Multimed Tools Appl (2018) 77:1537–1553
DOI 10.1007/s11042-017-4389-3

An efficient mechanism of cryptographic synchronization within selectively encrypted H.265/HEVC video stream

Boriša Jovanović[1] · Slavko Gajin[1]

Received: 14 July 2016 / Revised: 7 January 2017 / Accepted: 10 January 2017 /
Published online: 17 January 2017

Abstract The challenge to which the encryption of multimedia data needs to respond is ensuring the security of data intensive video stream in an efficient way. Unlike full data encryption, selective encryption manages to achieve this by encrypting only a part of the data stream, while providing a satisfactory level of video security. This optimizes the processing time and the size of encrypted data. Regardless of the encryption technique, there is a lack of cryptographic synchronization when providing random access to the selected part of the encrypted multimedia stream. In this paper we propose a novel and efficient method of cryptographic synchronization as an extension to the H.265/HEVC crypto encoder in order to support random access in selectively encrypted video stream.

Keywords HEVC · Selective encryption · Random access · Cryptographic synchronization

1 Introduction

Following advancement and rapid development of digital multimedia, larger bandwidths available within the communication network and increased processing power led to the everyday utilization of digital multimedia on different devices and in different areas of life. A large amount of both personal and business multimedia data has consequently become publicly available, and in turn can be more easily copied or modified.

High Efficiency Video Coding (HEVC) is the newest video coding standard of ITU-T and ISO/IEC, proposed in 2013 [3]. This coding standard was developed in response to the growing need for increased video resolution support, higher compression of moving picture

✉ Boriša Jovanović
borisa.jovanovic@vs.rs

Slavko Gajin
slavko.gajin@rcub.bg.ac.rs

[1] School of Electrical Engineering, University of Belgrade, Bulevar kralja Aleksandra 73,
Belgrade 11000, Serbia

and greater use of parallel processing architectures. The design of the HEVC video coding standard provides approximately 30% - 50% bit-rate reduction for the equivalent perceptual quality relative to the performance of the previous standard H.264/AVC High Profile [10]. This feature makes HEVC suitable for various applications such as Internet streaming, communication, real-time conversation comprising video chat, video conferencing and telepresence systems. Furthermore, HEVC can be efficiently used for digital storage media, and broadcasting of high definition (HD) television signal via satellite, cable and terrestrial transmission systems. This makes HEVC an attractive solution for a wide range of video applications, including various Internet services, commercial sectors, as well as military purposes.

This standard represents the most efficient video compression system available nowadays. However, the standard does not provide security mechanisms that would ensure confidentiality and authenticity. There are several publicly available selective encryption algorithms for the previous H.264/AVC standard and for the new H.265/HEVC standard. Selective encryption algorithms are used to protect video stream – a small part of the video stream is encrypted, with minimal resource overhead and still a sufficient security level for most applications.

However, encryption of a small part of the video stream at the transmiting side prevents or hinders random access within selectively encrypted HEVC video stream on the receiving side. Random access within selectively encrypted HEVC video stream, at the receiving side, means that selective encryption algorithm and HEVC decoder can start the deciphering and decoding process at any point of the video stream. This means that they are possible to jump to a particular position within the file, a particular position within the video stream, perform splicing operation or channel switching at any time. At the receiving side, selective encryption algorithm needs to know which parts of HEVC syntax elements necessary to decrypt and that the initial state from which starts when decrypt. In other words, the receiving side (decoder) must be cryptographically synchronized with a transmitting side (encoder). Parts that need to be decrypted are defined by selectively encryption algorithm and the initial state are defined by the selected symmetric cryptographic algorithm.

The essential characteristics of a selective encryption algorithm are: video stream syntax elements that are encrypted and applied symmetric cryptographic algorithm. These characteristics are different among different selectively encryption algorithms. Such diversity requires the existence of the cryptographic synchronization mechanism that is independent from the applied selectively encryption algorithm.

The main contribution of this paper is defining an original and efficient cryptographic synchronization mechanism within the selectively encrypted HEVC video stream that is independent from the applied selective encryption algorithm. This mechanism is achieved by defining the syntax and semantics of the new syntax element in the HEVC bitstream that provides cryptographic synchronization and allows random access to the selectively encrypted HEVC video stream. The efficiency of the offered solutions is reflected in the resulting data overhead. The size of the synchronization parameter is directly proportional to the size of the block of the applied cryptographic algorithm. It also does not depend on the parameters of the encoded video. Defining an additional syntax element is the central part of the paper and it is dedicated to the design of an efficient H.265/HEVC crypto encoder with random access capability.

The outline of the paper is as follows. Section 2 describes the manner of application of selective encryption in the existing solutions. It also briefly describes how the proposed solutions manage the random access problem pertaining to the encrypted HEVC bitstream. Section 3 elaborates on the random access within the HEVC bitstream. Section 4 describes the syntax and semantics of the HEVC bitstream's additional elements. These new syntax elements are designed to enable efficient random access to be applied to the selectively encrypted HEVC bitstream. Section 5 illustrates the experiment results obtained from testing the implemented HEVC crypto encoder with a modified version of the HEVC reference Model (HM) v15.0. The testing was accompanied with test sequences from each class. Finally, section 6 expounds the proposed design and presents the concluding remarks.

2 HEVC selective encryption and related work

At the beginning of 2013, the joint collaborative team comprising the ITU-T and ISO/IEC expert groups completed the HEVC standardization process. The new standard is the successor of the widely used H.264/AVC standard. The HEVC standard outperformed its predecessor with a 50% bitrate reduction with similar subjective quality [10]. As was the case for all past ITU-T and ISO/IEC video coding standards, in HEVC only the bitstream structure and syntax are standardized, as well as constraints on the bitstream and its mapping for the generation of decoded pictures [14].

The increased use of video content, encryption and selective encryption have attracted the attention of the research community in terms of the manner of protecting the video content confidentiality. The video content is large, and both stored and transmitted via a variety of media in a compressed form. Consequently, many researchers have proposed various selective encryption algorithms that are designed to encrypt the video in different compression steps. These algorithms perform encryption either in the pixel, transform or quantized transform domain, while some of them are designed for the bitstream domain.

Owing to its popularity, a number of selective encryption algorithms have been designed for the H.264/AVC standard. Lian et al. [8] present an algorithm for commutative encryption and watermarking of H.264/AVC. The algorithm presupposes combining the selective encryption of some macroblock header fields with watermarking of Discrete Cosine Transform (DCT) coefficient magnitude. It presents a watermarking solution in an encrypted domain without exposing the video content. In [7], the selective encryption of H.264/AVC is performed in fields like intra-picture prediction mode, residue data, inter-picture prediction mode and Motion Vector Difference (MVD). The drawback of the techniques proposed in [7, 8] lies in the fact that they are not format-compliant. Wang et al. [16] presented the partial encryption scheme on the code words of 4 × 4 and 16 × 16 intra-picture prediction mode, EGk code for MVD and level suffix by using an RC4 stream cipher. Park and Shin [12] proposed a selective encryption of H.264/SVC whereby the intra-picture prediction mode, MVD and texture sign bits are encrypted. However, the inter-picture prediction mode encryption, proposed in this paper, affects the compression efficiency by negatively changing the video statistics. In [5], Jiang et al. propose encrypting all intra-picture prediction modes by chaotic pseudo-random sequence. They are then scrambled by the means of circulating sequences that are controlled by keys. This consequently provides key distribution and synchronization scheme. The proposed algorithm ensures a good level of security, but with a slight change in bitrate. Yeung et al. [18] proposed perceptual video encryption at the transform stage by selecting one out of multiple unitary transforms. The unitary transforms were significantly different from the discrete cosine transform (DCT) or discrete sine transform (DST), and the resulting coding efficiency is very close to DCT.

In [19], Yeung et al. extended their selective encryption algorithm, based on the unitary transform, to the transforms of size 8 × 8 for high profiles of H.264. The main drawback of the transform coding based algorithm is that it requires modification in the codec transform module. This is however highly unlikely in case of hardware codec chips and even DSP codecs. Furthermore, keeping all transforms in the instruction cache, particularly in case of embedded devices, is a challenge as well. In their paper [17], Wu and Kuo studied selective encryption that is based on the Huffman table. This encryption technique uses different Huffman tables for different input symbols. The tables, as well as their sequence, are kept secret. As explained in [4], this technique is vulnerable to known plaintext attacks. In [2], Dubois et al. proposed format-compliant reduced selective encryption for H.264/AVC. Under their proposal, the percentage of encrypted bits in the H.264/AVC bitstream was reduced, while the minimum level of visual quality was preserved. The video content was pre-analysed in order to determine whether the quality had already deteriorated due to spatial and temporal prediction, or whether it should be selectively encrypted. Li et al. [6] devised a selective encryption technique for H.264/SVC on both entropy coders i.e. CABAC and Context Adaptive Variable-Length Coding (CAVLC). They proposed adjusting the CABAC initialization tables, thereby rendering the bitstream non-format compliant. Due to the context model change, this also resulted in a bitrate overhead. The algorithm encrypts inter-picture prediction mode with signs of textures for base layers by using the stream cipher Leak Extraction algorithm. None of these selective encryption algorithms addresses the problem of random access pertaining to the encrypted H.264/AVC bitstream.

Once the final version of the H.265/HEVC video compression standard was published it was followed by research and development of selective encryption algorithms for the new standard. Shahid and Puech [13] invented a format-compliant selective encryption algorithm for H.265/HEVC entropy coder – Context-Adaptive Binary Arithmetic Coding (CABAC). In this case, format-compliant selective encryption is performed on a subset of CABAC binstrings which fullfil the real-time constraints. Binstrings are non-binary syntax elements converted to binary form in the binarization step of CABAC engine in HEVC. As this selective encryption is performed independently of the entropy slices, it does not affect the HEVC parallelism. Since there are no changes in bitrate, the advantage of the proposed algorithm lies in its suitability for streaming via a heterogeneous network, e.g. application for military purposes. Experimental results from [13] show that this algorithm, together with the AES encryption algorithm in the CFB mode, provides the desired level of protection. Van Wallendael et al. [15] investigate multiple techniques for partial HEVC video stream encryption that are format-compliant. From their point of view, maintaining the post-encryption format compliance requires the encrypted syntax elements not to change the parsing behaviour of the decoder. Under this algorithm, the set of syntax elements not influencing the decoding process is identified and encrypted by using the AES symmetric encryption algorithm. Experimental results from Van Wallendael et al. [15], regarding compression efficiency and scrambling performance, may serve as good guidelines when making decisions which element to encrypt and with what compression efficiency loss. Kamel and Mokhtar [11] present a novel selective encryption technique for HEVC videos, based on enciphering the bins of selected Golomb-Rice code's suffixes with the AES algorithm in a CBC operating mode. This scheme preserves format compliance as well as the size of the encrypted HEVC bitstream, and provides for a high visual degradation with an optimized encryption space defined by the selected Golomb-Rice suffixes. Experimental results from this paper illustrate reliability and robustness of the proposed technique.

Selective encryption algorithms for the HEVC video stream proposed in [11, 13, 15] did not consider the problem of cryptographic synchronization of encryption and decryption, necessary, however, to support the random access mode. In [3], random access is defined as the act of starting the decoding process for a bitstream at a point other than the beginning of the stream. If the bitstream is encrypted by selective encryption of a syntax element in a format-compliant manner, random access may be slightly different. In respect of the random access point, the decoding process and decryptor in the selective encryption algorithm must have the information necessary to stay in line with the encryptor found on the side from which the encrypted video stream originates. In other words, the decryption process must be able to cryptographically synchronize with the encryption process. If the decryption process is not properly cryptographically synchronized, the decrypted data will be incorrect. Consequently, the result of the video stream decrypting and decoding will not be valid.

The use of the block cryptographic algorithm in the Electronic Code Book (ECB) mode, as shown in [15], does not require any parameters for cryptographic synchronization. However, it has the minimal cryptographic strength when encrypting large amounts of data. Cryptographic algorithms are used in other modes (Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB) or Counter (CTR)) in order to achieve the desired cryptographic strength. On the other hand, each mode displays slightly different behaviour when faced with data bit errors in the cipher text or with block synchronization errors between the encryptor and decryptor. The CBC, CFB and OFB modes can self-synchronize after one encrypted data block only if a whole block-size of cipher text is lost. In these modes, even if a single bit is lost (or has its value changed), decryption will be permanently thrown off. In the CTR mode, synchronization must be done “manually”, using the new values of the initial vector. This is done, because the initial vector value (counter value) for the CTR mode is not same as the value of the previous cipher text block. In case of the CBC, CFB and OFB modes, the initial vector for the decryption process of the current cipher text block is the previous cipher text block and being located somewhere nearly in previous video stream. As it is contingent on the selective encryption algorithm applied, the value of the previous cipher text block can be situated continuously or scattered and intercalated across to the part of the video stream data that was not ciphered. In case when the random access mode is found on the decoder side, such an arrangement of the cryptographic synchronization data can place an additional time burden onto the decryptor process (on the decoder side). Also, in this case the decryptor process must parse data from the previous picture which is not a random access picture (not intra coded picture). Moreover, in case of an error (or loss) of even a single bit from the previous cipher text block, the selective encryption algorithm at the decoder side would not be synchronized and therefore random access would not be possible. In case of video splicing or channel switching, the previous video stream, including the synchronization data, is not available (due to the channel switching operation that causes previous data from the newest channel video stream not to be available).

The proposed solution, for cryptographic synchronization within selectively encrypted HEVC video stream, does not depend on the selected mode of block cryptographic algorithm used within the selective encryption algorithm. The efficiency of the solutions proposed is productive in terms of the overhead percentage.

3 HEVC bitstream and random access capabilities

In the course of video stream processing and coding, at the highest level of abstraction, it is divided into pictures. The first picture of the video sequence is encoded using only the intra-picture prediction (spatial data prediction using the region-to-region principle, within the same picture, but without dependence on other pictures). In the HEVC, the new Clean Random Access (CRA) picture specifies the use of an independently coded picture at the Random Access Point (RAP) location, i.e. location in the bitstream where decoder can successfully begin to decode pictures, without the need to decode any pictures that previously appeared in the bitstream [14]. The random access point is located in the first picture of the stream and in all other subsequent pictures which have been intra-picture predicted. As regards all other pictures in the sequence, or pictures between two adjacent random access points, the encoding process uses the inter-picture prediction (temporal predictions based on an or more adjacent picture from the sequence), based on the previously encoded pictures.

Similarly to the H.264/AVC, the HEVC bitstream contains a number of access units. An access unit is a set of Network Abstraction Level (NAL) units connected with each other under the specified classification rule. These units are consecutive in decoding order and contain exactly one coded picture [3]. NAL units allow mapping of the Video Coding Layer (VCL) data i.e. the content of a picture, onto various transport layers, including RTP/IP, ISO MP4 and H.222/MPEG-2 Systems. They also provide a packet loss resilience framework [14]. Depending on whether they contain coded pictures or other associated data, NAL units are classified into VCL and non-VCL units, respectively. Table 1 shows the integer identifier of NAL unit type, their associated meanings and their class type under the HEVC standard. Each NAL unit contains an NAL unit header and an NAL unit payload. Parameters in the NAL unit header can be accessed by media gateways, also known as Media Aware Network Elements (MANEs), for intelligent, media aware actions on the video stream, such as stream adaptation.

Instead of relying on recovery point Supplemental Enhancement Information (SEI) messages, as was the case in the H.264/AVC, under the HEVC standard, open GOP (Group of Picture) random access points are directly signalled in the NAL unit header. In the H.264/AVC, the bitstream must always start with an Instantaneous Decoder Refresh (IDR) access unit. An

Table 1 Integer identifier of NAL unit type, meaning and NAL type class

ID of NAL unit type	Meaning	Type Class
0, 1	Slice segment of ordinary trailing picture	VCL
2, 3	Slice segment of the TSA picture	VCL
4, 5	Slice segment of the STSA picture	VCL
6, 7	Slice segment of the RADL picture	VCL
8, 9	Slice segment of the RASL picture	VCL
10–15	Reserved for the future use	VCL
16–18	Slice segment of the BLA picture	VCL
19, 20	Slice segment of the IDR picture	VCL
21	Slice segment of the CRA picture	VCL
22–31	Reserved for the future use	VCL
32	Video parameter set (VPS)	non-VCL
33	Sequence parameter set (SPS)	non-VCL
34	Picture parameter set (PPS)	non-VCL
35	Access unit delimiter	non-VCL
36	End of sequence	non-VCL
37	End of bitstream	non-VCL
38	Filler data	non-VCL
39, 40	SEI messages	non-VCL
41–47	Reserved for the future use	non-VCL
48–63	Unspecified (available for system use)	non-VCL

IDR access unit contains an independently coded picture which can be decoded without decoding any previous picture in the bitstream. An IDR access unit in the bitstream indicates that no subsequent picture in the bitstream will require reference to the picture prior to the IDR picture in order to be decoded. The IDR picture is applied within a coding structure also known as a closed GOP. In the HEVC, an open GOP random access point picture is signalled by a specific NAL unit type, and such a picture is named a Clean Random Access (CRA) picture or a Broken-Link Access (BLA) picture. The IDR, CRA and BLA pictures are jointly referred to as an Intra Random Access Point (IRAP) picture. An HEVC conforming bitstream may start with an IRAP picture of any type.

In the HEVC, the CRA and BLA pictures are a new design with special features that enable random access and bitstream splicing. The new CRA picture syntax specifies the use of an independently coded picture at the location of a clean Intra Random Access Point (IRAP). The IRAP is a location in the bitstream where the decoder can begin to effectively decode pictures, without the need to decode any pictures that have previously appeared in the bitstream i.e. it is a random access point. The location of splicing points from different original coded bitstreams can be indicated by BLA pictures. Bitstream splicing operation can be performed by simply changing the NAL unit type of a CRA picture in a bitstream to the value that indicates a BLA picture, and by concatenating the new bitstream at the position of an IRAP picture in the other bitstream.

4 Syntax and semantics of the solution proposed

To facilitate cryptographic synchronization in a selective encryption algorithm, and to enable an efficient random access capable HEVC crypto encoder, we have defined a new HEVC syntax element i.e. a new non-VCL NAL unit. Using value 48 – the first value from the unspecified non-VCL range (from 48 to 63, according to Table 1) – as the NAL unit type identifier, the new NAL unit is called the CSPS NAL unit (CSPS – Crypto Synchronization Parameter Set). This paper specifies the syntax and semantics of the NAL unit whose **nal_unit_type** has this particular value. These particular value for NAL unit types may be applied for a variety of purposes. Therefore encoders generating and decoders interpreting the content of the NAL units with these **nal_unit_type** values [3] must be designed with particular care. Table 1 was then extended with a new row that defines a new NAL Unit type, as shown in Table 2.

General NAL unit syntax and header syntax remained unchanged. Raw Byte Sequence Payload (RBSP) for the new NAL unit payload is defined in Table 3, according to the syntax specification format defined in [3].

These syntax tables specify a superset of syntax for all the allowed bitstreams. Additional constraints of the syntax may be specified, either directly or indirectly, in other clauses. An actual decoder should be able to identify entry points into the bitstream and should also be able to identify and manage non-conforming bitstreams. When **syntax_element** appears, it signals

Table 2 Extension of Table 1

ID of NAL unit type	Meaning	Type class
....	....	....
48	Crypto synchronization parameter set (CSPS)	non-VCL
49–63	Unspecified (available for system use)	non-VCL

that a syntax element is parsed from the bitstream. The bitstream pointer is then advanced to the next position beyond the syntax element in the bitstream parsing process. A statement (in tabular form syntax definition) can either be a syntax element with an associated descriptor or an expression used to specify conditions for the existence, type, and quantity of syntax elements. A descriptor is used to specify the parsing process of each syntax element. There are numerous predefined descriptors which also add to the definition of a new syntax element. The meaning of the descriptors from Table 3 is as follows: ***ue(v)*** – indicate and specify the parsing process for unsigned integer 0-th order Exp-Golombo-coded syntax element with the left bit first; ***b(8)*** – specify the byte having any pattern of 8 bit string; ***u(n)*** – indicate unsigned integer using n bits, while ***u(1)*** is used for one bit, interpreted as a flag. Other available descriptors and their meanings are defined in [3].

Semantics of crypto synchronization parameter set RBSP is defined as follows:

- **csps_parameter_set_id** identifies the CSPS for reference by another syntax element or by a selective encryption algorithm on the decoder side, aimed at identifying the point in the bitstream and parameters from which the last cryptographic resynchronization has been made.
- **csps_crypto_parameter_ctx_id** identifies the context of a selective encryption algorithm on the decoder side for which the synchronization data are assigned. The context of the selective encryption algorithm is a set of encryption algorithm identifiers, the secret key used and other parameters required for such a selectively encrypted bitstream. This identifier is used for separating the sources of encoded and encrypted video streams at the decoder side. Separation of the sources of the selectively encrypted video stream is particularly important in the bitstream splicing operation. This value must be unique.
- **csps_len_of_crypto_synh_data** is the integer number of bytes in cryptographic synchronization data. This value must not equal 0.
- **csps_crypto_synh_data[i]** is the i-th byte of cryptographic synchronization RBSP data. These RBSP data are defined in the following byte sequence: the first byte of the cs_crypto_synh_data contains the most significant (leftmost) eight bits of the cryptographic synchronization data; the next byte of the cs_crypto_synh_data contains the next eight bits of the cryptographic synchronization data, etc. This goes on until the last eight bytes of cs_crypto_synh_data which contain the least significant (rightmost) eight bits of the cryptographic synchronization data. In this case, it is assumed that the size of cryptographic synchronization data is an integer number of bytes.

Table 3 Crypto synchronization parameter set RBSP syntax in tabular form

crypto_synchronization_parameter_set_rbsp() {	Descriptor
csps_parameter_set_id	ue(v)
csps_crypto_parameter_ctx_id	ue(v)
csps_len_of_crypto_synh_data	ue(v)
for(*i* = 1; i < =cspc_len_of_crypto_synh_data; i++)	
csps_crypto_synh_data[i]	b(8)
csps_extension_flag	u(1)
if(csps_extension_flag)	
while(more_rbsp_data( ))	
csps_extension_data_flag	u(1)
rbsp_trailing_bits( )	
}	

- **csps_extension_flag** that equals 0 means that no csps_extension_data_flag syntax element is present in the CSPS RBSP syntax structure. The value of csps_extension_flag must be 0 in the bitstream, conforming to the standard HEVC decoder, while the value of 1 is reserved for the future use. This principle is used in the design presented in the paper so as to be compatible with definitions of other syntax elements. Under the latest version of HEVC standard, decoders may ignore all data that follow the value 1 for csps_extension_flag in the CSPS NAL unit.
- **csps_extension_data_flag** can have any value. Its presence and value do not affect conformity of the decoder with the standard HEVC decoder. The decoder's conformity with the standard HEVC decoder will ignore all **csps_extension_data_flag** syntax elements.

Once the syntax and semantics of the new non-VCL NAL units are defined, the next step is defining their place in the HEVC bitstream. Choosing where to store such a NAL unit requires making trade-offs between the requirements that must comply with the constraints of the NAL units, as defined in [3]. NAL units should also be easily used for the cryptographic synchronization of a selective encryption algorithm when implementing random access to the selectively encrypted HEVC bitstream.

The new non-VCL NAL unit should be placed inside the IRAP access unit since it is intended for the cryptographic resynchronization of the selective encryption algorithm during random access to the HEVC bitstream. An IRAP access unit can be an IDR access unit, BLA access unit or CRA access unit. The access unit can consist of coded picture VCL NAL units and zero or more non-VCL NAL units. The order of NAL units of the IRAP access unit within the HEVC bitstream, at the beginning or after the last VCL NAL unit of a previous coded picture, may be as shown on Fig. 1.

In this manner, the defined non-VCL NAL unit with **nal_unit_type** value of 48, i.e. CSPS NAL unit, which precedes the first VCL NAL unit inside the IRAP access unit and which is not followed by the last VCL NAL unit of the coded picture within the IRAP access unit, should resolve all constraints related to the decoding order of the NAL unit within the HEVC bitstream.

A CSPS RBSP includes parameters than can be referred by a selective encryption algorithm. These parameters are used for coding and selectively encrypting NAL units of an IRAP picture. At the start of the decoding operation and decryption process, each CSPS RBSP is initially considered inactive. During decoding and decryption of VCL NAL units with an

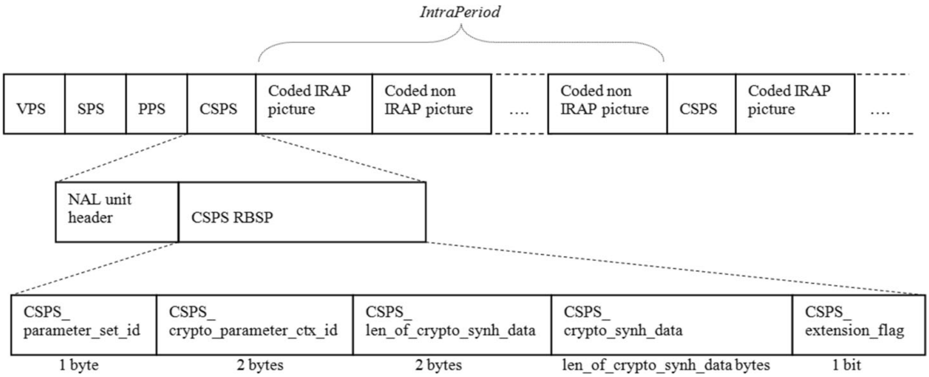

Fig. 1 Overall HEVC bitstream structure, with inserted CSPS NAL unit

IRAP picture, no more than one CSPS RBSP is considered active at any given moment. In regular use, on the decoder side, the **csps_parameter_set_id** value of the next CSPS NAL unit is larger by one compared to the previous-active CSPS NAL unit (function used to generate the next value for the **csps_parameter_set_id** parameter at the encoder side is incremented by one). If for any reason (random access to certain location at the video stream, error in the video stream data, video splicing operation and similar operation), the value of **csps_parameter_set_id** on the current CSPS NAL unit (currently decoded by decoder) does not equal the **csps_parameter_set_id** value of the active-previous CSPS NAL unit incremented by one, the activation process of the CSPS NAL unit must be scheduled. Activating any particular CSPS means resynchronizing the cryptographic algorithm that is used within the actual selective encryption algorithm. If any particular CSPS RBSP is activated, the previously active CSPS RBSP will be deactivated, provided it exists (e.g. activation of the first CSPS NAL unit – there is no previous unit). Video splicing operation (channel switching) into the selectively encrypted video stream can be detected by comparing values of **csps_crypto_parameter_ctx_id** parameters. If the value of the **csps_crypto_parameter_ctx_id** parameter at the current CSPS NAL unit (currently decoded by decoder) does not correspond to the value of the parameter at the active-previous CSPS NAL unit, then source of this video stream is different. In such a case, channel switching, together with switching and resynchronization of the selective encryption algorithm, must be done. It is essential that the value of the **csps_crypto_parameter_ctx_id** parameter is unique, to be able to uniquely distinguish different sources of the selectively encrypted video stream (e.g. different military equipment with video streaming capability).

Adding a new syntax element, with previously defined syntax and semantics, will augment the number of bytes written to the file that contains the selectively encrypted HEVC video, i.e. it will augment the number of bytes in the selectively encrypted HEVC video stream. We use $\boldsymbol{\Delta_B}$ to label the difference between the number of bytes written to the file at the selectively encrypted video stream with an additional non-VCL CSPS NAL unit ($\boldsymbol{N_{SE}}$), and the number of bytes written to the file in the plain video stream ($\boldsymbol{N_P}$):

$$\Delta_B = N_{SE} - N_P \tag{1}$$

Under the proposed solution of the efficient random access capable HEVC crypto encoder, the value of $\boldsymbol{\Delta_B}$ does not depend on the number of bytes in the video stream. This is the result of the fact that a selective encryption algorithm did not insert any additional bytes or bits into the video stream. The value of $\boldsymbol{\Delta_B}$ depends on the total number of frames (images) in the video stream, the size of the CSPS NAL unit data and the value of *IntraPeriod*[1*] parameter. Consequently, the number of overhead bytes can be calculated in advance, depending on the selected IntraPeriod value, and by using the following formula:

$$\Delta_B = \left\lceil \frac{N_f}{T_{IP}} \right\rceil * N_{\mathrm{CSPS}} \tag{2}$$

Where ($\boldsymbol{N_f}$) stands for the total number of frames in the video stream, ($\boldsymbol{N_{CSPS}}$) stands for the CSPS NAL unit size expressed in bytes and ($\boldsymbol{T_{IP}}$) stands for the value of the chosen *IntraPeriod* parameter in frame numbers.

[1*] IntraPeriod parameter defines the number of images (frames) between two adjacent IRAP images.

To analyse the impact of the added syntax elements on the increase in the number of bytes in the video stream, we can calculate the overhead in the selectively encrypted HEVC bitstream with the inserted CSPS parameters. We use $\boldsymbol{\Delta_P}$ to label the overhead in the selectively encrypted HEVC bitstream relative to the plain HEVC video stream, expressed in percentage points. The value of $\boldsymbol{\Delta_P}$ depends on the total number of additionally inserted bytes of SCPS parameters and the total number of bytes in the selectively encrypted HEVC video stream. We can calculate the $\boldsymbol{\Delta_P}$ parameter by using the following formula:

$$\Delta_P = \frac{100{*}\Delta_B}{N_{SE}} \tag{3}$$

Bytes written to the file at the selectively encrypted video stream with an additional non-VCL CSPS NAL unit ($\boldsymbol{N_{SE}}$) can be calculated by multiplying the total number of frames in the video stream ($\boldsymbol{N_f}$) by the mean value of the frame size in bytes ($\boldsymbol{S_f}$) while taking into account both VCL and non-VCL NAL units:

$$N_{SE} = N_f{*}S_f \tag{4}$$

Formulas 2, 3 and 4 indicate that the $\boldsymbol{\Delta_P}$ parameter can be calculated using the following formula:

$$\Delta_P = \frac{100{*}N_{\mathrm{CSPS}}}{S_f{*}T_{IP}}. \tag{5}$$

Let us consider the following: the possibility of cryptographic synchronization for every second of encrypted video stream, the fact that the $\boldsymbol{T_{IP}}$ parameter may range from 30 to 60 frames; and let us take into account that in case of the implementation proposed by this paper the size of synchronization data $\boldsymbol{N_{CSPS}}$ measures exactly 31 bytes. It follows that the value of the $\boldsymbol{\Delta_P}$ parameter is approximately equal to:

$$\Delta_P \approx \frac{100}{S_f}. \tag{6}$$

The above formula suggests that the $\boldsymbol{\Delta_P}$ parameter decreases when the mean value of the frame size increases. The increase in video resolution is followed by the increase in frame size mean value. We can therefore conclude that, in case of a higher-resolution video content, the overhead (in percentage points) for cryptographic synchronization data is smaller.

5 Experiment results

For the purpose of this study, we used the HEVC reference model (HM) v15.0[9] encoder implementation. Firstly, we encoded a set of sequences without selective encryption and without inserting cryptographic synchronization data. These sets of video streams are called plain video streams and are generated by an unmodified version of the HEVC reference model (HM) v15.0. Secondly, maintaining the conditions, we encoded the same set of sequences using the selective encryption algorithm defined in [13]. This was done by inserting cryptographic synchronization data into newly defined non-VCL NAL unit syntax elements at specific, above-mentioned, locations in the bitstream. These video streams are called selectively encrypted random access capable video streams and are generated by our modified version of the HEVC reference model (HM) v15.0. It is consequently possible to measure the bit rate impact that is exerted by enabling selective encryption with integrated synchronization data between the plain and the selectively encrypted random access capable video streams.

All test sequences are generated on an Intel(R) Xeon(R) X5570 CPU (2.93 GHz) with 6GB of RAM memory. Tests were conducted on a set of test sequences, as listed in Table 4. The aim was to measure the bit rate impact of the proposed solutions. The test set contains 11 8-bit sequences, whose resolution ranges from 2560 × 1600 down to 416 × 240 (Table 4). This is the subset of sequences used in the HEVC standardization process [1].

For the purposes of our research, we used the algorithm defined in [13] for selective encryption. The AES cryptographic algorithm in the CFB mode is used in this implementation in the same manner as in the algorithm. An important assumption to be made is that the secret keys for the applied symmetric cryptographic algorithm (in this case AES in the CFB mode) have previously been exchanged by using some well-known mechanisms for symmetric cryptography key exchange. It is necessary and sufficient in this mode to deliver the value of the initialization vector for the cryptographic synchronization purposes on the receiving side of communication (decoder side). The initialization vector is a data block of the same size as the cipher block of the used cryptographic algorithm. The initialization vector must be known to both the sender (encoder) and receiver (decoder), but may be rendered unpredictable to a third party. The initialization vector may be protected against unauthorized changes for the purpose of maximum security, which may be achieved by sending the initialization vector through ECB encryption. ECB encryption of IV data was not done in our work and will be the subject of future review and improvement of the offered solutions.

The security of the selectively encrypted video stream with cryptographic synchronization data, i.e. its resistance to known cryptanalysis attacks, depends entirely on the applied selective encryption algorithm. In [13], it is shown that the proposed algorithm is resistant to the known attacks.

In this particular implementation process, the **csps_crypto_synh_data[i]** represents the i-th byte of the initialization vector. When the size of the initialization vector (16 bytes in the case of the AES algorithm) is added to the size of other CSPS parameters (one byte for the size of synchronization data, twice by two bytes for identifiers, the size of the NAL unit header and bytes “00 00 01” at the start of non-VCL NAL unit), the size of the CSPS NAL units equals exactly 31 bytes.

Table 5 shows the number of bytes written to a file and the corresponding bit rate for both plain and selectively encrypted cryptographically synchronizable video streams. For example,

Table 4 The test set sequences used for evaluation of the proposed solution

Sequence	Resolution	Total frame	Frame rate (fps)	Class
Traffic	2560 × 1600	150	30	A
PeopleOnStreet	2560 × 1600	150	30	A
Kimono1	1920 × 1080	240	24	B1
ParkScene	1920 × 1080	240	24	B1
MobileCalendar	1280 × 720	504	50	E
City	1280 × 720	900	60	E
PartyScene	832 × 480	500	50	C
BQMall	832 × 480	600	60	C
BasketballPass	416 × 240	500	50	D
BlowingBubbles	416 × 240	500	50	D
RaceHorses	416 × 240	300	30	D

Table 5 Analysis of bit rate impact of the proposed solution on benchmark video sequences and *IntraPeriod* value of 32

Sequence	Plain		Selectively encrypted		Δ_B (b)	$\mathbf{\Delta_P}$ **(%)**
	Bytes written to file	Bitrate (kbps)	Bytes written to file	Bitrate (kbps)		
Traffic	1595493	2552.79	1595648	2553.03	155	0.0097
PeopleOnStreet	5154255	8246.88	5154410	8247.05	155	0.0030
Kimono1	1317307	1053.86	1317555	1054.04	248	0.0188
ParkScene	1813986	1451.19	1814234	1451.38	248	0.0136
MobileCalendar	1512970	1203.17	1513466	1203.55	496	0.0327
City	2518665	1343.28	2519564	1343.76	899	0.0356
PartyScene	1988867	1591.04	1989363	1591.49	496	0.0249
BQMall	1167012	933.60	1167601	934.08	589	0.0504
BasketballPass	475957	380.76	476453	381.16	496	0.1041
BlowingBubbles	481479	385.13	481975	385.58	496	0.1029
RaceHorses	357537	286.00	357847	286.27	310	0.0866

the value of the ***Traffic*** sequence indicates that the number of bytes written to the file is higher in the selectively encrypted video stream with an added non-VCL CSPS NAL unit. The exact difference i.e. $\mathbf{\Delta_B}$ equals 155 bytes. Since this sequence has 150 frames (see Table 4) and since the *IntraPeriod* equals 32, we can conclude that the resulting video stream has 5 I frames (5 IRAP picture), and therefore 5 non-VCL CSPS NAL units, each of 31 bytes. This is confirmed by its compliance with the basic requirements for selective encryption algorithms, which entail that no extra bit can be entered into the video stream. Added bytes belong exclusively to newly inserted non-VCL CSPS NAL units and are intended for cryptographic synchronization. Such bytes, statistically speaking, make up only 0.0097% of the total amount of data bytes in the file of the selectively encrypted video stream for ***Traffic*** sequence. Furthermore, in the case of the ***PeopleOnStreet*** sequence (also in class A) which has 150 frames, the difference in the number of bytes written to the file under the same conditions also measures 155 bytes. In the case of the ***PeopleOnStreet*** sequence, the added bytes account for only 0.0030% of the total amount of data in the selectively encrypted video stream. If we analyse sequences with lower resolution (e.g. ***BlowingBubbles*** of 500 frames), the difference in the number of bytes written to the file equals 496, which stems from the fact that within the 500 frames there are exactly 16 I frames (IRAP pictures) in the video stream encoded with *IntraPeriod* with value of 32. Figure 2 shows the difference in the number of bytes written to a file for all tested sequences.

Selectively encrypted random access capable video stream has as many CSPS NAL units as there are IRAP access units. The number of IRAP access units was directly affected by the value of the IntraPeriod parameter. Table 6 shows the bit rate impact of different values of the IntraPeriod parameter. The table shows the number of bytes written to the file and the corresponding bitrate for both plain and selectively encrypted random access capable video streams of ***Traffic*** sequence for different values of the *IntraPeriod* parameter, which takes values 8, 16, 24 and 32.

As seen in Fig. 3, the impact of added non-VCL CSPS NAL units on the increase in the total number of bytes written to the file is minimal, as is on the increase in the bitrate.

Judging by Table 6, Fig. 3 and formulas 1, 2 and 3, it can be concluded that the number of bytes (and therefore the number of SCPS NAL units) added to the selectively encrypted video stream is inversely proportional to the value of the *IntraPeriod* parameter. In the same video sequence, the total number of added bytes intended for cryptographic synchronization is

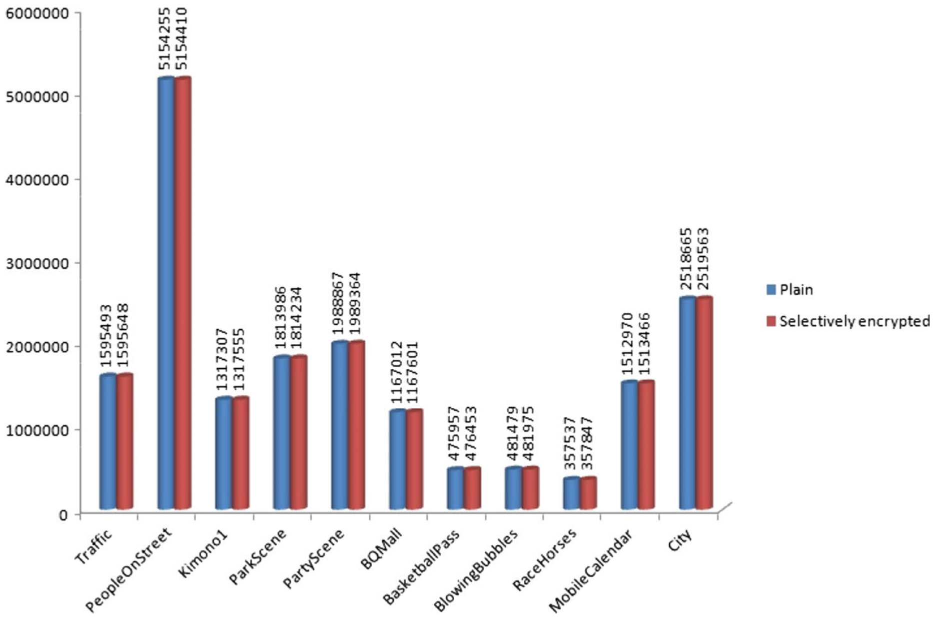

Fig. 2 Graphically illustrated difference in the number of bytes written to the file for benchmark video sequences at QP value of 32 and IntraPeriod with value of 32. QP – Quantization Parameter: a variable used by the decoding process for scaling of transform coefficient levels

reduced with the increase of the *IntraPeriod* parameter. This is a direct consequence of the change in the value of *IntraPeriod* parameter.

Selectively encrypted random access capable video streams, with inserted CSPS NAL units, are fully HEVC compatible. Namely, when these video streams are played in a commercial HEVC player, the decoder can decode them. However, the commercial decoder cannot decrypt them, which leads to cases shown in Fig. 4 (d, e and f). The decoder that does not recognize the selective encryption algorithm simply ignores non-VCL CSPS NAL units. Figure 4 shows selected frames of ***BlowingBubbles*** sequence with and without selective encryption and inserted data for cryptographic synchronization played in the commercial HEVC player.

6 Concluding remarks

This paper presents a novel design of an efficient random access capable HEVC crypto encoder. It sets out to provide a detailed analysis including shortcomings in the

Table 6 The analysis of the bit rate impact of the proposed solution for Traffic benchmark video sequence with different values of *IntraPeriod* parameter

IntraPeriod value	Plain		Selectively encrypted		Δ_B (b)	Δ_P (%)
	Bytes written to file	Bitrate (kbps)	Bytes written to file	Bitrate (kbps)		
8	3,257,174	5211.47	3,257,763	5212.42	589	0.0181
16	2,186,523	3498.43	2,186,833	3498.93	310	0.0142
24	1,833,577	2933.72	1,833,794	2934.07	217	0.0118
32	1,595,493	2552.78	1,595,648	2553.03	155	0.0097

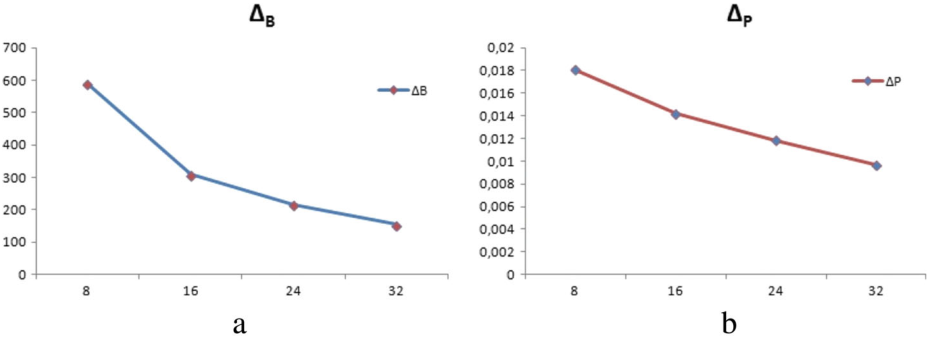

Fig. 3 Graphically illustrated impact of different values of the *IntraPeriod* parameter to: (**a**) number of added bytes (**b**) percentage expressed overhead

field of cryptographic synchronization of the most popular selective encryption algorithms for modern video encoding standards. The analysis is followed by the proposed design, along with the definition of the syntax and semantics of the new non-VCL NAL unit which contains data required for cryptographic synchronization. Applying these new defined syntax elements allows implementing an efficient HEVC crypto encoder with the random access capability. Moreover, the effectiveness of the proposed solution is reflected both in the minimum amount of additional data in the HEVC video stream and in its independence from the applied selective encryption algorithm. It depends only on the applied algorithms (selective encryption algorithm and cryptographic algorithm) in respect of the selection of data required for the cryptographic synchronization, whose aim is to enable random access. Such efficiency of the proposed solutions may be useful when integrating the selective encryption mechanisms within the architecture parallelism of the HEVC standard. Our future work will focus on improving random access mechanisms within the effective integration of the parallelism of the HEVC standards and either existing or newly defined selective encryption algorithms.

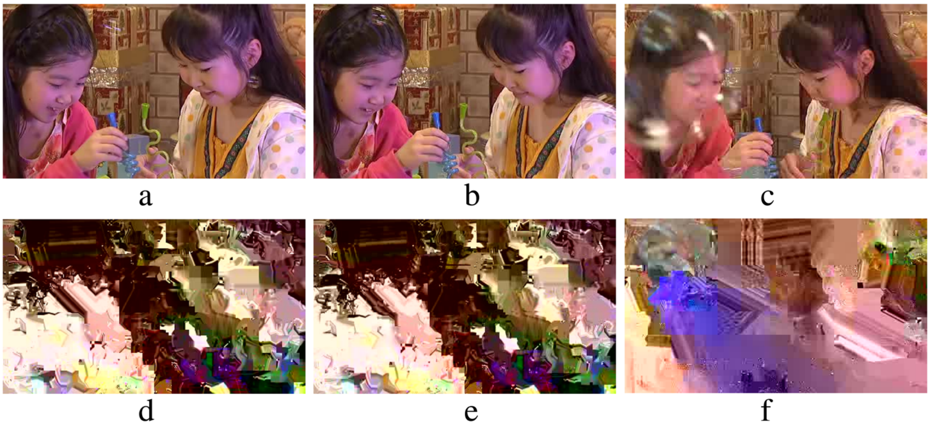

Fig. 4 Frames #0 #8 #32 (I, B and B respectively) of ***BlowingBubbles*** sequence with and without selective encryption for QP value 32 and *IntraPeriod* value 32: (**a**) Original #0, (**b**) Original #8, (**c**) Original #32, (**d**) SE with CSPS #0, (**e**) SE with CSPS #8 and (**f**) SE with CSPS #32

References

1. Baroncini V, Ohm JR, Sullivan G (2010) Report of subjective test results of responses to the joint call for proposals (CfP) on Video Coding Technology for High Efficiency Video Coding (HEVC). ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Doc. JCTVC-A204, Geneva, CH, Tech. Rep
2. Dubois L, Puech W, Blanc-Talon J (2011) Fast protection of H.264/AVC by reduced selective encryption of CAVLC. In: Proc. European Signal Processing Conference, Barcelona, Spain, pp 2185–2189. doi: 10.1109/TCSVT.2011.2129090
3. High Efficiency Video Coding, ITU-TRec.H.265 and ISO/IEC23008–2 (MPEG-H, Part 2), Apr (2013) version 1
4. Jakimoski G, Subbalakshmi K (2008) Cryptanalysis of some multimedia encryption schemes. IEEE Transactions on Multimedia 10(3):330–338. doi:10.1109/TMM.2008.917355
5. Jiang MQJ, Xing S (2009) An intra prediction mode-based video encryption algorithm in H.264. In: proc. International Conference on Multimedia Information Networking and Security 1, pp 478–482. doi: 10.1109/MINES.2009.26
6. Lee HJ, Nam J (2006) Low complexity controllable scrambler/descrambler for H.264/AVC in compressed domain. In: Proc. ACM International Conference on Multimedia, New York, N Y, USA pp 93–96. doi: 10.1145/1180639.1180668
7. Lian S, Liu Z, Ren Z, Wang Z (2005) Selective video encryption based on advanced video coding. In: *Lecture notes in Computer Science*, vol 3768. Springer-verlag, pp 281–290. doi: 10.1007/11582267_25
8. Lian S, Liu Z, Ren Z, Wang H (2007) Commutative encryption and watermarking in video compression. IEEE Transactions on Circuits and Systems for Video Technology 17(6):774–778. doi:10.1109/TCSVT.2007.896635
9. McCann K, Bross B, Han WJ, Kim IK, Sugimoto K, Sullivan GJ (2014) High Efficiency Video Coding (HEVC) Test Model 15 (HM 15) Encoder Description. JCTVC-Q1002, Valencia, Spain
10. Ohm J-R, Sullivan GJ, Schwarz H, Tan TK, Wiegand T (2012) Comparison of the coding efficiency of video coding standards - including high efficiency video coding (HEVC). IEEE Trans Circuits Syst Video Technol 22(12):1668–1683. doi:10.1109/TCSVT.2012.2221192
11. Ouamri M, Faraoun KM (2014) Robust and fast selective encryption for HEVC videos. J Commun Softw Syst 10(4)
12. Park S, Shin S (2009) An efficient encryption and key management scheme for layered access control of H.264/scalable video coding. IEICE Trans Inf Syst 92(5):851–858. doi:10.1587/transinf.E92.D.851
13. Shahid Z, Puech W (2014) Visual protection of HEVC video by selective encryption of CABAC binstrings. IEEE Transactions on Multimedia, vol 16:24–36. doi:10.1109/TMM.2013.2281029
14. Sullivan GJ, Ohm J-R, Han W-J, Wiegand T (2012) Overview of the high efficiency video coding (HEVC) standard. IEEE Circuits and Systems for Video Technology 22(12):1649–1668. doi:10.1109/TCSVT.2012.2221191
15. Van Wallendael G, Boho A, De Cock J, Munteanu A, Van de Walle R (2013) Encryption for high efficiency video coding with video adaptation capabilities. IEEE Transaction on Consumer Electronics 59(3):634–642. doi:10.1109/TCE.2013.6626250
16. Wang J, Fan Y, Ikenaga T, Goto S (2007) A partial scramble scheme for H.264 Video. In: Proc. 7th International Conference on ASIC, Guln, China, pp. 802–805. doi: 10.1109/ICASIC.2007.4415752
17. Wu C, Kuo C (2005) Design of Integrated Multimedia Compression and Encryption Systems. IEEE Transactions on Multimedia 7:828–839. doi:10.1109/TMM.2005.854469
18. Yeung SKA, Zhu S, Zeng B (2011a) Design of new unitary transforms for perceptual video encryption. IEEE Trans Circuits Syst Video Technol 21(9):1341–1345. doi:10.1109/TCSVT.2011.2125630
19. Yeung SKA, Zhu S, Zeng B (2011b) Perceptual video encryption using multiple 8 × 8 transforms in H.264 and MPEG-4. In: Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, pp 2436–2439. doi: 10.1109/ICASSP.2011.5946976

Boriša Jovanović is a graduate at Department of Computer Engineering and Informatics at the Military Technical Academy, Belgrade, Serbia, in 2006. Since 2008 he is working towards a Ph.D. at University of Belgrade – School of Electrical Engineering with financial support of the Ministry of Defense of the Republic of Serbia. Currently working in the Center of Applied Mathematics and Electronics in the Army of Serbia. His research interests include the following: network security, multimedia security, secure multimedia transmission over network, selective encryption.

Slavko Gajin received dipl. Eng., MS and PhD degrees from University of Belgrade, School of Electrical Engineering, Serbia, in 1993, 1999, and 2007, respectively. He is currently working as a director of Belgrade University Computer Centre, where he started working as a network engineer since he received bachelor's degree. He is also a professor at the Department of Computer Engineering and Computer Science at the School of Electrical Engineering, University of Belgrade, and the School of Electrical Engineering University of Banja Luka, where he is teaching topics in the field of computer networks and Internet technologies. His current research interests include the following: Computer networks, Network fault and performance monitoring, Multicomputers, e-Learning, video conferences.