding variables codIRange and codIOffset (both represented with 16 bit) [27, sect. 9.3.1.2]. Overall a very complex problem an attacker has to solve in order to obtain a partial decoding of the slice data. Thus, although there is no formal proof of the security of this schemes with respect to content confidentiality, a potential adversary is assumed to have a hard time to decode partially encrypted CABAC encoded video data.

2) Security: The preservation of the NALU structure and semantics is considered no security threat, as these data are insufficient to reconstruct the actual content (visually intelligible). However, an adversary can exploit this source-video dependent length data, which may pose a security threat in practical application (highly confidential video conferences).

3) Compression: There is no or just a negligible influence (due to encryption meta-information, such as cipher algorithm, mode and initialization vectors) on the compression performance.

4) Complexity: The cost of securely encrypting the video data is added, i.e., AES encryption which compared to compression and decompression is small. If NAL compliant encryption is applied a nearly negligible effort is necessary for syntax checks.

5) Preserved functionality: Format-compliance can only be preserved if the semantics of the NALU are changed, i.e., the NUT of the encrypted data has to be set to a value that the decoder ignores the encrypted data. However, if compliant adaptation is a goal the NALU header must not be changed. Thus both compliant adaptation and format-compliant encryption can not be achieved at the same time. NALU encryption with both schemes preserves the packetization, fast forward, subsequence extraction and scalability. Transcodability and DCT watermarking can not be conducted, but the schemes can be combined with substitution watermarking.

C. Sufficient Encryption

Most proposed schemes are suitable for sufficient encryption of H.264, the relaxed security requirements make it possible to employ encryption schemes, that offer functionalities that would violate the security requirements of other security and application scenarios. Contrary to transparent encryption there is no minimal quality requirement for the cipher video, which makes sufficient encryption easier to implement.

1) Suitable Video Encryption Schemes and Proposed Solutions: For sufficient encryption, a combination of compression-integrated encryption schemes is recommended, however, only if additional functionality weights out the disadvantages of these schemes in terms of security and performance. Also partial encryption schemes that encrypt a subset of NALUs (IDR, base layer) can be employed for sufficient encryption. In case of SVC we propose the encryption of the base layer (see figures 10(f) and 10(g) for replacement attacks, i.e., we have replaced the encrypted picture data by pictures containing only zero values). If format-compliance is desired, the base layer can be replaced by a uniformly grey video sequence (negligible compression performance deficits).

Partial encryption of NALUs is an option for sufficient encryption if format-compliance and functionality below the NAL, such as transcodability and DCT watermarking are not targeted. In the case of CAVLC partial decoding is a more realistic threat then in the case of CABAC and thus the application of CABAC and the encryption of the leading fraction of the NALUs including several bits ($>>128$) of the arithmetically coded data is proposed. In the case of CABAC the remaining fraction is hard to decode as all the internal states of the CABAC engine have to be known.

2) Security: Security proofs for sufficient encryption are not found in literature, but successful attacks against previously proposed schemes are reported [53]. Due to the application of unreliable quality metrics for low quality visual data, e.g., PSNR (peak-signal noise ration) does not perform well for that purpose [58], [20], the result of the attacks remain rather incomparable.

3) Compression: Compression-integrated encryption often is associated with a severe reduction of compression performance, however, many of the proposed schemes for H.264 perform very well (see table III for an overview).

In CAVLC, Intra-PM and MVD are encoded with exponential Golomb codes, which can be encrypted in a length preserving fashion. Coefficient sign encryption and CAVLC work well together, in [39] no decrease of compression ratio is reported, while a very small decrease of compression ratio is reported (a relative file size increase of 1% for higher rates 8% for lower rates are reported in [12]). Coefficient permutation is more expensive (up to 4% for lower rates and 11% for higher rates).

4) Complexity: In case of an online scenario (compression is conducted anyways) the complexity of compression-integrated encryption schemes is almost negligible compared to compression and decompression.

In case of an offline scenario the compression-integrated encryption schemes require that computationally expensive parts, such as binary arithmetic decoding have to be performed. Given the relative small cost of encryption compared to compression and decompression these approaches have significant disadvantages compared to bitstream-based schemes.

5) Preserved Functionality: All functionality can be preserved with the appropriate encryption scheme. DCT coefficient watermarking can be conducted if the access to the coefficient data is possible (see table III).

D. Transparent Encryption

The main additional requirement of transparent encryption is quality control.

1) Suitable Video Encryption Schemes and Proposed Solutions: Though many schemes have been proposed under the label perceptual encryption, quality control of the encrypted data is only discussed in a few contributions [44], [43]. In case of AVC, a transparent encryption approach has been proposed in [44], employing restricted MVD encryption and the encryption of less important bitplanes of DCT coefficients.

Furthermore previous DCT sign and coefficient encryption proposals can be extended by an explicit quality control, which controls the quality by restricting the sign (CAVLC and CABAC) and magnitude (CABAC) encryption to certain coefficients and additionally only the magnitude could be encrypted.

In [59] it is proposed to employ SVC for transparent encryption and if format-compliance is targeted to force a decoder to ignore the encrypted data by signalling in the NAL (unspecified NUTs). If the application of SVC is possible the encryption of the enhancement layers is the recommended solution (see figure 1(b) for an illustration of the approach).

2) Security: Security of transparent encryption schemes relies on the inability of an adversary to compute higher quality versions than already made public. Thus specifically tailored algorithms, inspired by super-resolution and denoising algorithms, are the main threat. Additionally, the preserved information in the ciphertext may be exploited. However, currently there are no known attacks against the recommended schemes and in case of the SVC-based transparent encryption algorithm the existence of such an efficient algorithm would also give efficient quality enhancement tools for ordinary AVC streams.

3) Compression: There is no compression overhead for the SVC-based encryption scheme.

In [44] only slight bitrate increases of less then 1% are reported.

4) Complexity: In case of the SVC encryption approach the qualities of the substreams have to be determined in the SVC compression process, which highly depends on the desired scalability properties of the SVC bitstream and is more complex then AVC encoding. If an SVC bitstream is available the scheme is efficient.

Only a small increase of complexity (similar to other compression-integrated schemes) is present in the online scenario. However, in an offline scenario costly parts of the decompression pipeline have to be performed.

5) Preserved Functionality: Commonly format-compliance is considered a necessity for the transparent encryption scenario and thus has to preserved. The proposed DCT watermarking schemes can not be applied, stream substitution watermarking can still be applied.

E. ROI Encryption

ROI encryption has been primarily proposed for privacy preserving encryption schemes.

1) Suitable Video Encryption Schemes: In [12], [14] sign encryption and permutations is proposed and reported to be meet the security constraints [14].

2) Security: According to [14] sign encryption and permutation prevents automatic face recognition and this is their proposed security metric for privacy preserving encryption. The goal of an adversary for this security notion is the development of a face recognition system that can identify faces even when encrypted. An adversary will try to combine attacks against the video encryption scheme and improved face recognition systems, e.g., permutations are known to be susceptible to known plaintext attacks [35].

3) Compression: Only small decreases in compression complexity are reported.

4) Complexity: As the privacy-threatening regions (faces) have to be detected, which is done on the raw video data on can assume an online scenario. Thus the impact of the overall system complexity is small.

5) Preserved Functionality: An important feature for ROI encryption is that the remaining video can be decoded in sufficient quality, such that privacy-preserving surveillance is possible. The recommended schemes have this property and are also format-compliant.

F. Discussion

The current state-of-the-art in H.264 video encryption can offer solutions for all of the security and application scenarios, content confidentiality and sufficient encryption only make sense if additional functionality, such as transcodability or watermarking, is preserved. H.264 encryption schemes are capable to preserve diverse functionality, but naturally at some cost in terms of security, runtime performance and compression performance. Table III summarizes important aspects and properties of the diverse encryption algorithms. Naive denotes AES in cipher feedback mode, MPV denotes an encryption algorithm that is MP-secure on the video data, CF denotes schemes that employ container formats, FC denotes schemes that NAL-compliantly encrypt NALUs and signal the encrypted data in the H.264 syntax, NC denotes schemes that NAL-compliantly encrypt NALUs but do not signal the encrypted data (semantics are preserved), S denotes DCT coefficient sign encryption, L denotes DCT coefficient level encryption (only applicable with CABAC), SDCT denotes secret DCT transforms, MVD denotes motion vector difference encryption, SSO denotes secret scan orders, Inter denotes inter prediction mode encryption and Intra denotes intra prediction mode encryption. The first rows identify the suitable encryption schemes for a security and application scenario. Additionally the table identifies whether schemes can be combined and whether they are format-compliant. The row labelled “Compliant packetization” indicates that conventional packaging tools and protocols can be employed. “Compliant adaptation” (for SVC) identifies schemes that allow conventional H.264 SVC adaptation, while “Adaptation”

	Naive	MPV	CF	FC	NC	S	L	SDCT	MVD	SSO	Inter	Intra
Highest level security	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Content confidentiality	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗
Sufficient encryption	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Transparent encryption	✗	✗	✗	SVC	SVC	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$
FC transparent encryption	✗	✗	✗	SVC	✗	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$
ROI encryption	✗	✗	✗	✗	✓	✓	✓	✓	✗	✓	✗	✓
Combinable	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓
Format-compliant	✗	✗	✗	✓	✗	✓	✓	✓	✓	✓	✓	✓
Compliant packetization	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
Compliant adaptation	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Adaptation	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Transcodability	✗	✗	✗	✗	✗	✓	✗	✓	✓	✗	✓	✗
DCT Watermarking	✗	✗	✗	✗	✗	✓	✗	✓	✓	$\sim$	✓	✓
BSS Watermarking	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Complexity (online)	bpp	$bpp + \epsilon$	bpp	bpp	bpp	< 1	$\approx bpp$	1	< 1	≈ 1	< 1	< 1
Add. Comp. (offline)	0	0	0	0	0	+H	+H	+H	+H	+H	+H	+H
Compression pres.	✓	$\sim$	$\sim$	✓	✓	$\sim$	✓	✓	$\sim$	$\sim$	✓	✓
Error propagation	H.264-Full	H.264	H.264	H.264	H.264	H.264	H.264	H.264	H.264	H.264	H.264	H.264

TABLE III
OVERVIEW OF H.264 ENCRYPTION APPROACHES

indicates schemes which require changes in the adaptation tools. Note that format-compliance and compliant adaptation can not be met simultaneously with bitstream oriented approaches. "Transcodability" refers to H.264 transcoding by requantization of the coefficients [64]. The encryption schemes are evaluated for their interoperability with watermarking systems, namely DCT watermarking and BSS (bitstream stream substitution) watermarking. Only some compression-integrated schemes can be combined with DCT watermarking. All schemes that employ encryption with additive ciphers (and no cipher feedback mechanisms) can be used with BSS watermarking. The online complexity is given in necessary pseudo-random bits per input pixel; bpp denotes the bit per pixel after compression. The offline complexity of bitstream-oriented schemes is almost the same as the online complexity, compression-integrated schemes, however, have to perform parts of the decompression / compression pipeline, which is indicated by the entry "+H". The preservation of compression efficiency is evaluated in the row labelled "Compression pres.". Error propagation is also analyzed, only the Naive scheme is susceptible to propagate errors, however, a bit error in the CBC mode corrupts only two blocks (with the size of the blocks of the employed block cipher). Thus even for the Naive scheme error propagation is very limited for common cipher / mode choices and only if exotic modes, such as PCBC (propagating cipher block chaining) are employed the entire bitstream will be corrupted. However, all schemes, as presented here, are sensitive to loss of synchronization, as a secure PRNG is employed. Packet loss is a realistic threat in the case of UDP-based RTP transmission; the straight-forward solution of repeatedly sending synchronization information, i.e., IV for the secure PRNG (e.g., AES in counter mode), has been shown to only have a very small impact on compression efficiency [19].

Many of H.264 encryption approaches have been proposed without an analysis and discussion of the functionality they preserve and of their interplay with other protocols and operations defined for H.264. The survey bridges this gap and provide an overview especially considering the aspect of preserved and additional functionality of H.264 encryption approaches.

Considering that most of the proposed H.264 encryption schemes leak visual information, we have to state the lack of sufficiently evaluated objective quality / intelligibility / security metrics to assess the amount of leaked visible information. Although a few proposals for quality / intelligibility / security metrics have been made [45], [63], none actually evaluates the performance of the proposed metrics with respect to the correlation to human perceived quality / intelligibility. Currently there are no suitable and evaluated metrics available and thus we have deliberately omitted results in this survey ([15, figure 9] shows an example where the security metric ESS indicates no visual information leakage, while an edge image is clearly visible). Given that many encryption schemes are brought forward with the claim that they severely reduce quality and intelligibility (their security is based on this quality / intelligibility reduction) the algorithmic inassessibility of this claim is certainly of great discontent. The lack of appropriate assessment tools for video encryption may also be partly responsible for the lack of schemes where the quality can be controlled. Apart from a some contributions [44], [17], [59], few approaches offer the ajustability of visual quality.

VI. OUTLOOK AND FURTHER RESEARCH DIRECTIONS

The most apparent deficit in the current research is that, although security in the context of video encryption is defined with respect to quality and intelligibility, neither quality nor intelligibility can be assessed. The lack of objective assessment methods makes video encryption schemes incomparable; the analysis on the basis of a visual inspection of single frames,

as it is the current state-of-the-art can hardly be considered satisfactory in a scientific context. Thus further research should focus on the development of objective metrics for the assessment of the security of video encryption schemes for the different security and application scenarios. A prerequisite for the development of novel objective quality / security metrics are subjective tests, in which the actually perceived quality and intelligibility is determined by human observers. Contributions to this line of research have already been made for transparent JPEG2000 encryption [58]. Objective assessment on the basis of face recognition rates has been proposed for the analysis of privacy preserving encryption [14]. For the assessment of video encryption schemes in the application scenarios of transparent encryption and sufficient encryption state-of-the-art objective quality metrics may be suited (however, this has to be backed up by empirical evidence, i.e., subjective quality evaluation tests [58]). For content confidentiality, however, novel intelligibility metrics as well as an evaluation framework for these metrics are needed (again subjective tests will have to be an integral part).

Further efforts in the area of H.264 encryption should also consider the standardization of security tools within H.264. Bitstream-oriented encryption, as well as other security features, such as authentication / message integrity and scalable authentication / message integrity, could be optimally integrated into the existing H.264 framework, as due to the well-designed NAL abstraction a backwards-compatible integration would be possible. This can be done by using previously reserved NAL unit types for signalling security related data, such as encrypted NAL units. The security extension could even be implemented without the definition of a novel file format, as has been necessary in JPEG2000 [26], i.e., the secured H.264 bitstreams could be completely backwards compatible to current H.264 bitstreams.

VII. Conclusion

In this survey we have presented, evaluated and discussed video encryption schemes for H.264. The choice of a video encryption scheme depends on the application-context, what are the security threats in this scenario and which functionality of the bitstream and video data has to be preserved in the encrypted domain. A focus of this survey has been the interoperability of video encryption with existing processes for the video data, such as packetization, (scalable) streaming, rate adaptation, frame extraction, fast forward and watermarking. The diverse contributions cover a wide range of application scenarios and this survey provides a guide to find the appropriate H.264 encryption scheme for a target application.

References

[1] Jinhaeng Ahn, Hiuk Jae Shim, Byeungwoo Jeon, and Inchoon Choi. Digital video scrambling method using intra prediction mode. In *Proceedings of Advances in Multimedia Information Processing, PCM '04*, volume 3333 of *Lecture Notes in Computer Science*, pages 386–393, Tokyo, Japan, December 2004. Springer-Verlag.

[2] Fadi Almasalha, Nikita Agarwal, and Ashfaq Khokhar. Secure multimedia transmission over RTP. In *Proceedings of the Eighth IEEE International Symposium on Multimedia (ISM'08)*, Berkeley, CA, USA, December 2008. IEEE Computer Society.

[3] J. Apostolopoulos. Architectural principles for secure streaming & secure adaptation in the developing scalable video coding (SVC) standard. In *Proceedings of the IEEE International Conference on Image Processing, ICIP '06*, pages 729–732, October 2006.

[4] John Apostolopoulos. Secure media streaming & secure adaptation for non-scalable video. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'04)*, volume 3, pages 1522–4880. IEEE, October 2004.

[5] H. Kodikara Arachchi, X. Perramon, S. Dogan, and A.M. Kondoz. Adaptation-aware encryption of scalable H.264/AVC video for content security. *Signal Processing: Image Communication*, 24(6):468–483, 2009. Scalable Coded Media beyond Compression.

[6] M. Bellare, T. Ristenpart, P. Rogaway, and T. Stegers. Format-preserving encryption. In *Proceedings of Selected Areas in Cryptography, SAC '09*, volume 5867, pages 295–312, Calgary, Canada, August 2009. Springer-Verlag.

[7] Cyril Bergeron and Catherine Lamy-Bergor. Compliant selective encryption for H.264/AVC video streams. In *Proceedings of the IEEE Workshop on Multimedia Signal Processing, MMSP'05*, pages 1–4, October 2005.

[8] Bharat Bhargava, Changgui Shi, and Sheng-Yih Wang. MPEG video encryption algorithms. *Multimedia Tools and Applications*, 24(1):57–79, September 2004.

[9] P. Carrillo, H. Kalva, and S. Magliveras. Compression independent object encryption for ensuring privacy in video surveillance. In *Proceedings of International Conference on Multimedia & Expo, ICME '08*, pages 273–276. IEEE, June 2008.

[10] M. U. Celik, A. N. Lemma, S. Katzenbeisser, and M. van der Veen. Lookup-table-based secure client-side embedding for spread-spectrum watermarks. *IEEE Transactions on Information Forensics and Security*, 3(3):475–487, September 2008.

[11] H. Cheng and X. Li. Partial encryption of compressed images and videos. *IEEE Transactions on Signal Processing*, 48(8):2439–2451, 2000.

[12] Frederic Dufaux and Touradj Ebrahimi. H.264/AVC video scrambling for privacy protection. In *Proceedings of the IEEE International Conference on Image Processing, ICIP '08*, San Diego, CA, USA, October 2008. IEEE.

[13] Frederic Dufaux and Touradj Ebrahimi. Scrambling for privacy protection in video surveillance systems. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(8):1168–1174, 2008.

[14] Frederic Dufaux and Touradj Ebrahimi. A framework for the validation of privacy protection solutions in video surveillance. In *Proceedings of the IEEE International Conference on Multimedia & Expo, ICME '10*, Singapore, July 2010. IEEE.

[15] Dominik Engel, Thomas Stütz, and Andreas Uhl. A survey on JPEG2000 encryption. *Multimedia Systems*, 15(4):243–270, 2009.

[16] Yibo Fan, Jidong Wang, Takeshi Ikenaga, Yukiyasu Tsunoo, and Satoshi Goto. A new video encryption scheme for H.264/AVC. In *Advances in Multimedia Information Processing, PCM'07*, pages 246–255. Springer-Verlag, 2007.

[17] M. Grangetto, E. Magli, and G. Olmo. Multimedia selective encryption by means of randomized arithmetic coding. *IEEE Transactions on Multimedia*, 8(5):905–917, 2006.

[18] M. Grangetto, E. Magli, and G. Olmo. Conditional access to H.264/AVC video by means of redundant slices. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'07)*, volume 6, pages 485–488, September 2007.

[19] Hermann Hellwagner, Robert Kuschnig, Thomas Stütz, and Andreas Uhl. Efficient in-network adaptation of encrypted H.264/SVC content. *Elsevier Journal on Signal Processing: Image Communication*, 24(9):740 – 758, July 2009.

[20] Heinz Hofbauer and Andreas Uhl. Visual quality indices and low quality images. In *IEEE 2nd European Workshop on Visual Information Processing*, pages 171–176, Paris, France, July 2010.

[21] Internet Streaming Media Alliance. ISMA Encryption and Authentication Specification 2.0, Nov 2007.

[22] Razib Iqbal, Shervin Shirmohammadi, and Abdulmotaleb El-Saddik. Secured MPEG-21 digital item adaptation for H.264 video. In *Proceedings of International Conference on Multimedia & Expo, ICME '06*, pages 2181–2184, Toronto, Canada, July 2006. IEEE.

[23] Razib Iqbal, Shervin Shirmohammadi, and Abdulmotaleb El Saddik. A framework for MPEG-21 DIA based adaptation and perceptual encryption of H.264 video. In Roger Zimmermann and Carsten Griwodz, editors, *Proceedings of SPIE, Multimedia Computing and Networking 2007*, volume 6504. SPIE, 2007.

[24] Razib Iqbal, Shervin Shirmohammadi, Abdulmotaleb El Saddik, and Jiying Zhao. Compressed-domain video processing for adaptation, encryption, and authentication. *IEEE Multimedia*, 15(2):38–50, April 2008.

[25] ISO/IEC 15444-12. Information technology – JPEG2000 image coding system, Part 12: ISO base media file format, April 2005.

[26] ISO/IEC 15444-8. Information technology – JPEG2000 image coding system, Part 8: Secure JPEG2000, April 2007.

[27] ITU-T H.264. Advanced video coding for generic audivisual services, November 2007.

[28] Y. Kim, S. Yin, T. Bae, and Y. Ro. A selective video encryption for the region of interest in scalable video coding. In *Proceedings of the TENCON 2007 - IEEE Region 10 Conference*, pages 1–4, Taipei, Taiwan, October 2007.

[29] R. Kuschnig, I. Kofler, M. Ransburg, and H. Hellwagner. Design options and comparison of in-network H.264/SVC adaptation. *Journal of Visual Communication and Image Representation*, pages 529–542, September 2008.

[30] Sang Gu Kwon, Woong Il Choi, and Byeungwoo Jeon. Digital video scrambling using motion vector and slice relocation. In *Proceedings of Second International Conference of Image Analysis and Recognition, ICIAR'05*, volume 3656 of *Lecture Notes in Computer Science*, pages 207–214, Toronto, Canada, September 2005. Springer-Verlag.

[31] Chang-Youl Lee, Hyun-Jun Choi, Young-Ho Seo, and Dong-Wook Kim. A blind watermarking algorithm for H.264/AVC using entropy coder (CABAC). In *Proceedings of the 7th International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS '06*, Incheon, Korea, April 2006.

[32] Ho-Jae Lee and Jeho Nam. Low complexity controllable scrambler/descrambler for H.264/AVC in compressed domain. In Klara Nahrstedt, Matthew Turk, Yong Rui, Wolfgang Klas, and Ketan Mayer-Patel, editors, *Proceedings of ACM Multimedia 2006*, pages 93–96. ACM, 2006.

[33] Chunhua Li, Xinxin Zhou, and Yuzhuo Zhong. NAL level encryption for scalable video coding. In *Advances in Multimedia Information Processing, PCM'08*, pages 496–505. Springer-Verlag, December 2008.

[34] Shujun Li, Guanrong Chen, Albert Cheung, Bharat Bhargava, and Kwok-Tung Lo. On the design of perceptual MPEG-video encryption algorithms. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(2):214–223, 2007.

[35] Xin Li, Bahadir Gunturk, and Lei Zhang. Image demosaicing: A systematic survey. In *Proceedings of SPIE, Visual Communications and Image Processing, VCIP '08*, volume 6822, pages 68221J–68221J–15, San Jose, CA, USA, January 2008. SPIE.

[36] Yuan Li, Liwei Liang, Zhaopin Su, and Jianguo Jiang. A new video encryption algorithm for H.264. In *Proceedings of the Fifth International Conference on Information, Communications and Signal Processing, ICICS'05*, pages 1121– 1124. IEEE, December 2005.

[37] Shiguo Lian, Zhongxuan Liu, Zhen Ren, and Haila Wang. Secure advanced video coding based on selective encryption algorithms. *IEEE Transactions on Consumer Electronics*, 52(2):621–629, 2006.

[38] Shiguo Lian, Zhongxuan Liu, Zhen Ren, and Haila Wang. Commutative encryption and watermarking in video compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(6):774–778, 2007.

[39] Shiguo Lian, Zhongxuan Liu, Zhen Ren, and Zhiquan Wang. Selective video encryption based on advanced video coding. In *Proceedings of the Pacific-Rim Conference on Multimedia, Advances in Multimedia Information Processing, PCM '05*, volume 3768 of *Lecture Notes in Computer Science*, pages 281–290. Springer, 2005.

[40] Shiguo Lian, Jinsheng Sun, Guangjie Liu, and Zhiquan Wang. Efficient video encryption scheme based on advanced video coding. *Multimedia Tools and Applications*, 38(1):75–89, March 2008.

[41] Fuwen Liu and Hartmut Koenig. A survey of video encryption algorithms. *Computers & Security*, 29(1):3–15, 2010.

[42] Yang Liu, Chun Yuan, and Yuzhuo Zhong. A new digital rights management system in mobile applications using H.264 encryption. In *Proceedings of the 9th International Conference on Advanced Communication Technology*, volume 1, pages 583–586, February 2007.

[43] E. Magli, M. Grangetto, and G. Olmo. Conditional access techniques for H.264/AVC and H.264/SVC compressed video. *IEEE Transactions on Circuits and Systems for Video Technology*, 2008. to appear.

[44] Enrico Magli, Marco Grangetto, and Gabriella Olmo. Conditional access to H.264/AVC video with drift control. In *Proceedings of the IEEE International Conference on Multimedia and Expo, ICME'06*. IEEE, July 2006.

[45] Yinian Mao and Min Wu. A joint signal processing and cryptographic approach to multimedia encryption. *IEEE Transactions on Image Processing*, 15(7):2061–2075, July 2006.

[46] Ayoub Massoudi, Frédéric Lefèbvre, Christophe De Vleeschouwer, Benoit Macq, and Jean-Jacques Quisquater. Overview on selective encryption of image and video, challenges and perspectives. *EURASIP Journal on Information Security*, 2008(Article ID 179290):doi:10.1155/2008/179290, 18 pages, 2008.

[47] Peter Meerwald and Andreas Uhl. Robust watermarking of H.264-encoded video: Extension to SVC. In *Proceedings of the Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP '10*, pages 82–85, Darmstadt, Germany, October 2010.

[48] Cai Mian, Jia Jia, and Yan Lei. An H.264 video encryption algorithm based on entropy coding. In *Proceedings of the Third International Conference on International Information Hiding and Multimedia Signal Processing, IIH-MSP'07*, pages 41–44, Washington, DC, USA, 2007. IEEE Computer Society.

[49] National Institute of Standards and Technology. FIPS-197 - advanced encryption standard (AES), November 2001.

[50] M. Noorkami and R. M. Mersereau. Digital video watermarking in P-frames. In *Proceedings of SPIE, Conference on Security, Steganography and Watermarking of Multimedia Contents IX*, volume 6505, San Jose, CA, USA, January 2007. SPIE.

[51] Lintian Qiao and Klara Nahrstedt. Comparison of MPEG encryption algorithms. *International Journal on Computers and Graphics (Special Issue on Data Security in Image Communication and Networks)*, 22(3):437–444, 1998.

[52] I. E. G. Richardson. *H.264 and MPEG-4 video compression: video coding for next generation multimedia*. Wiley & Sons, 2003.

[53] A. Said. Measuring the strength of partial encryption schemes. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'05)*, volume 2, September 2005.

[54] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1103–1120, September 2007.

[55] Andrew Senior, editor. *Protecting Privacy in Video Surveillance*. Springer, 2009.

[56] Z. Shahid, M. Chaumont, and W. Puech. Fast protection of H.264/AVC by selective encryption of CABAC. In *Proceedings of the IEEE International Conference on Multimedia & Expo, ICME '09*, Cancun, Mexico, June 2009. IEEE.

[57] Susanna Spinsante, Franco Chiaraluce, and Ennio Gambi. Masking video information by partial encryption of H.264/AVC coding parameters. In *Proceedings of the 13th European Signal Prociessing Conference, EUSIPCO'05*. EURASIP, September 2005.

[58] Thomas Stütz, Vinod Pankajakshan, Florent Autrusseau, Andreas Uhl, and Heinz Hofbauer. Subjective and objective quality assessment of transparently encrypted JPEG2000 images. In *Proceedings of the ACM Multimedia and Security Workshop (MMSEC '10)*, Rome, Italy, September 2010. ACM. accepted.

[59] Thomas Stütz and Andreas Uhl. Format-compliant encryption of H.264/AVC and SVC. In *Proceedings of the Eighth IEEE International Symposium on Multimedia (ISM'08)*, Berkeley, CA, USA, December 2008. IEEE Computer Society.

[60] Thomas Stütz and Andreas Uhl. Efficient format-compliant encryption of regular languages: Block-based cycle-walking. In B. De Decker and I. Schaumller-Bichl, editors, *Proceedings of the 11th Joint IFIP TC6 and TC11 Conference on Communications and Multimedia Security, CMS '10*, volume 6109 of *IFIP Advances in Information and Communication Technology*, pages 81 – 92, Linz, Austria, May 2010. Springer.

[61] Thomas Stütz and Andreas Uhl. (In)secure multimedia transmission over RTP. In *Proceedings of the 18th European Signal Processing Conference, EUSIPCO '10*, Aarlborg, Danmark, August 2010. EURASIP.

[62] Po-Chyi Su, Chih-Wei Hsu, and Ching-Yu Wu. A practical design of content protection for H.264/AVC compressed videos by selective encryption and fingerprinting. *Multimedia Tools and Applications*, January 2010. online publication.

[63] Jing Sun, Zhengquan Xu, Jin Liu, and Ye Yao. An objective visual security assessment for cipher-images based on local entropy. *Multimedia Tools and Applications*, March 2010. online publication.

[64] Jean-Baptiste Thomas, Gael Chareyron, and Alain Tremeau. Image watermarking based on a color quantization process. In *Proceedings of the SPIE*, volume 6506, San Jose, CA, USA, January 2007. SPIE.

[65] Nithin Thomas, David Bull, and David Redmill. A novel H.264 SVC encryption scheme for secure bit-rate transcoding. In *Proceedings of*

the Picture Coding Symposium, PCS'09, Chicago, IL, USA, May 2009. IEEE.

[66] Nithin Thomas, Damien Lefol, David Bull, and David Redmil. A novel secure H.264 transcoder using selective encryption. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'07)*. IEEE, September 2007.

[67] L. Tong, F. Dai, Y. Zhang, and J. Li. Prediction restricted H.264/AVC video scrambling for privacy protection. *Electronic Letters*, 46(1):47–49, January 2010.

[68] A. Uhl and A. Pommer. *Image and Video Encryption. From Digital Rights Management to Secured Personal Communication*, volume 15 of *Advances in Information Security*. Springer-Verlag, 2005.

[69] A. Vetro, C. Christopoulos, and T. Ebrahimi. From the guest editors - Universal multimedia access. *IEEE Signal Processing Magazine*, 20(2):16 – 16, 2003.

[70] S.J. Wee and J.G. Apostolopoulos. Secure scalable streaming and secure transcoding with JPEG2000. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'03)*, volume I, pages 547–551, Barcelona, Spain, September 2003.

[71] Jiangtao Wen, Mike Severa, Wenjun Zeng, Max Luttrell, and Weiyin Jin. A format-compliant configurable encryption framework for access control of video. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(6):545–557, June 2002.

[72] S. Wenger, M.M. Hannuksela, T. Stockhammer, M. Westerlund, and D. Singer. RTP Payload Format for H.264 Video. RFC 3984, February 2005.

[73] Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576, July 2003.

[74] Y. G. Won, T. M. Bae, and Y. M. Ro. 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*, pages 407–421, Korea, November 2006. Springer.

[75] Chung-Ping Wu and C.-C.J. Kuo. Design of integrated multimedia compression and encryption systems. *Transactions on Multimedia*, 7(5):828–839, October 2005.

[76] Siu-Kei Au Yeung, Shuyuan Zhu, and Bing Zeng. Partial video encryption based on alternating transforms. *IEEE Signal Processing Letters*, 16(10):893–896, October 2009.

[77] Wenjun Zeng, Junqiang Lan, and Xinhua Zhuang. Security for multimedia adaptation: Architectures and solutions. *IEEE MultiMedia*, 13(2):68–76, 2006.

[78] Wenjun Zeng and Shawmin Lei. Efficient frequency domain selective scrambling of digital video. *IEEE Transactions on Multimedia*, 5(1):118–129, March 2003.

[79] D. Zou and J. Bloom. H.264 stream replacement watermarking with CABAC encoding. In *Proceedings of the IEEE International Conference on Multimedia and Expo, ICME '10*, Singapore, July 2010.

[80] Dekun Zou and Jeffrey A. Bloom. H.264/AVC stream replacement technique for video watermarking. In *Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '08*, pages 1749–1752, Las Vegas, NV, USA, March 2008. IEEE.

[81] Yuanzhi Zou, Tiejun Huang, Wen Gao, and Longshe Huo. H.264 video encryption scheme adaptive to DRM. *IEEE Transactions on Consumer Electronics*, 52(4):1289–1297, November 2006.