Journal of Ambient Intelligence and Humanized Computing
https://doi.org/10.1007/s12652-020-01895-2

ORIGINAL RESEARCH

Fuzzy-logic threat classification for multi-level selective encryption over real-time video streams

Amna Shifa[1] · Mamoona Naveed Asghar[1,2] · Adeel Ahmed[1] · Martin Fleury[3]

Received: 4 February 2019 / Accepted: 13 March 2020

Abstract
This paper proposes a Fuzzy-logic Threat Classification (FTC) model as the basis of a method to auto-detect three different confidentiality levels for videos streamed from heterogeneous, mobile devices via web edge servers, possibly part of a Content Distribution Network (CDN). The FTC consists of three parallel Fuzzy Inference Systems (FIS) corresponding to device, network, and type of video application, for the real-time, intelligent selection of an appropriate confidentiality level for a specific end-user. After selection of the level, an encryption module implements the corresponding form of encryption. In tests to demonstrate the concept, there were three increasing confidentiality levels, namely (1) low-level with no encryption, (2) Medium level with an in-house cipher [variant of eXclusive OR (XOR)], named P-XOR (XOR with additional rounds of permutation) applied to Selective Encryption (SE) and (3) high level with the Advanced Encryption Standard again for SE of compressed video syntax components. Results were obtained by considering realistic specifications of multiple digital devices, networks, and differing real-time streaming applications. Visual analysis of encrypted test video clips established that the FTC outputs an appropriate privacy level by reason of the implemented FISs. Absolute encryption times across the privacy levels were distinguished by their real-time response level, which is proportionate to the required degree of confidentiality.

Keywords Classification model · CDN · Edge servers · Fuzzy rule-based system · HEVC and H.264/AVC codecs · Confidentiality levels · Selective encryption

1 Introduction

Video chat, video-on-demand, group video conferencing, and video clip access are some of the applications available to stream over un-trusted networks. Apart from publically available video clips, people taking up these applications require mechanisms to exchange video in a confidential manner, as do companies to protect their content. However, it may be wasteful of resources, in terms of bandwidth and computational overhead or latency, in all cases, to apply the same level of encryption. Therefore, the intention of this paper is to provide a means of distinguishing between different classes of real-time video streaming in terms of the level of encryption that should be applied to them. Because the classification should be performed in real-time and because it depends on more than one factor, such as type of network and streaming device, artificial intelligence is appropriate. Given the characteristics of this problem, Fuzzy-logic Threat Classification (FTC) seems to be particularly appropriate.

This paper considers three classes of video traffic: (1) public Real-Time (streaming) Application (RTA) (such as streaming of video clips on YouTube and other similar sites), which may require no encryption because the material is anyway publicly available; (2) protected RTA (such as group conferences), which may allow a relaxation of the confidentiality protection in the interests of reduced latency and allowing participation by all; and (3) private RTAs [such as personal videos chats and Video-on-Demand (VoD)], for

✉ Martin Fleury
MartinFleury@uos.ac.uk; fleury.martin55@gmail.com

Amna Shifa
amna.shifa@gmail.com

Mamoona Naveed Asghar
mamona.asghar@iub.edu.pk; masghar@ait.ie

Adeel Ahmed
adeelmcs@gmail.com

[1] The Islamia University of Bahawalpur, Punjab, Pakistan

[2] Software Research Institute, Athlone Institute of Technology, Athlone, Ireland

[3] School of EAST, University of Suffolk, Ipswich, UK

Published online: 22 April 2020

which private users might want a guarantee of full confidentiality, as might commercial operators to protect their content from copying; these video traffic classes are no definitive but serve to illustrate the concept of this paper, which is to employ computational intelligence, namely fuzzy logic, to decide, automatically and in real-time, which of the three confidentiality levels (i.e., low, medium, and high) a particular streaming session falls into. This decision is made, in the paper, not only according to the application, but also according to the devices receiving the video streams and the network characteristics, particularly at the bottleneck link.

One way that the video may reach end-users is, via a cloud, over a Content Distribution Network (CDN) to edge servers (Fan et al. 2018), where cached content may be held in the case of VoD and video clip streams. Some companies provide privacy related solutions to the CDNs with access control and encryption over streaming videos (Kollective Technology 2018). However, there remains a need to consider whether all types of streaming videos require encryption or only specific ones (Cui et al. 2018). Despite existing solutions, the following questions can still arise in CDN privacy configuration:

Q1 Do all types of streamed video require the same level of confidentiality?
Q2 Would a confidentiality threat classification model be helpful for personal, semi-personal, and impersonal video streaming?
Q3 How can a specific confidentiality level for a streamed video be determined in real-time for a particular end-user, in terms of their mobile device and their wireless access network?

This research work provides a solution for the aforementioned questions by designing and implementing a Fuzzy Threat Classification (FTC) model. In the implementation, to reduce further encryption computational overhead, a lightweight Selective Encryption (SE) scheme (Lookabaugh and Sicker 2004) is applied. SE allows sufficient encryption to be put in place (Furht et al. 2005) to so distort the video stream that it is no longer useful to somebody lacking the decryption key. The selected compression syntax components are encrypted either with the Advanced Encryption Standard (AES) (National Institute of Standards and Technology 2001) or with an in-house variant of the eXclusive OR (XOR) cipher, named P-XOR (XOR with additional rounds of permutation). In experiments, these encryption schemes are applied to either H.264/AVC or HEVC (two current standardized codecs0 encoded videos.

In addition, as previously mentioned, at the lowest privacy level no encryption is applied. SE does not in any case encrypt all of the video stream but only sufficient components of the compressed video bitstream so that an attempt at decoding (if that is possible without crashing the decoder, depending on the form of the SE) results in a highly distorted video (Shahid and Puech 2014). Thus, by only selecting some syntax components of the video stream, SE already results in some computational savings. However, except for particular applications, such as military or security surveillance, the overhead of complete encryption of all video is not required and can actually pose a security risk if intermediate processing is required (because of the risk of key exposure). The proposed FTC model is the basis of a method to determine the appropriate privacy level for each end-user streaming session, which is then provided in real-time through SE. To the authors' best knowledge, this proposed solution is innovatory and can provide a practical solution to the privacy threat problems of streaming videos via CDNs or by another means to achieve real-time delivery. The research contribution is summarized as follows. The paper:

- Develops an FTC to auto-detect the privacy level for streaming video, which could also be applicable to multimedia sources in general such as streamed audio.
- Sets the fuzzy rules to select the privacy level in real-time.
- Uses actual specifications of digital devices, networks, and real-time applications to test the FTC-based method.
- Implements SE using a standard encryption algorithm AES or lightweight cipher P-XOR to provide privacy for a particular end-user.
- Presents an evaluation of the FTC model with the visual results of applying each privacy level to the benchmark videos. (Diffie-Hellman key exchange algorithm allows rapid session-wise key management, avoiding of man-in-the-middle-attacks on encryption keys.)
- Verifies, through video structural distortion and entropy analysis upon the benchmark videos, the strength of the ciphers against inference attacks. Timing results show the efficiency of implemented ciphers for the real-time streaming of privacy-preserved videos.

The remainder of this paper is organized as follows. Section 2 comments upon recent contributions to the research literature around this subject, while also discussing the research background from fuzzy rule-based systems, while setting the context of video streaming, and video SE. Then, Sect. 3 presents a higher-level view of the method, while Sect. 4 provides detailed descriptions of: the Fuzzy Inference Systems (FISs); the adopted SE scheme; together with the means of encryption, including the in-house P-XOR cipher. After that, Sect. 5 describes the results of applying the FTC with the various FISs (Mamdani and Assilian 1975). It also reveals promising visual results for the videos tested. Finally, Sect. 6 rounds off by considering the implications for those planning privacy provision for their video content streamed over an CDN or by means of another method of real-time

video streaming, which could include within an Internet-of-Things (IoT), especially if the P-XOR cipher if employed.

2 Related studies and context

This section considers recent research literature around this paper's topic. The section additionally includes discussion of the research literature on fuzzy logic systems and SE for compressed videos. There is also consideration of the current context of video streaming.

2.1 Related studies

Prior work into security threat classification has focused on security models, which are then expressed through access control policies linked to authorization methods. A recent review of those models can be found in the chapter (Badva et al. 2016). In the chapter, 20 different security models are covered, including well-known models such as the Bell–LaPadula but more recent models such as the Non-Interference Model. Though these models build-in a degree of flexibility, they do not necessarily use artificial intelligence or act in a dynamic manner, as occurs in the proposal of this paper. The reader will also find other overviews of security issues in the book of which Badva et al. (2016) is a chapter. The reader can additionally build up their knowledge further of very recent, developments in the ever-growing security field by reference to the massive compendium of chapters in Gupta et al. (2018), with an emphasis on emerging security applied to applications such as cloud computing, smartphones, wireless and mobile ad hoc networks, and IoT. Two chapters in particular are relevant to the current paper in that they seek to anticipate threats to network infrastructure and software products. Thus, in the chapter of Garg et al. (2018), dependencies between network nodes are analyzed to find how multiple vulnerabilities within a network can lead to attacks on a target node. The method of doing that, elucidated in the chapter, is through attack graph theory. Then in the chapter (Biswas and Patra 2018), the authors accept that vulnerabilities in software systems will exist but, to avoid the economic impact, if these are exploited, mitigation strategies are proposed, along with a focus on identifying potential critical vulnerabilities. It is worth remarking, that fuzzy logic is also well-suited to risk assessment of new software developments, providing indices of risk severity in Hsieh et al. (2018). Returning, to the collection of chapters, an interesting chapter also, from the point of view of the current paper (Singh et al. 2018), which details a method of trust computation in the context of Flying Ad Hoc Networks (FANETs), i.e., networks of cooperating drones. Trust management is necessary because the nodes within FANETs have limited resources.

One area of recent interest, where security is a principal concern, is that of the IoT and, in particular, the application of IoT to emerging smart cities (Plageras et al. 2017). The reason for this concern is that IoT devices are vulnerable (Rouse 2015) because of their resource constraints yet they offer an entry point to the conventional Internet. For example, in Hernandez-Ramos et al. (2015), the authors provide a layered framework for managing IoT security within a smart city. The IoT devices are intended to be the well-known ARM microcontrollers. However, the means of security management appears to be relatively conventional, based, as it is, around software managers.

The IoT is intimately connected to cloud computing, as cloud platforms provide a way of processing the data garnered by the IoT devices. The survey (Stergiou et al. 2018) considers past work on integrating the emerging IoT and cloud platforms with an emphasis on security issues. For example, the relative advantages of AES encryption and Rivest–Shamir–Adelson (RSA) asymmetric encryption in respect to the IoT, clouds, and the integration of the two are listed. It is apparent that AES block-based encryption is a relatively lightweight form of encryption, more suitable for the IoT, though it is by no means as lightweight as the XOR–based encryption algorithms considered elsewhere in the current paper. Because, RSA asymmetric encryption normally relies on a Public Key Infrastructure (PKI), which provides a hierarchy of servers to be present for the purpose of public key authentication, it is more suitable for cloud computing. Besides, RSA encryption is only normally used at the key exchange stage to encrypt the symmetric keys later used to encrypt a data stream.

In an IoT environment, security issues are indeed important to consider (Shifa et al. 2016) due to the resource-constrained nature of the devices utilized. In terms of confidentiality or privacy protection, Shifa et al. (2019) recently proposed lightweight encryption for multimedia content, reflecting the discussion in the previous paragraph. In terms of authentication, Tewari and Gupta (2017) have proposed a lightweight authentication protocol. However, in Wang et al. (2018) that protocol was apparently itself shown to be vulnerable to a key disclosure attack by virtue of the bitwise operations required of the IoT devices, RFID tags, as the number of bitwise rotation operations possible was limited to just 96. Going beyond authentication, the authos of Memos et al. (2018) consider an IoT Smart City framework for a set of multiple components that constitute security: that is authentication, access control, confidentiality, privacy, secure middleware, policy enforcement, mobile security, and trust. The application is, like the current paper, video surveillance, with compression by the HEVC codec, one of the codecs used herein. For video encryption, the authors consider SE, similar to that previously published by the authors of the current paper (Asghar and Ghanbari 2013; Asghar

et al. 2015). However, in the current paper, SE is employed by us with a light-weight cipher as an alternative to the underlying AES previously used in our past publications, to account for the resource constraint of potential devices and a need for real-time operation. Thus, SE avoids the overhead of full-encryption but beyond that AES for the SE is relatively heavyweight, as it includes successive rounds of bit manipulations, even though, with current computational capabilities, it is secure against brute-force attacks. Thus, a light-weight cipher reduces the computational overhead.

There remain potential vulnerabilities within an IoT environment. For example, at the authentication stage, two-factor authentication requires the user to have something that they know and something that they possess. For example, this might be a password and a smart card. The work in Reddy et al. (2019) echoes widespread disquiet with two-factor authentication, e.g., Kan (2019), by demonstrating that two recent two-factor protocols, Qi and Chen (2017) and Lu et al. (2016), are vulnerable to a number of attacks. For example, Qi and Chen (2017) is said to: lack user anonymity; be prone to user impersonation attacks; be vulnerable to ephemeral leakage attacks; and that an insider with access to the server can breach the authentication process. Instead, Reddy et al. (2019) provides a three-factor authentication protocol that is one that utilizes a third set of biometric credentials (or credentials associated with the user's environment such as ambient noise). After cryptanalysis of the protocol, the multiple gains of the proposed three-factor authentication are demonstrated by comparison with two-factor schemes in the literature.

2.2 Fuzzy logic systems

Fuzzy logic describes the fuzziness or lack of precision in data with the help of fuzzy sets. Fuzzy-based classifiers are developed on the basis of membership functions and FIS rules. FIS rules, which themselves reflecting common sense, allow effective reasoning, akin to that of an expert in the field (Zadeh 2015). Fuzzy-based classifiers utilize mathematical principles as a means of knowledge representation and the degree of membership. Therefore, one can say that they lean towards numerical processing of data. Such classifiers are developed with multi-valued logical values, rather than binary values, as fuzzy logic uses a continuum of logical values between 0 (false) and 1 (true). A FIS is constructed based upon linguistic variables, as would be employed by an expert in the field, together with eliciting from an expert the fuzzy rules that operate on the linguistic variables. A fuzzy inference engine is constructed by means of composition of those fuzzy rules. After the FIS has been applied, the process of defuzzification converts the still fuzzy output to crisp or non-fuzzy output. Thus, it is necessary to specify rules of defuzzification. The simplicity in the implementation process for a fuzzy rule-based system, even with numerous parameters of digital devices, network characteristics and the requirements of real-time applications, made it an effective way for us to develop a classifier.

Device capability enters into our FIS, and in an IoT real-time interconnections between heterogeneous and ubiquitous devices has been one of the most important concerns (Li et al. 2015a). Collotta and Pau (2015) implemented fuzzy logic to determine the sleeping time of the devices according to the battery level and to the ratio of throughput to workload in the smart home for efficient power management.

Fuzzy system engineering has also been adopted for confidential data transfer over a shared network. Gandotra et al. (2017) presented a fuzzy system for the better performance of supervised algorithms to automatically detect malware. Ashfaq et al. (2017) considered intrusion detection systems and presented semi-supervised learning algorithms, using a divide-and-conquer approach and then categorizing the intrusion threat according to the magnitude of the fuzziness. Their future work involves detecting multi-type attacks, with the help of fuzzy logic. Similarly, Mudia and Chavan (2015) proposed a fuzzy-based automatic image-encryption system for secret sharing on the Internet. Recently, Cuka et al. 2019 compared two fuzzy-based systems for the selection of IoT nodes in the context of opportunistic networks (ones in which nodes are only connected temporarily). They found that there was a trade-off between the two fuzzy systems compared in terms of complexity and the suitability of the selection. Notice that similarly, the proposed system in this paper also involves an element of selection, as it uses a hierarchical fuzzy-based systems for the selection of privacy levels for the real-time delivery of multimedia.

Elsewhere, fuzzy logic is particularly useful when a dynamic response is required, as it can react in real-time and requires reduced training, as it can take into account the reactions of the user themselves. For example, in Ribino and Lodato (2019), the user's physiological signs, that is heart rate and variability in the fuzzy model. In addition, another input is gathered through a mobile device carried by the user, namely the acoustic noise level. The fuzzy modelling then permits an individual's at-risk level to be assessed, without the need for the individual themselves to report a problem during a dangerous event. Likewise the authors (Rainer et al. 2018), a robotic museum tour guide, which makes expressive oral presentations at given positions on its route, is dynamically trained through fuzzy models. The training is controlled by feedback from members of the public participating on a tour in terms of what went well and what was less successful during the presentations. Adjustments to the fuzzy rules depend on the age and cultural attributes of the robotic guide's audience.

Fuzzy logic has, in fact, been used in CDNs for performance improvement. For example, Chen and Liao 2010

presented a fuzzy-based decision system for multimedia-content request routing in a CDN to decrease the drop rate and improve the network utilization. Roy et al. 2015 also used fuzzy concepts to dynamic select edge servers and load balance CDNs associated with a cloud. In the scheme, an edge server is found that is nearest to the end-user's location and has the lowest response time and load. However, it must be said that the current paper is the first time that fuzzy logic is proposed to allow CDNs to select a privacy level for the end-users by considering multiple selection inputs for each FIS.

2.3 Video streaming context

According to CISCO predictions, extrapolating from current network statistics, video streaming will take up 82% of the total network traffic by the year 2020 (Cisco 2018). From the security point-of-view it is possible to distinguish as least three classes of video traffic that will contribute to this increase. Each of these classes has different confidentiality requirements (or more widely privacy levels when one considers other aspects of personal security), which is as well, because, as shown in this paper, encrypting video potentially has costs in terms of a bitrate overhead and the computational overhead. For resource-challenged mobile devices communicating over bandwidth-limited wireless connections or networks, these costs may be significant and may affect the commercial competitiveness of an RTA, if there is a choice between rival versions of an application.

In fact, CDNs can stream to end users many kinds of video (Stocker et al. 2017), ranging from personal to public videos. For example, there is: video chatting; video conferencing (Webex, webcast, webinars); VoD; social media live chats (Facebook live, Instagram live and so on); social media stored videos; live Internet TV; and stored media streaming applications, such as from Netflix, YouTube, Twitch, Hulu. Amazon Prime and so on. Banks of transcoders in a cloud may be employed to format video so that in (say) a video call, the video is compatible between different codecs. CDN operators have a heavy responsibility for maintaining the privacy of individuals in their personal and semi-personal video communication. To ensure the confidentiality of CDN transmission, encryption is an effective procedure (Long et al. 2018). While the implementation of encryption over streaming videos is a challenging task, because of the trade-off between encryption bitrate overhead and efficient transmission over CDNs and on to their end-users. There is also a storage cost, for pre-stored video in particular, if a type of HTTP Adaptive Streaming (HAS) (Bentaleb et al. 2018) is used, because there are typically five or more versions of a video clip or TV program at different quality levels, according to network congestion levels (Seufert et al. 2015). Implementation of full encryption for real-time streaming of video may not be practical due to the computational cost of encryption leading to latency, especially if high-resolution video, typically 1080 progressive (p) video, is streamed.

2.4 Selective encryption within video encoders

Codecs play a vital role in video streaming, as few networks have sufficient available bandwidth to support uncompressed streaming. In terms of codecs, the H.264/Advanced Video Coding (AVC) (Wiegand et al. 2003) codec remains widely deployed on streaming servers (such as those of YouTube), due to its ease of software deployment and the range of supported hardware for rapid compression and decompression. More recently, the High Efficiency Video Coding (HEVC) codec (Sullivan et al. 2012) provides a 30–45% improvement in the compression ratio over H.264/AVC. However, hardware has been slow to support HEVC (Ram and Panwar 2017), possibly due to the complexity of a full implementation. This is the main reason that, since its release in 2013, HEVC has not been adopted as widely as H.264/AVC. However, there is growing awareness of its potential (Li et al. 2015b). Due to the involvement of hardware companies like AMD, ARM, Intel and NVIDIA, HEVC hardware support is expected to be available within 1–2 years' time. Furthermore, owing to the rapid adaptation of CDNs like those of Amazon, Google, Hulu and Netflix, HEVC is expected to be used by major content distributors. Both YouTube and Netflix have already stated that they intend to implement it within months (Reddit 2018). Therefore, HEVC is considered to be a strong contender in the future (Ohm and Sullivan 2013). Though in this paper, standardized codecs are utilized because these are highly favored by commercial companies, it is possible, if standardization is not an issue, to replace the frequency transform that is an essential stage in hybrid image codecs by a fuzzy transform. Typical frequency transforms are the Discrete Cosine Transform or the Discrete Wavelet Transform, as employed in the JPEG series of still-image codecs. For example, in Martino and Sessa (2018), a multi-level fuzzy transform is shown to be competitive in time with JPEG compression, though decoded image quality currently is not competitive. In Martino and Sessa (2018), the authors consider their multi-level scheme has the ability, through built-in quality thresholding, to become quality competitive.

Selective encryption (SE) as applied herein takes advantage of entropy coding. Entropy engines comprise the last stage of hybrid video codecs, being used to remove any remaining statistical redundancy (Ghanbari 2003). H.264/Advanced Video Coding (AVC) and its Scalable Video Coding (SVC) extension (Schwarz et al. 2007) utilize the same alternative two entropy coding modes, i.e., Context Adaptive Variable Length Coding (CAVLC) (Chen et al. 2006) and Context Adaptive Binary Arithmetic Coding

(CABAC) (Marpe et al. 2003). Within H.264/AVC either CABAC or CAVLC entropy engine can be selected, as the two coders support a trade-off between compression efficiency and computational complexity (Wang et al. 2013). The HEVC CABAC encoder is a somewhat modified version of the H.264/AVC CABAC coder but there is no alternative CAVLC coder. Thus, HEVC is confined to CABAC coding, which achieves about 15% greater compression than CAVLC but is significantly more complex.

SE itself is commonly integrated within a codec (Furht et al. 2005), particularly as encrypting video before compression usually removes any exploitable redundancy. SE considers the most significant information (as regards distortion) with a choice of different stages of a hybrid codec, such as on the original pixels, the transform coefficients, the quantization indexes and the bit-planes for the encryption (Massoudi et al. 2008). By leaving encryption to the last stage of such an encoder, SE is less likely to upset the compression achieved in earlier stages of encoding. Entropy-integrated SE for the standardized H.264/AVC, H.264/SVC, and HEVC codecs with the CABAC engine has already been proposed by various researchers, that is Shahid et al. (2011), Asghar and Ghanbari (2013), and Shahid and Puech (2014), respectively. SE is applied specifically to encrypt a set of entropy bins, the bits output as part of entropy coding, that pass through the bypass mode. The bypass mode is the coding mode that does not use context modelling to vary the coding rate. The reason for this design is so as to achieve a similar bit rate in the statistical sense to that without encryption and also to maintain decoder format compliance (Asghar and Ghanbari 2013; Shahid and Puech 2014). As codec standardization is controlled by the format of the compressed bitstream entering a decoder, it is important to ensure the encrypted bitstream does not break the detailed specification of a conformant bitstream.

One should mention, that an alternative to SE is Region-of-Interest (ROI) encryption in which some parts of a video are protected to reduce the encryption overhead and computational complexity (Farajallah et al. 2015; Peng et al. 2013). However, ROI encryption is application specific, while SE potentially offers a more general solution. Besides, SE provides format complaint encrypted bitstream at the decoder, compared to either full or complete encryption of the video stream or ROI encryption. Decoder format compatibility allows video to be processed at intermediate points in a network, e.g., through transcoding, without the need for the decryption key to be available at those points. Additionally, SE can support interoperability (Asghar et al. 2017) in which multiple encryptions of the same video stream are transported. Furthemore, it has a potential role in consumer electronics applications (Lookabaugh and Sicker 2004). Thus, applying encryption at the entropy coding stage minimizes the problems of full and ROI encryption, which is why that form of encryption is chosen for this paper.

In the proposed FTC-based method, SE is applied with two alternative ciphers: (1) AES in Cipher Feed Back mode (CFB) mode (National Institute of Standards and Technology 2001), and the in-house P-XOR. Both encrypt bits selected from the entropy-coded bitstream, which is coded with the CABAC entropy coder. The proposed FTC method resolves the key problem tackled in this paper, which is choosing an appropriate cipher to match the differing confidentiality requirements of end-users and/or content providers.

3 Overall method

The overall FTC for the proposed video streaming method is comprised of the following two components.

Classifier This is developed through Fuzzy sets, rules and the inference system for real-time detection of confidentiality requirements for end-users.

Encryption module SE is applied over streamed videos through either P-XOR or AES-CFB encryption.

The FTC method itself can be understood through the flow chart given in Fig. 1, with the classifier extending to the 'Input Video' box and the encryption module being below that box.

3.1 Classifier overview

In the Fuzzy Classifier, a total of three FISs were designed and implemented in two layers. Three layer-1 processes are shown in Fig. 1. Thus, layer-1 comprises of three FISs operating in parallel; these are (1) Device Specifications (DS′) with input variables: Energy (battery time), Storage, and Screen-size of each device; (2) Network Specifications (NS′), with input variables being Bandwidth, Throughput, and Channel-Quality (bit-error rate); and (3) Real-time Application (RA′) with input variables being confidential (personal), protected (semi-personal) and public (impersonal). These three Layer-1 FISs have twelve crisp (non-fuzzy) inputs (three for each FIS) and each FIS produces a single crisp output.

Layer-1's three outputs become the crisp inputs for the layer-2 FIS, as is apparent from Fig. 2, which shows the layering more clearly than Fig. 1. In Layer-2, the active FIS is Privacy Level (PL′), which produces three types of outputs, i.e., low, medium and high privacy requirements for the streaming videos. The single crisp output (after defuzzification) from layer-2 will decide on the confidentiality level of the encryption module.

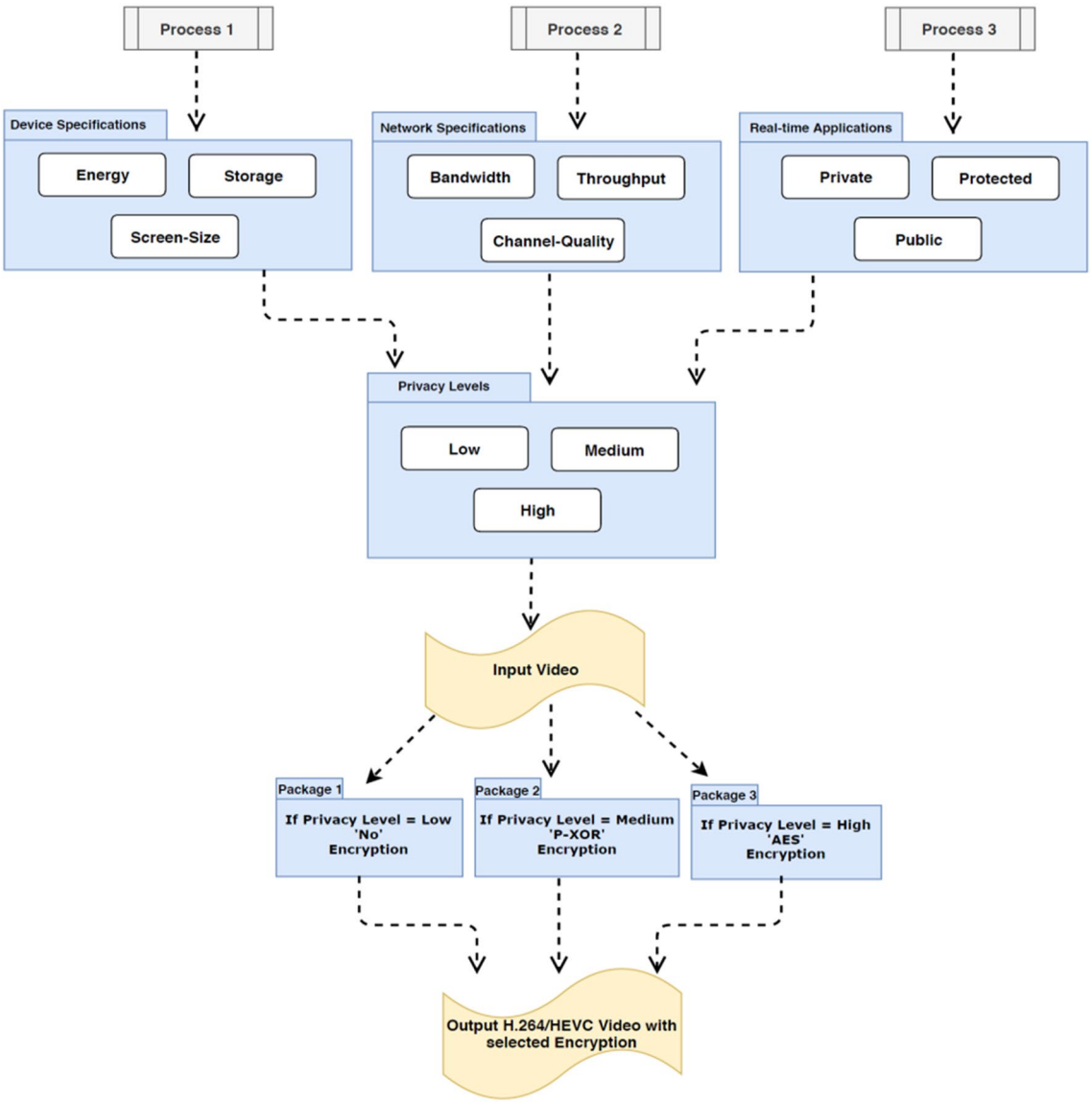


Fig. 1 FTC-based method

3.2 Encryption module overview

The encryption module performs SE on compressed H.264/AVC or HEVC streamed videos. In Fig. 1, depending on the determination of the type of encryption, one of the three encryption packages is chosen (no-encryption, P-XOR encryption, or AES encryption in Fig. 1). Depending on that choice, selected syntax or parameter elements from the compression process are encrypted according to the chosen form of encryption.

Because entropy coding is the last compression stage of a hybrid encoder such as H.264/AVC or HEVC, the bin-strings (sequences of bits output by an entropy coder) selected at this stage are quickly encrypted and do not greatly disturb the compression ratio of the encoders. That is to say the bitrate is not significantly increased as a result of encryption. The entropy engine used by the two codecs, i.e., H.264/AVC or HEVC for better compression, is known as Context Adaptive Binary Arithmetic Coding (CABAC) (Sze and Budagavi 2013). Some CABAC bin-strings are selected to be encrypted in the SE process. The form of encryption is according to the choice of the FTC.

In general, the steps by the CABAC coder are: (1) Binarization, (2) Context Modeling (CM), and (3) Binary Arithmetic Coding (BAC). In the initial binarization, all non-binary syntax elements are converted to bin-strings. A bin is a bit position in each bin-string that is passed to a coding mode decision module (Asghar and Ghanbari 2013). There are two types of BAC coding mode decision: one coding mode is BAC-regular and the other is called BAC-bypass coding

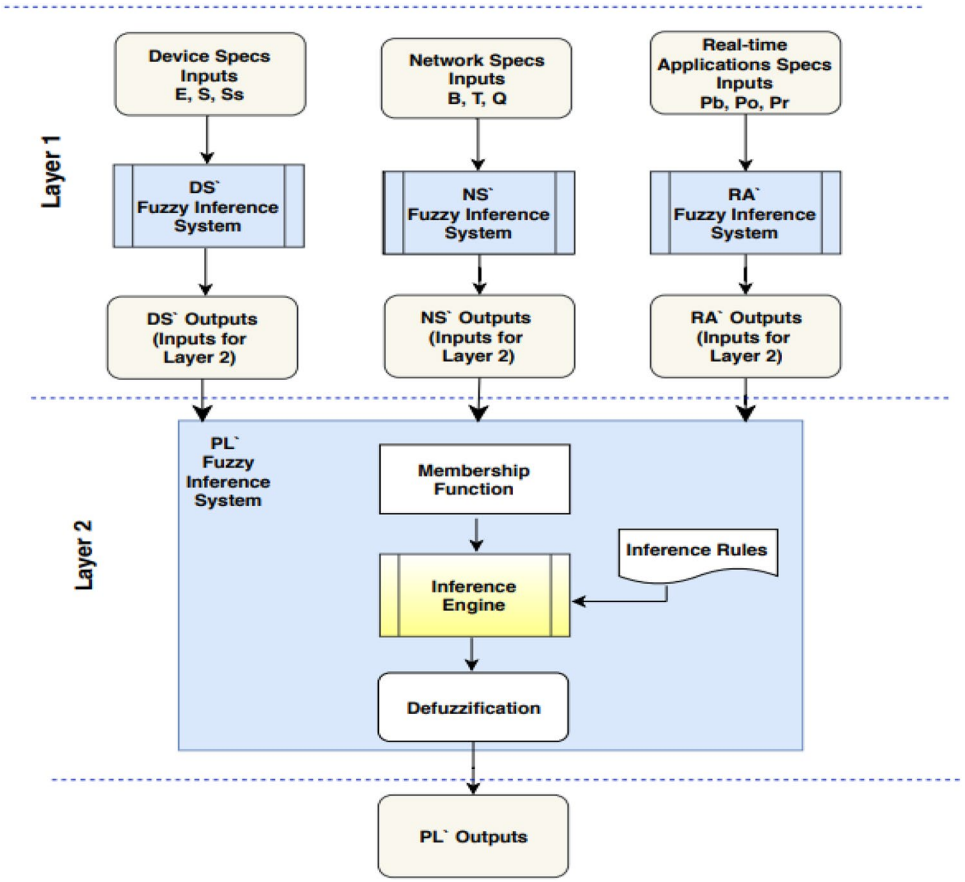


Fig. 2 Functionality of the fuzzy classifier

mode. After passing to BAC-regular coding, bins are transferred to the next step, which is CM based on the probability distribution of that bin. After that, the BAC-regular engine performs coding. When the bins are passed to BAC-bypass coding mode then the CM step is skipped and the bins directly move towards the BAC engine for coding. This process of CABAC entropy coder is shown in Fig. 3. Notice that only the by-passed bins (given in the green-colored box) are used for applying SE because, as previously mentioned, CM is thereby not affected and, hence, the prediction statistics are not affected. These bins are the arithmetic sign information of the Motion Vector Differences (MVDs) and the sign of the residual transform coefficients (TC) levels (known as texture in the coding community) or some other

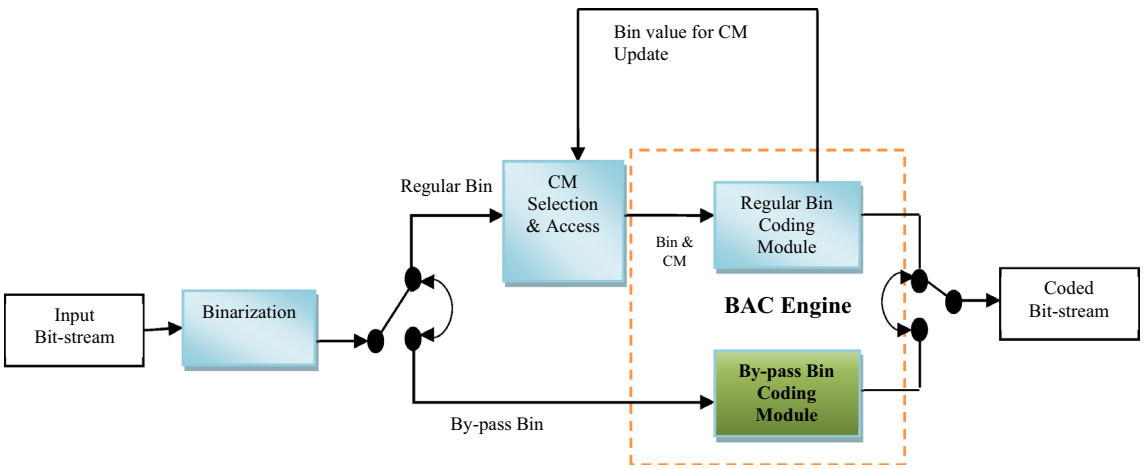


Fig. 3 Process of selecting CABAC bin-strings for SE

less significant bits, which are also thought to be uniformly distributed.

In summary, in the proposed FTC's encryption module, the arithmetic signs of MVDs and the residual TCs at different levels are chosen for SE. The SE of these selected bin-strings has already been shown to be format compliant and compression-friendly for real-time applications (Asghar et al. 2015). For further details about the chosen bin-strings for SE, refer to the papers Asghar and Ghanbari 2013 and Shahid and Puech 2014.

4 Detailed description of the method

In this Section, the high-level overview of the two components of the method are now described in more detail.

4.1 Classifier: detailed description

The description of the classifier module is now broken down into its constituent parts, namely the fuzzy sets employed; the mathematical model for those sets; their representation as membership functions; and lastly, their implementation as IF–THEN rules, along with the means of defuzzification.

4.1.1 Fuzzy sets employed

To construct the rules for the FIS: DS′, three types of devices were considered, namely mobile or smart phones, tablets, and laptops. As an illustration, the actual specifications of devices used for the implementation of FIS: DS′ are given in Table 1. For Network specifications, three types of network connectivity for transmission, i.e., 2G, 3G/4G and LTE/5G were considered. While for RTAs, three types of inputs were considered: Private RTA (e.g., personal videos chats, and VoD); Protected RTA (e.g., group conferences); and Public RTA (publically-available video clips on YouTube and other platforms). The final PL selection has three inputs coming from layer-1 and produces three outputs (i.e., low, medium and high for each streamed video).

Therefore, for the implementation of the Classifier, there were four main fuzzy sets with their linguistic variables (i.e. DS′, NS′, RA′ and PL′), three layer 1 sets and one layer 2 set, along with twelve subsets (three for each set). A combination of fuzzy subsets from each Layer-1 fuzzy set (DS′, NS′, RA′) are used for the final Layer-2 decision. Usually a fuzzy set is defined as A = {y: p(y)}. The fuzzy sets are now given below.

1. Device Specifications (DS) for particular device selection
   DS′ = {y: y are the device specification parameters}
   DS′ = {*Energy (E), Storage (S), Screen Size (Ss)*}
   where each subset E, S, Ss = {Fair (F), Good(G), Excellent (Ex)}
2. Network Specifications (NS) for checking transmission compatibility
   NS′ = {y: y are the Network specification parameters}
   NS′ = *{Bandwidth (B), Throughput(T), Channel-Quality(Q)}*
   where each subset B, T, Q = {Fair (F), Good(G), Excellent (Ex)}
3. Real-time Applications (RA)
   RA′ = {y: y are the real-time application parameters}
   RA′ = *{Private (Pr), Protected(Po), Public(Pb)}*
   where each subset Pb, Po, Pr = {Fair (F), Good(G), Excellent (Ex)}

The fuzzy set and the term set defined for the layer 2 linguistic varaible PL′ are given below:

4. Privacy Level (PL)
   PL′ = {y: y are the output parameters}
   PL′ = { Low(L), Medium (M), High (H)}

4.1.2 Mathematical model of fuzzy sets

Letting U̓ be the Universe of Discourse, the proposed fuzzy sets are then represented as:

Table 1 Devices with their specifications

Device type	Device names	Energy (battery time in hours)	Storage (in GB)	Screen diagonal size (in inches)
Mobile phones	iPhone-4	6	64	3.5
	iPhone 6	10	128	5.5
	iPhone 7	15	256	5.5
Tablets	Apple ipad-5	10	64	9.7
	Samsung tab 7 plus	6	32	7
Laptops	HP-15-AY540	6	1024	15.6
	Dell-XPS 13	6	512	13.3
	Dell-XPS 17	10	512	17

$$\mathrm{DS}' = \{(y, \mu_{\mathrm{DS}'}(y)) \,|y \in \mathrm{U}'\}, \tag{1}$$

$$\mathrm{NS}' = \{(y, \mu_{\mathrm{NS}'}(y)) \,|y \in \mathrm{U}'\}, \tag{2}$$

$$\mathrm{AS}' = \{(y, \mu_{\mathrm{RA}'}(y)) \,|y \in \mathrm{U}'\}, \tag{3}$$

$$\mathrm{PL}' = \{(y, \mu_{\mathrm{PL}'}(y)) \,|y \in \mathrm{U}'\}, \tag{4}$$

where the $\mu\mathrm{DS}'(y)$, $\mu\mathrm{NS}'(y)$ and $\mu\mathrm{AS}'(y)$ are the degrees of membership of y, assuming values in the range 0–1. Thus, one can say that

$$\{\mu_{\mathrm{DS}\}}(y), \mu_{\mathrm{NS}\}}(y), \mu_{\mathrm{RA}\}}(y)\mu_{\mathrm{PL}\}}(y)\} \in [0, 1] \tag{5}$$

and the fuzzy subsets are:

$$\{\mu_{\mathrm{E}'}(y), \mu_{\mathrm{S}'}(y), \mu_{\mathrm{Ss}'}(y)\} \in [0, 1], \tag{6}$$

$$\{\mu_{\mathrm{E}'}(y), \mu_{\mathrm{T}'}(y), \mu_{\mathrm{Q}'}(y)\} \in [0, 1], \tag{7}$$

$$\{\mu_{\mathrm{Pr}'}(y), \mu_{\mathrm{Po}'}(y), \mu_{\mathrm{Pb}'}(y)\} \in [0, 1]. \tag{8}$$

The following are representations of the fuzzy sets, when U′ is a finite and discrete form of data:

$$DS' = \left\{ \frac{\mu\mathrm{DS}'(\mathrm{y1})}{y1} + \frac{\mu\mathrm{DS}'(\mathrm{y2})}{y2} + \frac{\mu\mathrm{DS}'(\mathrm{y3})}{y3} \ldots \right\} \tag{9}$$

$$DS' = \sum_{i=1}^{n} \left(\frac{\mu\mathrm{DS}'(yi)}{yi} \right) \tag{10}$$

The subset E′ of DS′ is represented as:

$$E' = \left\{ \frac{\mu\mathrm{E}'(\mathrm{y1})}{y1} + \frac{\mu\mathrm{E}'(\mathrm{y2})}{y2} + \frac{\mu\mathrm{E}'(\mathrm{y3})}{y3} \ldots \right\} \tag{11}$$

The values of elements belonging to E′ are computed in Eq. (11). Thus, one arrives at Eqs. (12), (13), and (14).

$$E' = \sum_{i=1}^{n} \left(\frac{\mu\mathrm{E}'(yi)}{yi} \right), \tag{12}$$

$$S' = \sum_{i=1}^{n} \left(\frac{\mu\mathrm{S}'(yi)}{yi} \right), \tag{13}$$

$$Ss' = \sum_{i=1}^{n} \left(\frac{\mu\mathrm{Ss}'(yi)}{yi} \right) \tag{14}$$

The equations for other sets and subsets are represented as:

$$NS' = \left\{ \frac{\mu\mathrm{NS}'(\mathrm{y1})}{y1} + \frac{\mu\mathrm{NS}'(\mathrm{y2})}{y2} + \frac{\mu\mathrm{NS}'(\mathrm{y3})}{y3} \ldots \right\} = \sum_{i=1}^{n} \left(\frac{\mu\mathrm{NS}'(yi)}{yi} \right) \tag{15}$$

$$B' = \sum_{i=1}^{n} \left(\frac{\mu\mathrm{B}'(yi)}{yi} \right), \tag{16}$$

$$Q' = \sum_{i=1}^{n} \left(\frac{\mu\mathrm{Q}'(yi)}{yi} \right), \tag{17}$$

$$T\} = \sum_{i=1}^{n} \left(\frac{\mu\mathrm{T}'(yi)}{yi} \right), \tag{18}$$

$$RA' = \left\{ \frac{\mu\mathrm{RA}'(\mathrm{y1})}{y1} + \frac{\mu\mathrm{RA}'(\mathrm{y2})}{y2} + \frac{\mu\mathrm{RA}'(\mathrm{y3})}{y3} \ldots \right\} = \sum_{i=1}^{n} \left(\frac{\mu\mathrm{RA}'(yi)}{yi} \right) \tag{19}$$

$$Pb' = \sum_{i=1}^{n} \left(\frac{\mu\mathrm{Pb}'(yi)}{yi} \right), \tag{20}$$

$$Po' = \sum_{i=1}^{n} \left(\frac{\mu\mathrm{Po}'(yi)}{yi} \right), \tag{21}$$

$$Pr' = \sum_{i=1}^{n} \left(\frac{\mu\mathrm{Pr}'(yi)}{yi} \right) \tag{22}$$

$$PL' = \left\{ \frac{\mu\mathrm{PL}'(\mathrm{y1})}{y1} + \frac{\mu\mathrm{PL}'(\mathrm{y2})}{y2} + \frac{\mu\mathrm{PL}'(\mathrm{y3})}{y3} \ldots \right\} = \sum_{i=1}^{n} \left(\frac{\mu\mathrm{PL}'(yi)}{yi} \right) \tag{23}$$

The cardinality of the fuzzy sets is:

$$|\mathrm{DS}'| = |\mathrm{NS}'| = |\mathrm{RA}'| = |\mathrm{PL}'| \tag{24}$$

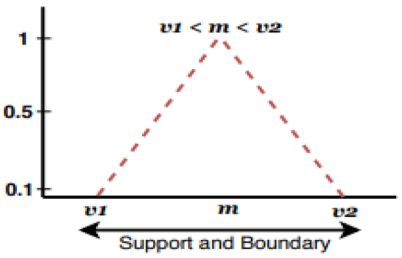
Then, the subsets of the layer 1 sets are represented as:

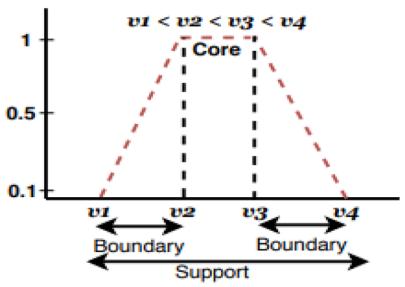
$$\mathrm{E},\ \mathrm{S},\ \mathrm{Ss} \subseteq \mathrm{DS}' \tag{25}$$

$$\mathrm{B},\ \mathrm{T},\ \mathrm{Q} \subseteq \mathrm{NS}' \tag{26}$$

$$\mathrm{Pb},\ \mathrm{Po}, \mathrm{Pr} \subseteq \mathrm{RA}' \tag{27}$$

In each fuzzy set, one has all finite subsets with specific ranges, which can be written as:





$$\mu PL`(y)=\begin{cases}\frac{y-v1}{v2-v1}, & v1 < y \le v2\\ \frac{y-v2}{m-v2}, & v2 < y \le m\\ 0, & Otherwise\end{cases} \quad (32)$$

The parameters $v1$ and $v2$ locate the base of the triangle and the parameter m locates the peak, where $v1 < m < v2$.

$$\mu PL`(y)=\begin{cases}0, & y < v1 \; or \; y > v4\\ \frac{y-v1}{v2-v1}, & v1 \le y \le v2\\ 1, & v2 \le y \le v3\\ \frac{v4-y}{v4-v3}, & v3 \le y \le v4\end{cases} \quad (33)$$

The parameters are defined by a lower limit $v1$, an upper limit $v4$, a lower support limit v2, and an upper support limit $v3$, where $v1 < v2 < v3 < v4$.

Fig. 4 Characteristics of triangular and trapozoidal membership functions

Table 2 Computed values for the fuzzy subsets of fuzzy set DS′

Energy (in hours)	Storage (GB)	Screen size (inches)	Term set	Degree of membership
6–10	512, 1024	13.3, 15.6, 17	Excellent (Ex)	0.7–1
2–7	32, 64, 128	7, 9.7, 10	Good (G)	0.3–0.7
0–3	8, 16	4, 5.5, 6	Fair (F)	0–0.3

$$|E| = |S| = |Ss| \quad (28)$$

$$|B| = |T| = |Q| \quad (29)$$

$$|Pb| = |Po| = |Pr| \quad (30)$$

The degree of truth in fuzzy logic is presented through membership functions denoted with a dot notation, (•).

$$\mu_{DS'}(\cdot),\ \mu_{NS'}(\cdot),\ \mu_{RA'}(\cdot),\ \mu_{PL'}(\cdot) \in [0, 1] \quad (31)$$

4.1.3 Membership functions

Membership functions represent the fuzzy sets. To get accurate results, a combination of triangular and trapozidal membership functions were used in the Classifier (Pedrycz 1994). The characteristics of these two types of membership function of these two types of function are represented grapphically in Fig. 4, with the guiding equations given by (32) and (33).

Then, Tables 2, 3, 4, 5, 6, 7, 8 show the degrees of membership functions with associated fuzzy rules for Layer-1 and Layer-2 linguistic variables. In summary, the number of computed fuzzy rules is:

$$\text{No. of Rules} = m^v, \quad (34)$$

Table 3 Layer-1 fuzzy rules for DS′ (D1 = laptop, D2 = tablet, D3 = mobile/smartphone)

Rule #	E	S	Ss	Device type	Rule #	E	S	Ss	Device type	Rule #	E	S	Ss	Device type
1	Ex	Ex	Ex	D1	10	G	Ex	Ex	D1	19	F	Ex	Ex	D1
2	Ex	Ex	G	D2	11	G	Ex	G	D2	20	F	Ex	G	D2
3	Ex	Ex	F	D3	12	G	Ex	F	D3	21	F	Ex	F	D3
4	Ex	G	Ex	D1	13	G	G	Ex	D1	22	F	G	Ex	D1
5	Ex	G	G	D2	14	G	G	G	D2	23	F	G	G	D2
6	Ex	G	F	D3	15	G	G	F	D3	24	F	G	F	D3
7	Ex	F	Ex	D1	16	G	F	Ex	D1	25	F	F	Ex	D1
8	Ex	F	G	D2	17	G	F	G	D2	26	F	F	G	D2
9	Ex	F	F	D3	18	G	F	F	D3	27	F	F	F	D3

Table 4 Computed values of the fuzzy subsets of fuzzy set NS′

Bandwidth (Mhz)	Throughput (kbps)	Channel-quality (BER ratio) (%)	Term set	Degree of membership
14	120,40	86–95	Excellent (Ex)	0.7–1
12	8,10	70–85	Good (G)	0.3–0.7
10	2,4,6	61–70	Fair (F)	0–0.3

Table 5 Layer-1 fuzzy rules for NS′ (Ntw1 = LTE, 5G, Ntw2 = 3G,4G, Ntw3 = 2G)

Rule #	B	T	Q	Network type	Rule #	B	T	Q	Network type	Rule #	B	T	Q	Network type
1	Ex	Ex	Ex	Ntw1	10	G	Ex	Ex	Ntw1	19	F	Ex	Ex	Ntw1
2	Ex	Ex	G	Ntw2	11	G	Ex	G	Ntw2	20	F	Ex	G	Ntw2
3	Ex	Ex	F	Ntw3	12	G	Ex	F	Ntw3	21	F	Ex	F	Ntw3
4	Ex	G	Ex	Ntw1	13	G	G	Ex	Ntw1	22	F	G	Ex	Ntw1
5	Ex	G	G	Ntw2	14	G	G	G	Ntw2	23	F	G	G	Ntw2
6	Ex	G	F	Ntw3	15	G	G	F	Ntw3	24	F	G	F	Ntw3
7	Ex	F	Ex	Ntw1	16	G	F	Ex	Ntw1	25	F	F	Ex	Ntw1
8	Ex	F	G	Ntw2	17	G	F	G	Ntw2	26	F	F	G	Ntw2
9	Ex	F	F	Ntw3	18	G	F	F	Ntw3	27	F	F	F	Ntw3

Table 6 Computed values of the fuzzy subset of fuzzy set RA′

Real-time applications	Term set	Degree of membership
Private (Personal Video calls, Facebook Live, Instagram Live, VoD, Webcasting etc.)	Excellent (Ex)	0.7–1
Protected (Group Conferencing (Webex, Webinars, training videos) Live TV etc.)	Good (G)	0.3–0.7
Public (YouTube, Netflix, Vimeo, Twitch, Hulu, Daily motion etc.)	Fair (F)	0–0.3

Table 7 Layer-1 fuzzy rules for RA′ (A1 = personal videos sharing, A2 = group video sharing, A3 = public videos)

Rule #	Pb	Po	Pr	Application type	Rule #	Pb	Po	Pr	Application type	Rule #	Pb	Po	Pr	Application type
1	Ex	Ex	Ex	A1	10	G	Ex	Ex	A1	19	F	Ex	Ex	A1
2	Ex	Ex	G	A2	11	G	Ex	G	A2	20	F	Ex	G	A2
3	Ex	Ex	F	A3	12	G	Ex	F	A3	21	F	Ex	F	A3
4	Ex	G	Ex	A1	13	G	G	Ex	A1	22	F	G	Ex	A1
5	Ex	G	G	A2	14	G	G	G	A2	23	F	G	G	A2
6	Ex	G	F	A3	15	G	G	F	A3	24	F	G	F	A3
7	Ex	F	Ex	A1	16	G	F	Ex	A1	25	F	F	Ex	A1
8	Ex	F	G	A2	17	G	F	G	A2	26	F	F	G	A2
9	Ex	F	F	A3	18	G	F	F	A3	27	F	F	F	A3

Table 8 Layer-2 Fuzzy Rules for PL′

Rule #	DS′	NS′	RA′	PL′	Rule #	DS′	NS′	RA′	PL′	Rule #	DS′	NS′	RA′	PL′
1	D1	Ntw1	A1	High	10	D2	Ntw1	A1	High	19	D3	Ntw1	A1	High
2	D1	Ntw1	A2	Medium	11	D2	Ntw1	A2	Medium	20	D3	Ntw1	A2	Medium
3	D1	Ntw1	A3	Low	12	D2	Ntw1	A3	Low	21	D3	Ntw1	A3	Low
4	D1	Ntw2	A1	High	13	D2	Ntw2	A1	High	22	D3	Ntw2	A1	High
5	D1	Ntw2	A2	Medium	14	D2	Ntw2	A2	Medium	23	D3	Ntw2	A2	Medium
6	D1	Ntw2	A3	Low	15	D2	Ntw2	A3	Low	24	D3	Ntw2	A3	Low
7	D1	Ntw3	A1	High	16	D2	Ntw3	A1	High	25	D3	Ntw3	A1	High
8	D1	Ntw3	A2	Medium	17	D2	Ntw3	A2	Medium	26	D3	Ntw3	A2	Medium
9	D1	Ntw3	A3	Low	18	D2	Ntw3	A3	Low	27	D3	Ntw3	A3	Low

where m represents the number of membership functions and v represents the number of linguistic variables. There are three variables in each fuzzy set with three membership functions. Therefore, each fuzzy set in the Classifier has the following number of rules:

$$\text{No. of Rules } = 3^3 = 27 \tag{35}$$

4.1.4 IF–THEN rule implementation and defuzzification

The Classifier implements the rules in qualitative reasoning mode. Thus, the input–output relationship of the system is expressed as a collection of fuzzy IF–THEN rules. The following are a few examples of the rules for detecting an appropriate confidentiality or privacy level.

Layer-1 Fuzzy Rules for DS′, NS′ and RA′ Fuzzy Sets

(Device Specifications)

1. **IF** (E **IS** G) **AND** (S **IS** Ex) **AND** (SS **IS** F) **THEN** select device D3 (Table 3: Rule # 13)

(Network Specifications)

2. **IF** (B **IS** F) **AND** (T **IS** G) **AND** (Q **IS** Ex) **THEN** select network NTW1 (Table 5: Rule # 22)

(Real-Time Application)

3. **IF** (Pb **IS** F) **AND** (Po **IS** Ex) **AND** (Pr **IS** G) **THEN** select application A2 (Table 7: Rule # 20)

Layer-2 Fuzzy Rules for PL′ Fuzzy Set (from Table 8)

1. **IF** (DS **IS** D1) **AND** (NS **IS** Ntw1) **AND** (RA **IS** A1) **THEN** High PL Required
2. **IF** (DS **IS** D2) **AND** (NS **IS** Ntw1) **AND** (RA **IS** A2) **THEN** Medium PL required)
3. **IF** (DS **IS** D3) **AND** (NS **IS** Ntw2) **AND** (RA **IS** A3) **THEN** Low PL Required

All FISs use the defuzzification technique known as center of mass/gravity. In this rounding off technique, the fuzzy centroid is calculated as a final defuzzified value. Mathematically, the overall Layer 1 and Layer 2 defuzzified output O_n is represented as:

$$O_n = \frac{\int \mu y'(i).idi}{\int \mu y'(i).idi}, \tag{36}$$

where n represents the 1, 2, 3, 4 outputs of the four FISs, i repesents the fuzzy set, and μy′ = μDS′,μNS′,μRA′ or μPL′.

We now pass-on to a detailed description of the second of the two modules of the proposed method.

4.2 Encryption module: detailed description

This section elaborates the implemented encryption module, which will perform SE on the compressed H.264/AVC or HEVC streamed videos. Previous Sect. 3.2 outlined how it is within the entropy coding stage of one of those hybrid codecs that the parameters are selected for encryption, leaving the remainder of the compressed video stream unencrypted. The inclusion of the encrypted parameters within their original positions in the compressed video bitstream has two main consequences: (1) a watchable version of the original video should not be obtainable through decompression or decoding without access to the encryption key and subsequent decryption; and (2) because selective encryption of the parameters is performed, rather than full encryption; the bitrate overhead should be statistically unchanged, provided suitable parameters are chosen, i.e., ones that have a uniform distribution. Justification of the parameters selected can be found in Asghar and Ghanbari (2013) and Shahid and Puech (2014), as previously mention in Sect. 3.2.

Pseudo-code of simulation performed for Encryption module

```
Input: Video, Privacy Level (High, Medium ,Low)
Output: Privacy Protected Video
switch (PL`)
            case  Low_PL`:
            break;
            case  Medium_ PL`:
               if (Signs-MVD && residual_Coeff)
                     DH-Key_Exchange();
                     MVD.Enc_P-XOR();
                    Texture.Enc_P-XOR();
                 End if
             break;
             case  High_PL`:
                if (Signs-MVD && residual_Coeff)
                     DH-Key_Exchange();
                     MVD.Enc_AES();
                    Texture.Enc_AES();
                 End if
             break;
End Switch

//Diffie-Hellman Key Exchange Method
int DH-Key_Exchange()
Input: Public_key1, Publick_key2, Private_Key1, Private_Key2
    long int P=Public_Key1;
    long int Q=Public_Key2;
    long int a=Private_Key1;
    long Int b= Private_Key2;
    X=Q^a modP;
    Y=Q^b modP;
    temp = X;
    X= Y;
    Y=temp;
    Ka=Y^a modP;
    Kb=X^b modP;
    c=ka=kb;
 Return (c);
Output: Secret-key (c)

// P-XOR buffer Encryption/Decryption
int Enc_P-XOR( ) //Encryption
Input: Video, Initialization Vector (iv), Secret-key, key-size
=128;
    unsigned char *input= Sign_MVD, Signs_Texture;
    int offset1 >= 8, offset2 >= 8;
    while (1)
      offset1 = Random no. ranging from 1 to 8;
      offset2 = Random no. ranging from 1 to 8;
       uiSign = uiSign >> offset1;
       uiSign = uiSign ^ Secret_Key;
       uiSign = uiSign >> offset2;
    End while
Output: Video bit-stream
```

```
//AES-CFB buffer Encryption/Decryption (mbedTLSLibrary)

int Enc_AES( )
Input: inputvideo, Initialization Vector (iv), Secret-key, key-size
=128;

#if defined(CIPHER_MODE_CFB)

int aes_SecKey(aes_Process *prc, const unsigned char* Secret-
key, unsigned int key-size)
  if (key-size =128)
  prc->rounds  = 10;
  aes_encrypt_cfb();
  break;
int aes_encrypt_cfb(aes_Process *prc, int mode-type, size_t l-size,
size_t *iv_off, unsigned char iv, const unsigned char *inputvideo,
unsigned char *outvid )
   int i;
   size_t s = *iv_off;
   if(mode-type == ENCRYPT_AES)
      while (l-size -- )
         if(s == 0 )
           aes_encrypt_cfb (prc, ENCRYPT_AES, iv, iv );
           iv[l] = *output++ = (unsigned char)( iv[s] ^
           *inputvideo++ );
            s = ( s + 1 ) & 0x0F;
          End if
       End while
    End Else
 *iv_off = s;
 End if
Else
while(l-size -- )
         if( s == 0 )
           aes_encrypt_cfb (prc, ENCRYPT_AES, iv, iv);
           c = *input++;
           *outvid++ = (unsigned char) ( i ^ iv[s] );
            iv[s] = (unsigned char) i;
            n = ( s + 1 ) & 0x0F;
          End if
       End while
 End Else
 return(0);
Output: Video bit-stream
```

Fig. 5 Implementation of the encryption module

4.2.1 Encryption algorithms

Descriptions of the in-house encryption algorithm P-XOR and also the standardized encryption algorithm AES with CFB mode are now given in this section. As a point of reference, the complete pseudo-code of the FTC encryption module is given in Fig. 5.

4.2.1.1 P-XOR cipher Basically all symmetric ciphers work on three principles, i.e., Substitution, Permutation and XOR. Because the substitution process (in ciphers such as AES) is computationally demanding, it is not included in P-XOR. However, XOR is actually a weak candidate for encryption. Because of this shortcoming, permutation is also performed on a byte-wise basis. P-XOR consists of an initial permutation round, followed by a single XOR, and then by a final permutation round. This means that it is relatively simple to compute in hardware, making it appropriate for real-time streaming applications.

In P-XOR, permutation is performed by means of a right shift (bit-wise) operator ($\gg$) to cyclically shift the bit patterns of input data bits (128 bits at a time) to the right by a given offset number. Two random offset numbers, each contained in a different byte, are set in the algorithm for bit-wise permutation of each input in each of the two permutation rounds. After the first permutation, XOR uses the exclusive disjunction $\oplus$ operation between the input bits for SE and the 128-bit key. By choosing a 128-bit encryption/decryption key, with a key space greater than 2^{100}, the same as that of the AES mode selected, the risk of a brute force attack is reduced. After that a second permutation is performed. Decryption is performed by first reversing the second permutation. As XOR is a self-inverse, the next decryption step is XORing the cipher text with the cipher key, followed by the third decryption step, which is reversing the initial permutation.

P-XOR is particularly suitable for IoT devices (Li et al. 2015a), which typically are resource-challenged. Noura et al. (2018) proposed a one-round cipher (implemented on static images) for an IoT in which the substitution and permutation principle were selected for the encryption. However, as mentioned above, substitution is usually more compute demanding than the XOR operation. Because, shuffling or permutation is one of the basic encryption principles, image encryption algorithms have certainly incorporated that step (Gao and Chen 2008; Zhang and Liu 2011). However, those based on shuffling alone are vulnerable to attack (Arroyo et al. 2009; Wang and He 2011). It is for that reason that the in-house P-XOR is reserved for medium level confidentiality, in the sense that lightweight ciphers must trade-off between real-time operation and complete invulnerability.

4.2.1.2 Advanced encryption standard AES (also known as Rijndael) has been widely deployed as an encryption standard since 2000 (National Institute of Standards and Technology 2001). Until the present, AES is considered a very secure cipher and, hence, is extensively utilized for confidentiality in cyber-physical systems (Saifurrab and Mirza 2016). AES is a symmetric key block cipher, using a 128-bit key for 10 rounds, a 192-bit key for 12 rounds, or a 256-bit key for 14 rounds of operation. AES processes data in the form of states which are defined as a 4×4 matrix. In AES, every round comprises four stages/phases: (1) Byte-substitution, (2) Mix Columns, (3) Shift Rows and (4) Add Round Key.

CFB is one of the modes of operation used for AES encryption. In CFB mode, AES can be utilized as a stream cipher and operated on a bit/byte level (Furht et al. 2005; Stallings 2010). In CFB mode, values of plaintext are encrypted and transferred one at a time. CFB mode is chosen for implementation in the FTC model due to its self-synchronization nature and its aforementioned suitability for real-time video streaming sessions (Asghar et al. 2015). CFB uses an initialization vector (*iv*) for the initial block. However, there is no need to keep the *iv* secret. In CFB, the previously encrypted ciphertext block becomes the input and it is XORed with the current plaintext block resulting in the 'current' ciphertext block as output. In this mode, if *iv* is changed for some plaintext blocks then the resulting ciphertext will be different for each block. However, notice that chaining dependency occurs in CFB, as each ciphertext block depends on the current plaintext block and all the preceding plaintext blocks.

In the FTC-Encryption module, these two symmetric ciphers encrypt the data-bits as a stream cipher with an 128-bit encryption key. The encryption key is specific to each end-user video streaming session. To secure the key, herein, it is generated and distributed through the Diffie-Hellman key exchange method (Rescorla 1999) to avoid man-in-the-middle-attacks when transferring secret keys. This is the same method availed of for transfer of the P-XOR key and its two permutation values.

5 Results and discussion

In order to evaluate the performance of the FTC model, both the Classifier and Encryption modules are put into practice in this section. First the Classifier operates to select the appropriate privacy level to process the video bit-streams for multimedia transmission over CDNs. The Classifier was implemented with four FIS in total (Sect. 3.1) and details of these implementations are given in Sect. 4.1.

In the encryption module, the two ciphers, i.e., P-XOR and AES were applied in the selective encryption process

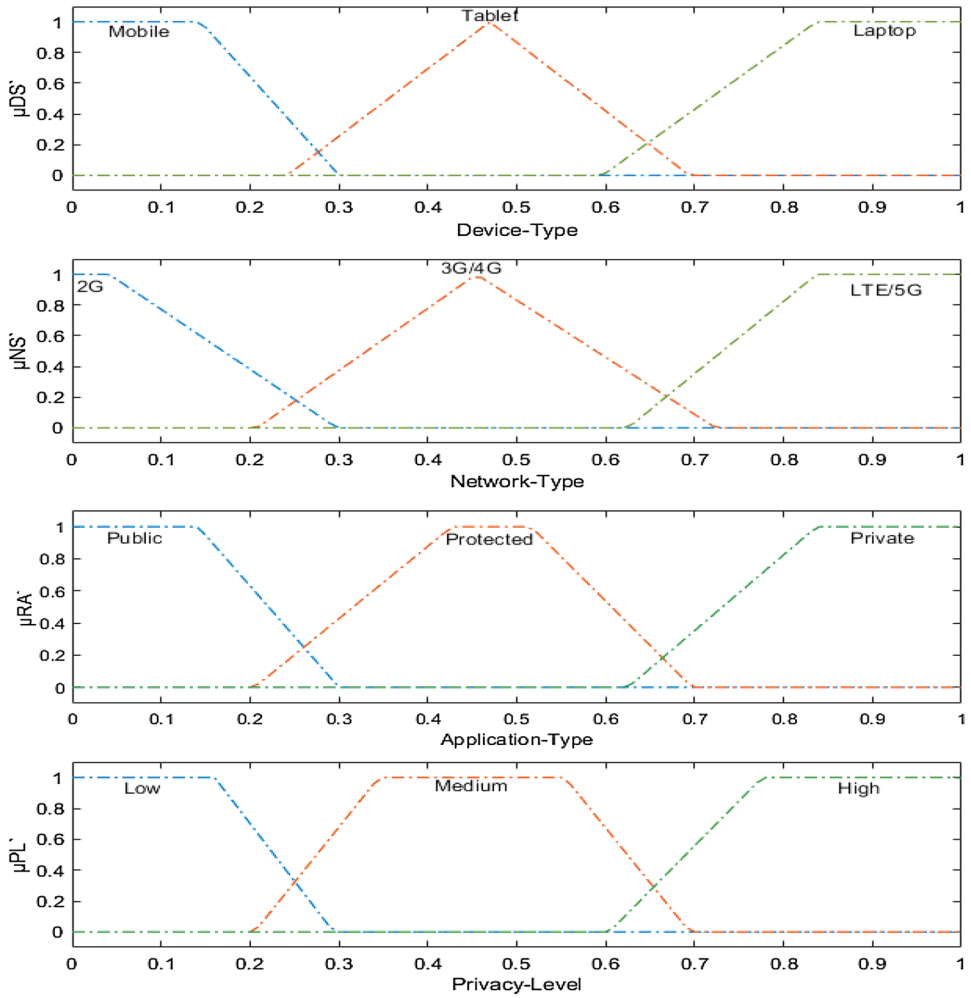


Fig. 6 Implementation of the Classifier membership functions

to test videos in order to validate the confidentiality of the output. Detailed results are given in Sect. 4.2.

5.1 Fuzzy inference system

In this paper, the Mamdani model is used for the implementation of the fuzzy inference process of the Classifier. Results were generated with the Fuzzy Logic Toolbox™ in MATLAB R2018b version (R2018b 2018). As mentioned in the methodology (Sect. 3.1), we have implemented four FISs in two layers with four fuzzy sets (DS′, NS′, RA′ and PL′). Every fuzzy set has three fuzzy subsets (refer to Sect. 3.1 for details). Layer-1 FISs produced three outputs, which are considered crisp inputs for the final FIS:PL′. The FIS:PL′ output one of the three privacy levels, which is the fundamental input to the FTC encryption module, which then encrypts each video to the desired level of confidentiality or privacy as they pass through edge servers.

The implemented membership functions for the Classifier (refer back to Fig. 3) are shown in graphical form in Fig. 6. Figure 7 is an example of applying the fuzzy if … then rules, shown in Fig. 7 in graphical form. The output surfaces for PL′, available as output from the Matlab Toolbox are illustrated in Fig. 8. By examining these surfaces, it can be clearly established that the real-time application (RA′) input is the key input in determining the appropriate privacy level, which the encryption module then applies.

5.2 Experimentation on videos with level-wise encryption

Benchmark YUV videos of different resolutions and characteristics (motion, texture) were selected for the validation tests. The tested videos were Vidyo1 (HD), Four People (HD) and Crew (4CIF), which were downloaded from Xiph.org (2018), an on-line repository. The characteristics of each video are given in Table 9.

The FTC encryption module was simulated with code in the C/C++ programming languages within video encoder JSVM (2018) (for base layer H.264/AVC, as that layer corresponds to non-scalable H.264/AVC). Likewise, for H.265/HEVC, HM version 16.18 (HM (16.18) 2018) was used. The

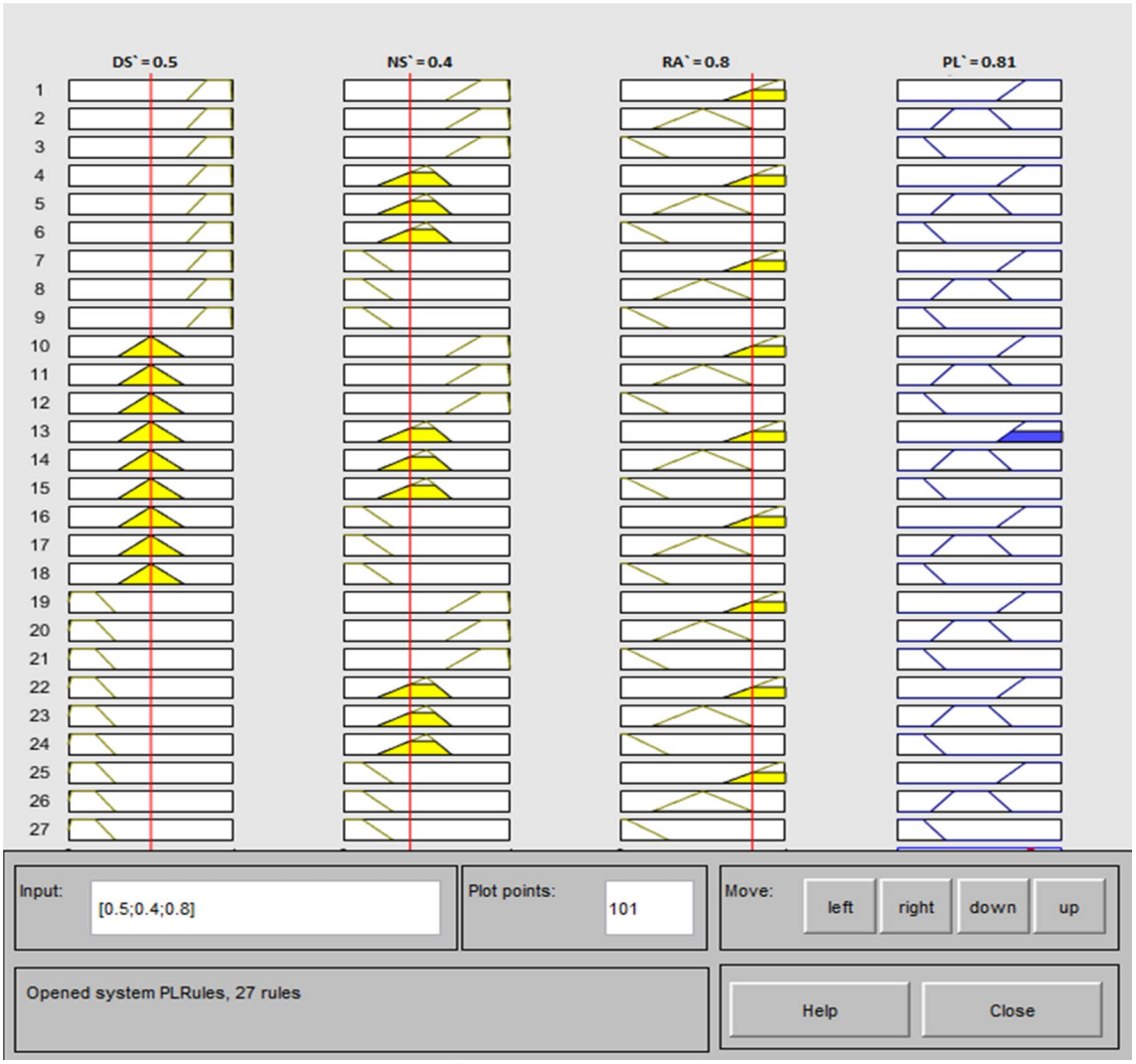
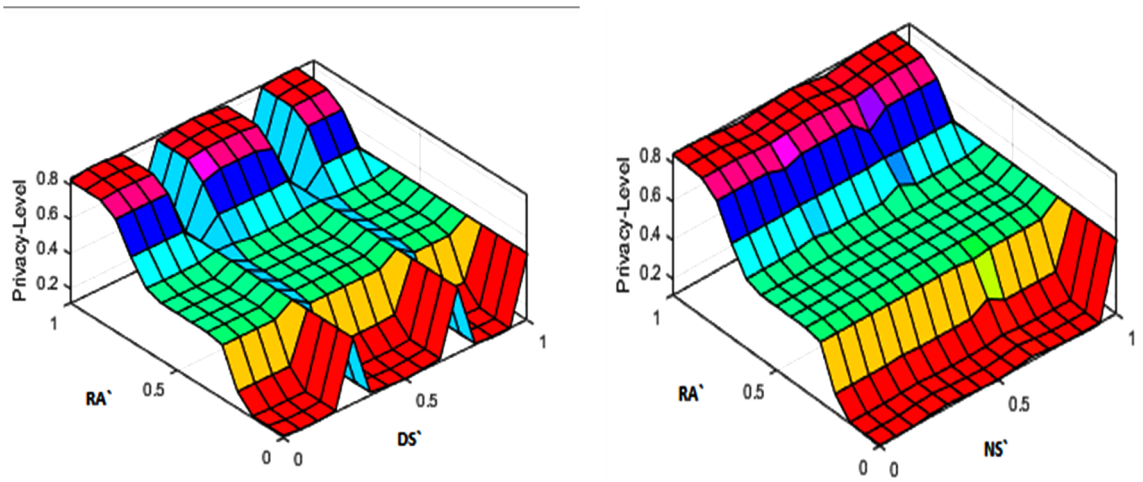


Fig. 7 Implementation of 27 rules for the final output of the Privacy Level (PL′) FIS



(a) Real-time applications with device specs.

(b) Real-time applications with network specs

Fig. 8 3D fuzzy model surface views of the Privacy Level (PL′)

Table 9 Summary of experimental videos

Sr. no.	Video	File size (kB) (YUV)	Resolution (W×H) (pixels)	Frame rate (fps)
1.	Suzie	5569	176×144	25
2.	Akiyo	11,138	176×144	25
3.	Stefan	13,365	352×288	30
4.	Mother-daughter (MD)	44,550	352×288	30
5.	Crew	178,200	704×576	30
6.	Four people	811,350	1280×720	60
7.	Vidyo1	895,950	1280×720	60
8.	Life	2,505,493	1920×1080	120
9.	Jockey	1,822,500	1920×1080	120

experiments were performed on YUV videos with chroma sampling at 4:2:0 and the frame structure as IBBP… Group of Pictures (GoP) with an intra-refresh period of length 16. The Quantization Parameter (QP) was set to 32, in a range from 0 to 51. A total of 100 frames of each video were encoded for the experiments. All experiments were performed on a 64-bit operating system with a 2.30 GHz Core i5-6200U processor and 8 GB RAM. The experiments were performed over nine videos having varying characteristics of color, motion activity pixel resolution, as well as varying camera shots, such as the use of zooms. In this way, it is possible to judge any video content dependency in the SE methods. A summary of the videos is given in Table 9 with details of file size, video frame pixel resolution, and frame display rates.

As previously mentioned, encryption was applied to the arithmetical signs of MVDs and to the arithmetical signs of residual TC with P-XOR and AES in the JSVM and HM video encoders at the CABAC entropy coding stage. AES-CFB was implemented through the mbedTLSLibrary (Anson 2018).

Figures 9 and 10 show the visual results from applying appropriate privacy level ciphers for H.264/AVC and H.265/HEVC encoded video. In Figs. 9 and 10, parts (a, b, c) depict the application of the low privacy level. That is to say the videos were encoded without applying encryption. In Figs. 9 and 10, parts (a1, b1, c1) depict the application of a medium privacy level, i.e., videos were selectively encrypted by using the P-XOR cipher, and parts (a2, b2, c2) depict the application of a high privacy level, i.e., videos were selectively encrypted by means of the AES-CFB cipher. The results given in Figs. 9 and 10 show better visual confidentiality through the distortion achieved for medium and high PL′ with the P-XOR and AES-CFB ciphers.

In Table 10, the Peak Signal-to-Noise Ratio (PSNR), valid as a metric when making comparisons between different representations of the same video (Huynh-Thu and Ghanbari 2012), is applied to each of the test videos for each privacy level, with both H.264/AVC and HEVC encoding.

Fig. 9 Visual results of PL selection applied to **a** Akiyo **b** Crew **c** MD **d** Four People **e** Vidyo1 **f** Jockey and **g** Life when encoded with H.264/AVC. **a–c** Low-level privacy (no encryption). **a1–g1** Medium-level privacy (encryption with P-XOR) and **a2–g2** high-level privacy (encryption with AES-CFB)

The PSNR is used to measure the maximum possible absolute difference between the input YUV bit-stream and the encrypted bit-stream in decibels (dB) and calculated by (37). The PSNR value is directly proportional to the relative video quality; a higher PSNR means better quality.

$$\mathrm{PSNR} = 10.\log_{10}\frac{(2^{y}-1)^{2}}{\mathrm{MSE}}, \tag{37}$$

where MSE is the Mean Square Error between the input video and the video after distortion has been introduced, with y being the bits per pixel. From Table 10, it is apparent that for either codec, the luminance image (Y) PSNR value is always considerably lower at the medium level privacy level and lower still at the high level privacy level. If the luminance frames alone were viewed alone, at the recorded dBs, the video clips would certainly be unwatchable. The chrominance images (U and V) do not display such large dips in their distortion. However, in general they follow the same trend as the luminance. Though at the higher dBs of the U and V sequences, it is possible that viewed in isolation some indication of the content would be possible, it is unlikely that there could be a satisfactory viewing experience if one of the U or V sequences were looked at in isolation.

As an alternative video quality metric, the Structural Similarity (SSIM) index (Chen and Bovik 2011), though similar to PSNR in being an objective measure, seeks to reflect the Human Visual System in the way that subjective tests do. Unlike subjective tests, SSIM results are reproducible and avoid the cost of assembling a panel of suitable viewers. SSIM scored very highly in the Video Quality Experts Group (VCEG) tests (Winkler 2005), which explains why it


(a) Low PL` (Frame # 57)


(a1) Medium PL` (Frame # 57)


(a2) High PL` (Frame # 57)

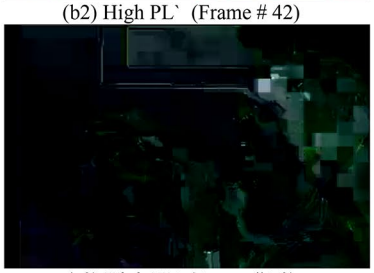

(b) Low PL` (Frame # 42)


(b1) Medium PL` (Frame # 42)

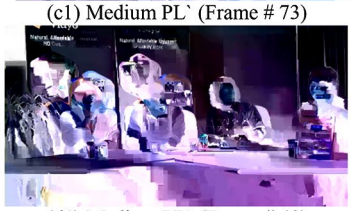

(b2) High PL` (Frame # 42)

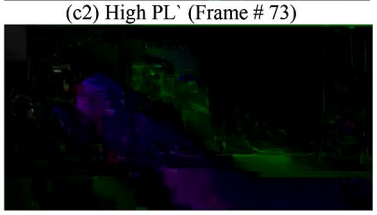

(c) Low PL` (Frame # 73)


(c1) Medium PL` (Frame # 73)


(c2) High PL` (Frame # 73)

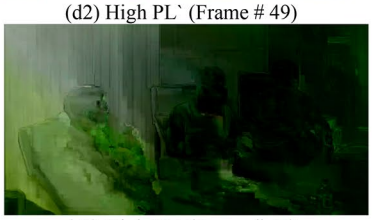

(d) Low PL` (Frame # 49)


(d1) Medium PL` (Frame # 49)

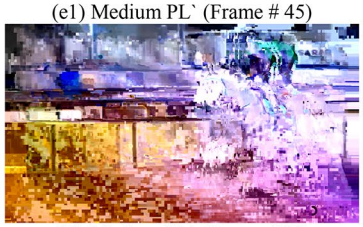

(d2) High PL` (Frame # 49)

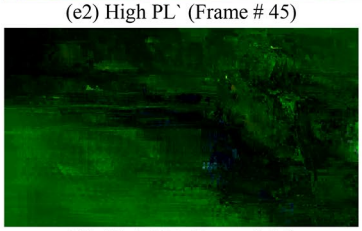

(e) Low PL` (Frame # 45)

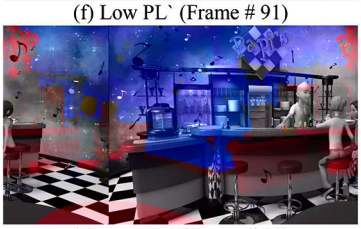

(e1) Medium PL` (Frame # 45)

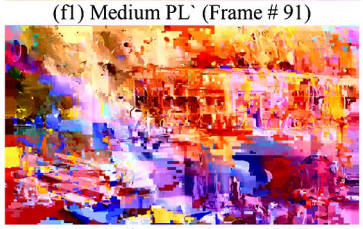

(e2) High PL` (Frame # 45)

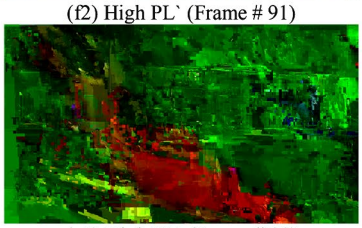

(f) Low PL` (Frame # 91)


(f1) Medium PL` (Frame # 91)


(f2) High PL` (Frame # 91)


(g) Low PL` (Frame # 88)


(g1) Medium PL` (Frame # 88)

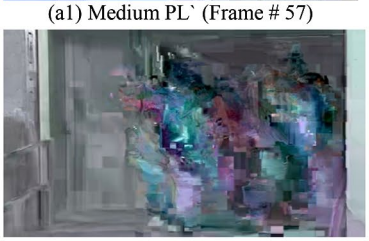

(g2) High PL` (Frame # 88)

(a) Low PL` (Frame # 57)

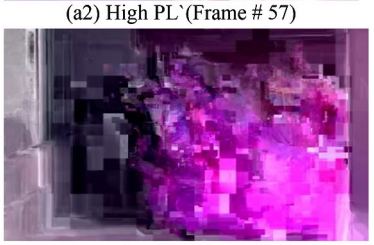
(a1) Medium PL` (Frame # 57)

(a2) High PL`(Frame # 57)

(b) Low PL` (Frame # 42)



(b1) Medium PL` (Frame # 42)



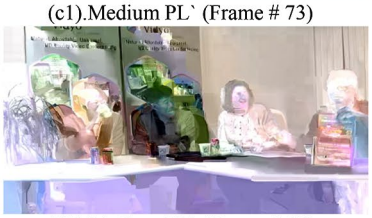
(b2) High PL`(Frame # 42)

(c) Low PL` (Frame # 73)

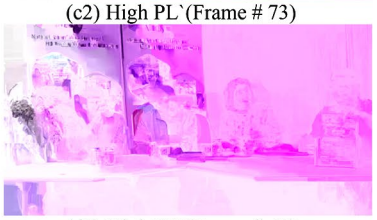
(c1).Medium PL` (Frame # 73)

(c2) High PL`(Frame # 73)

(d) Low PL` (Frame # 49)



(d1).Medium PL` (Frame # 49)



(d2) High PL`(Frame # 49)

(e) Low PL`(Frame # 45)

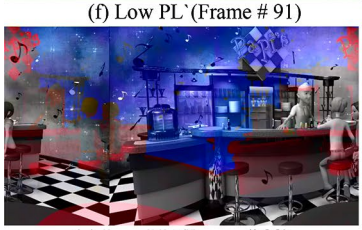
(e1) Medium PL` (Frame # 45)

(e2) High PL` (Frame # 45)

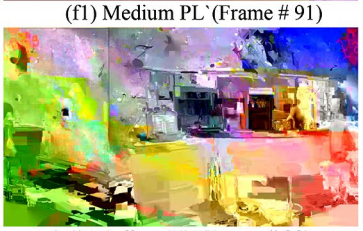
(f) Low PL`(Frame # 91)

(f1) Medium PL`(Frame # 91)

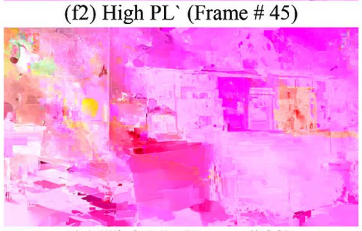
(f2) High PL` (Frame # 45)



(g) Low PL`(Frame # 88)



(g1) Medium PL`(Frame # 88)



(g) High PL`(Frame # 88)

◂**Fig. 10** Visual results of PL selection applied to **a** Akiyo **b** Crew **c** MD **d** Four People **e** Vidyo1 **f** Jockey and **g** Life when encoded with HEVC. **a**–**c** Low-level privacy (no encryption). **a1**–**g1** Medium-level privacy (encryption with P-XOR) and **a2**–**g2** high-level privacy (encryption with AES-CFB)

frequently appears, along with PSNR, in comparative quality studies. The quality range of SSIM is from 0 to 1, with 1 representing the same video sequence when compared to the pre-compressed version.

However, in Figs. 11 and 12 SSIM index plots also show that the test videos are drastically changed when SE with either P-XOR or AES-CFB for the medium and high privacy levels, whichever of the two standard encoders was first applied. The SSIM plots confirm that both types of encryption algorithms destroy the structure of the videos which makes attempts by an attacker to reconstruct the videos difficult without decryption of the parameters selectively encrypted. From Figs. 11 and 12, it is also apparent that for the same video sequence the SSIM quality tends to be lower for H.264/AVC than for HEVC, even after encryption. This might have been expected given, as previously remarked upon, the reported tests such as in Sullivan et al. 2012 that show greater video quality for the same data rate after decoding. The relative privacy levels are confirmed by the quality levels, as P-XOR encrypted videos have better relative video quality (compared to the YUV version) than AES-CFB encrypted videos. Particularly, for the HEVC results, the active Stefan video of a tennis player results in more privacy after encryption by the SSIM measure. However, the effect is not always consistent between the two codecs. For example, Akiyo, which is largely static results in less privacy by the SSIM metric when HEVC is applied compared to when H.264/AVC is used to encode the video sequence. H.264/AVC is a macroblock-based codec, which means that planar areas are always broken into the same-sized macroblocks and then encoded. However, HEVC decomposes a video frame into a quadtree, which means that

Table 10 Comparative PSNR for specific PL′ over H.264/AVC and HEVC encoding

Sr#	Video	H.264/AVC			HEVC		
		Low PL′ (no encryption) {Y,U,V}	Medium PL′ (SE with P-XOR) {Y,U,V}	High PL′ (SE with AES-CFB) {Y,U,V}	Low PL′ (no encryption) {Y,U,V}	Medium PL′ (SE with P-XOR) {Y,U,V}	High PL′ (SE with AES-CFB) {Y,U,V}
1.	Suzie	{38.6, 42.4, 41.9}	{9.5,20.3,23.5}	{8.3,29.6,24.9}	{35.2,43.1,43.0}	{12.5,26.1,23.8}	{10.5,28.1,21.7}
2.	Akiyo	{39.2, 45.6, 45.2}	{8.7,16.3, 19.6}	{6.0,11.2,16.0}	{37.2,40.1,41.6}	{17.1,16.2,20.2}	{12.2,14.9,22.0}
3.	Stefan	{40.3,43.5,43.9}	{9.0,15.1,18.7}	{7.0,16.2,17.0}	{31.8,37.0,37.2}	{12.6,19.7,19.8}	{9.1,18.9,18.3}
4.	MD	{36.6, 41.9, 43.1}	{8.9,14.8,20.2}	{7.3, 21.2, 19.9}	{37.5,43.0,43.9}	{6.3,19.3,20.3}	{5.6,16.6,26.9}
5.	Crew	{34.5,40.3,40.0}	{10.5,26.7,22.1}	{9.3,25.5,19.6}	{35.1,41.3,41.0}	{17.3,22.5,19.7}	{14.0,16.4,19.8}
6.	Four People	{39.5,44.9,44.5}	{8.3,28.0,29.0}	{6.8,20.2,26.4}	{36.1,38.8,40.0}	{9.0, 23.8, 27.9}	{7.1,18.0,18.3}
7.	Vidyo1	{39.5,44.9,44.5}	{10.1,26.7,24.6}	{7.9,21.8,27.0}	{40.8,46.4,45.3}	{12.0,25.8,27.9}	{11.9,17.4,19.7}
8.	Jockey	{38.1,41.0,41.5}	{9.5,26.0,22.5}	{8.2,17.7, 18.2}	{39.7,42.5,42.8}	{13.9,23.4,21.4}	{11.1,16.2,14.2}
9.	Life	{34.7,38.9,39.4}	{9.0,12.2,13.7}	{8.9,21.7,20.8]	{35.6,41.2,41.3}	{8.4,14.8,19.1}	{7.7,19.4,13.6}

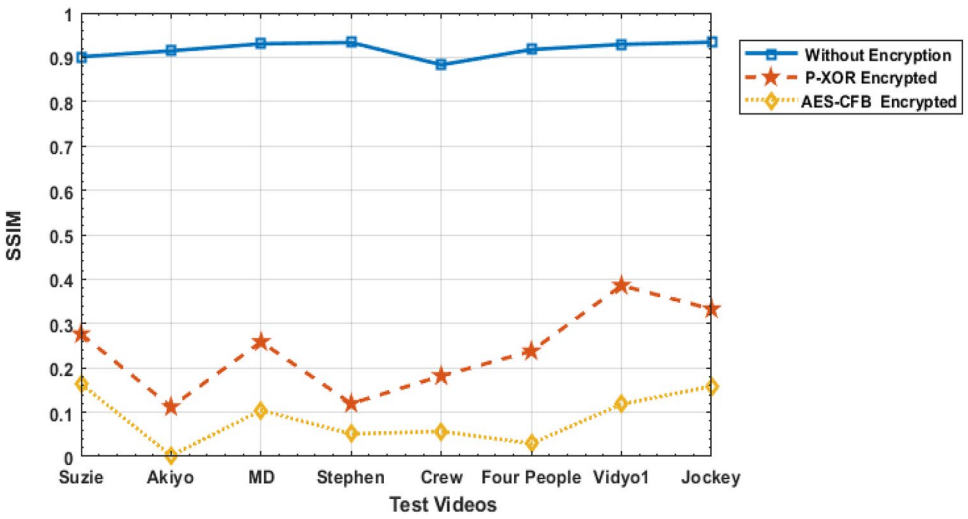


Fig. 11 Comparative SSIMs for different PL′ after H.264/AVC video encoding

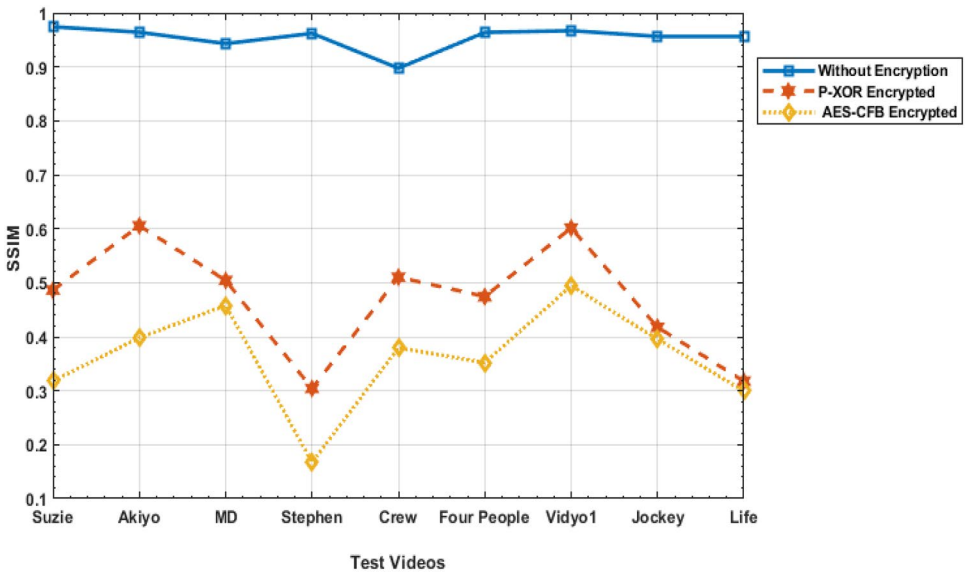


Fig. 12 Comparative SSIMs for different PL′ after HEVC video encoding

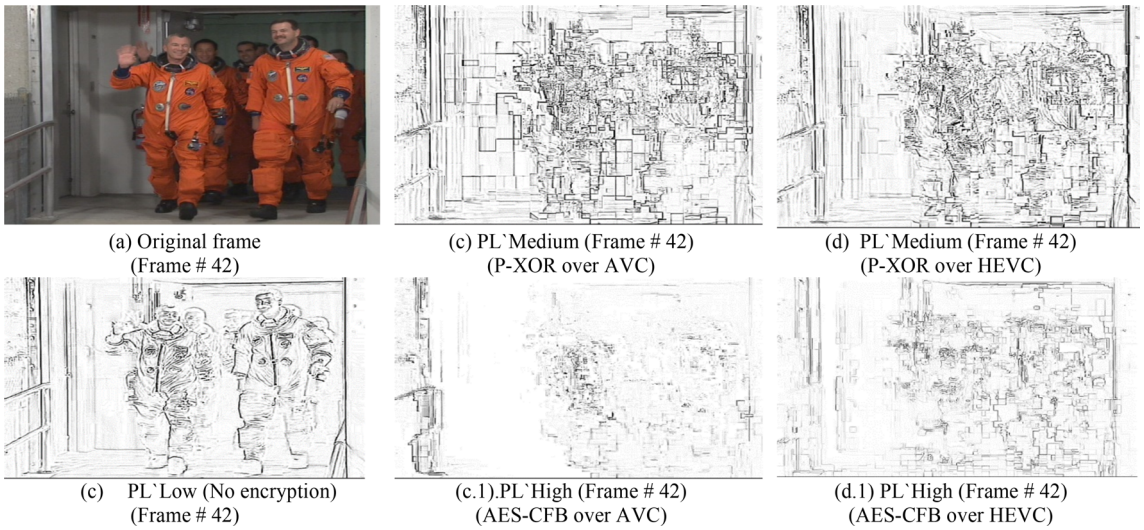


Fig. 13 The comparative visual impact of low, medium and high level privacy for the Crew video after applying the Laplacian edge detector

planar areas can be encoded as one if the texture characteristics are largely similar. This may explain the relative effects between the two codecs in terms of subsequent encryption according to the content. However, these second-order variations are subordinate to the ordering of the privacy levels and the general behavior of the two codecs.

5.2.1 Structural distortion analysis

SE applied in a correct way disturbs the structure of a video clip. Therefore, apart from visual distortion testing, structural distortion analysis of applied SE with either P-XOR or AES-CFB over the two encoders was examined. This was tackled through a 3×3 Laplacian edge detection (Shivakumara et al. 2011), as changes to edges indicate changes to the structure of each video frame. The ratio (R) (Shahid and Puech 2014) for detection of edges can be calculated through the following equation:

$$\mathrm{R} = \frac{\sum_{\mathrm{x,y=1}}^{\mathrm{n}} |E(x,y) - E'(x,y)|}{\sum_{\mathrm{x,y=1}}^{\mathrm{n}} |E(x,y) + E'(x,y)|}, \tag{38}$$

where $E(x, \mathrm{y}), E'(z, \mathrm{y})$ represents the pixel values of the edges detected in a binary version of the video frames for original and encrypted images, respectively. The detected edges of all three privacy levels are shown in Figs. 13 and 14. The

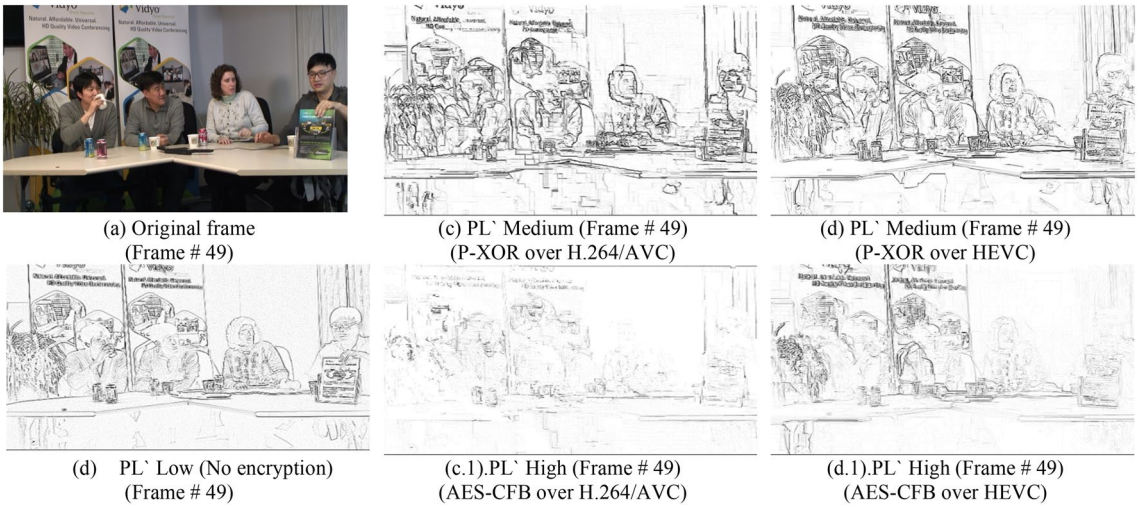


Fig. 14 The comparative visual impact of low, medium and high level privacy for the Four People video after applying the Laplacian edge detector

comparative results show that the medium and high privacy videos are distorted in a way that an interceptor cannot easily acquire useful information from the encrypted video clips.

5.2.2 Entropy analysis

Pixels within video frames are highly correlated with neighboring pixels in the vertical, horizontal, and diagonal directions. If all pixels in encrypted video frames have the same gray level or the same intensity of color components, a frame image will present the minimal entropy (randomness) value, and the information cannot be retrieved through that video frame. After the application of SE, the pixel values were truncated to a maximum 255 and a minimum 0 in frames. This introduces the spread of dark or quite bright colours throughout the video frames (as shown in Figs. 9 and 10). The entropy analysis has been performed to validate the SE over encrypted video at each privacy level. Entropy is calculated as:

$$H(g) = -\sum_{i=0}^{2N-1} p(g)\log_2(g) \tag{39}$$

where g is the gray level value and its probability is $p(g)$. Figure 15 graphically represents the comparative pixel randomness with P-XOR or AES-CFB encryption with either H.264/AVC or HEVC encoding of Crew, Four People and Vidyo1 test videos. Though there are differences between the patterns for both codecs, a detailed analysis of these are beyond the scope of the current paper. Whichever codec (H.264/AVC or HEVC) is applied, the entropy is reduced after the application of P-XOR. After applying AES, in general, there are further reductions in entropy.

5.2.3 Computational analysis for real-time streaming

The total execution times taken by the proposed FTC encryption module are illustrated in Fig. 16. Notice that the vertical axes' scales are different between the two plots of Fig. 16 as plot (b) is larger by a factor of 10^4, as indicated on the chart. The comparative results show that H.264/AVC is much faster in terms of computation compared to HEVC, as the execution time is directly proportional to the computational complexity. Importantly, the encryption times depend on the privacy level, with a higher privacy level resulting in a longer execution time. This demonstrates the trade-off between increased privacy and increased computation time. For real-time streaming, increased computation time at the encoder, when added to that of network latency may be harmful. Increased computation time also results in increased energy consumption, which is important for battery-powered mobile devices. In that respect, the increased computation time of Jockey and Life is a reflection of the greater pixel resolution and YUV file size of those video sequences (refer to Table 9). However, from the relative timings between Jockey and Life, there is also a content-dependency to the encoding times.

The results of Fig. 17 depict the higher compression achieved with HEVC compared to that of H.264/AVC (for 100 frames encoded with either H.264/AVC or HEVC).

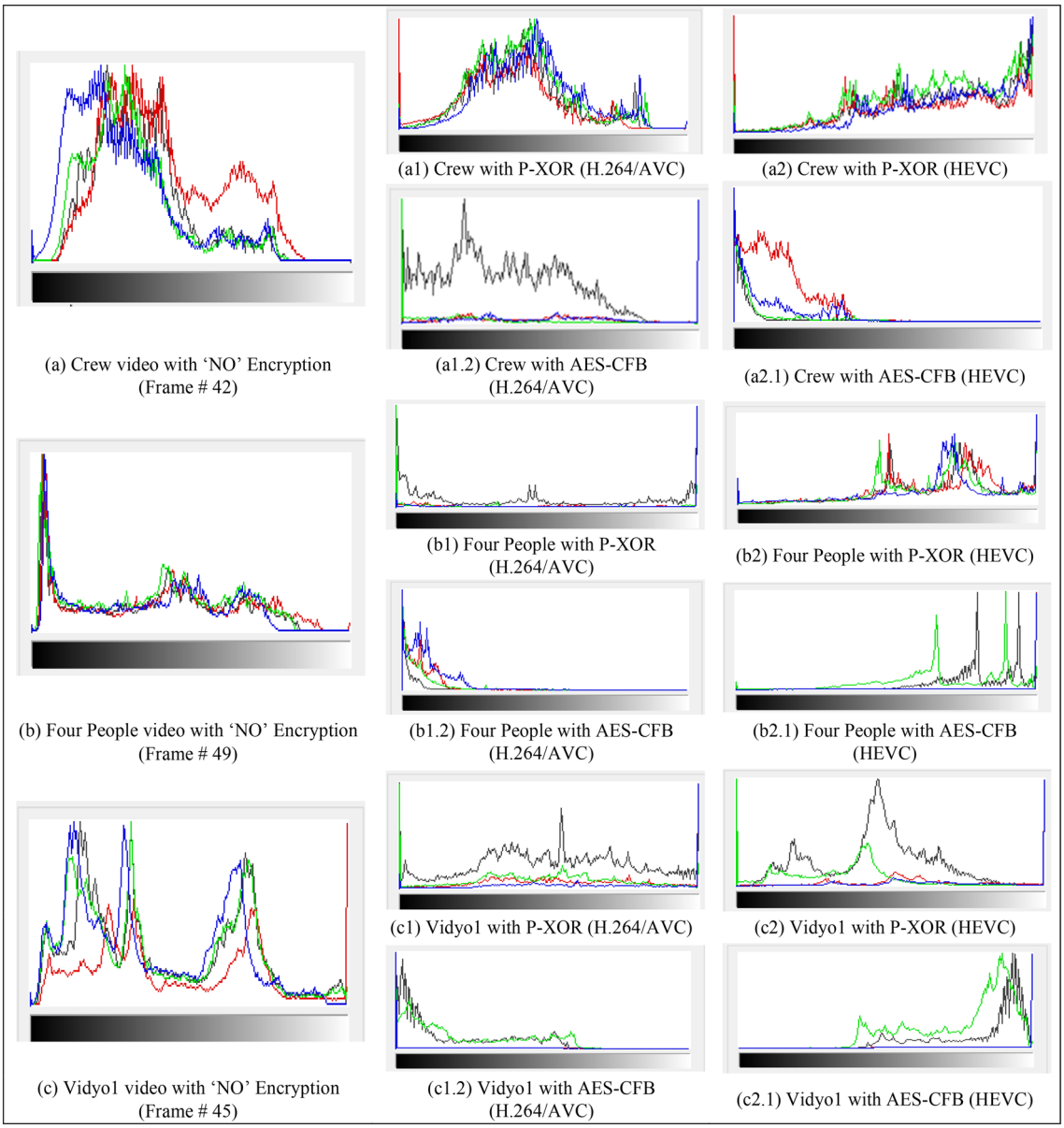
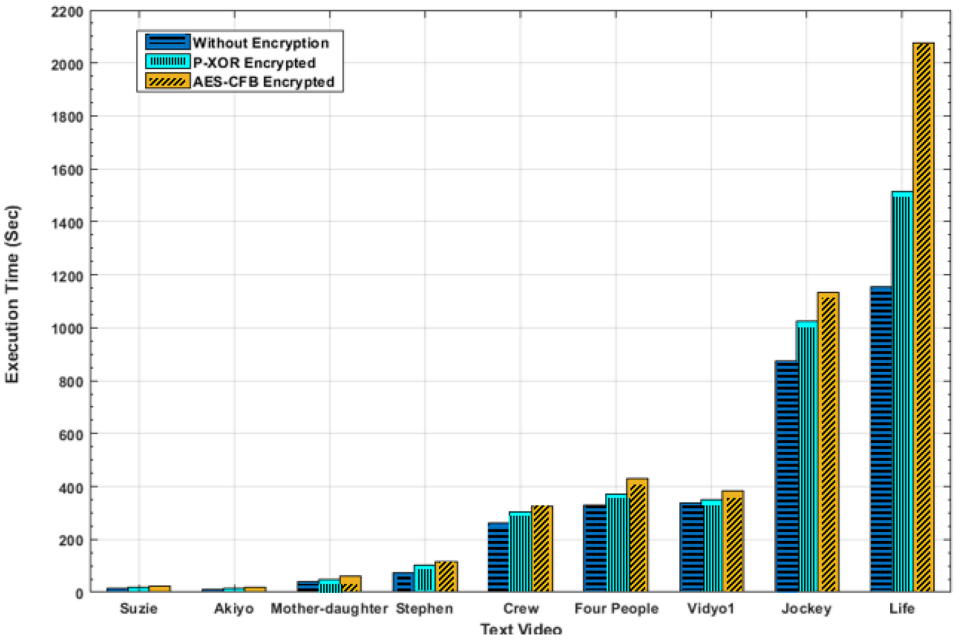


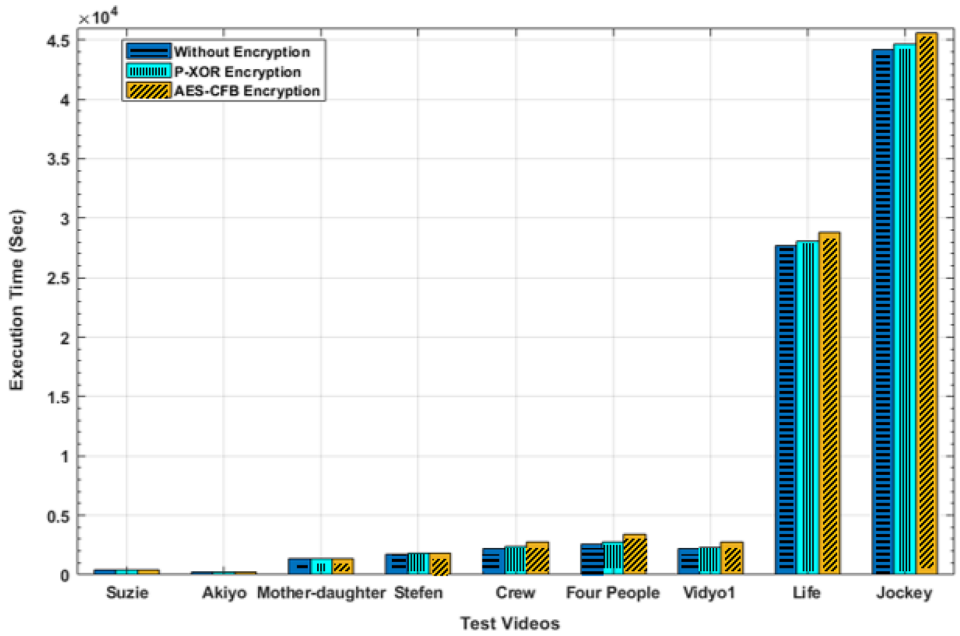
Fig. 15 Entropy analysis of privacy level wise encryption applied to **a** Crew, **b** Four People, and **c** Vidyo1, with the horizontal axis being ranged across the pixel values (0–255) and the vertical axis recording the entropy value

Thus, for high resolution video compression, HEVC is more suitable than H.264/AVC. This is particularly apparent for the higher resolutions of the Jockey and Life (refer to Table 9). The content dependency aspect of the achievable compression is also more apparent when comparing higher resolution Jockey and Life. Additionally, for comparative crypto-encoding time analysis, the Absolute Encryption Time (AET) for each implemented PL′ (high = AES-CFB encryption, medium = P-XOR encryption, low = No encryption) over H.264/AVC and HEVC video encoders is given in Table 11. The AET is calculated as:

$$AET = Execution\ time - Encoding\ time \tag{40}$$



(a) Execution time (s) with H.264/AVC encoder



(b) Execution time (s) with HEVC encoder

Fig. 16 Comparative execution times when encrypting after encoding with **a** H.264/AVC **b** HEVC codecs

The results verify that P-XOR provides sufficient level of security with lower complexity in comparison to AES-CFB, as the AETs are always lower for P-XOR compared to AES-CFB.

6 Conclusions

Interception is the most common attack on transmitted videos, particularly those sourced via CDNs, and encryption is the solution of service providers. Full encryption applied

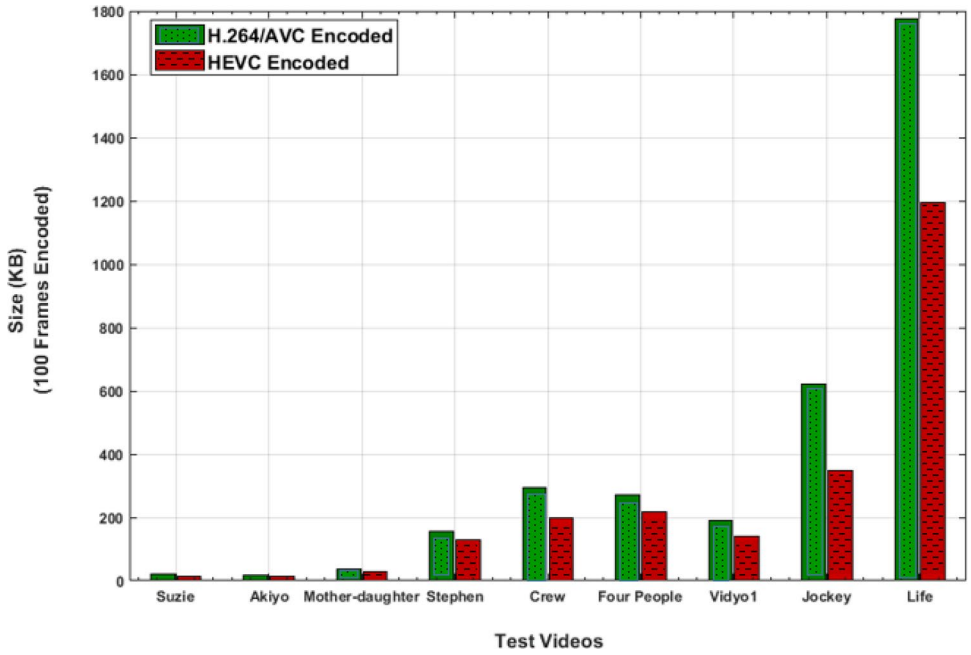


Fig. 17 The comparative compression ratios of H.264/AVC and HEVC

to any type of streamed videos results in a greater computational cost and latency. Therefore, the important thing to focus on is that, while all types of streamed video (except publically available video) require privacy, what level of confidentiality or privacy can be maintained with minimal computation in a real-time environment. This paper proposed a Threat Classification Model using fuzzy logic for real-time detection of an appropriate privacy level for transmitted videos. In this model, firstly, the Classifier module works with four Fuzzy Inference Systems and outputs an appropriate privacy level for a given end-user. Fuzzy logic is applied by considering the device type, network characteristics, and real-time application as primary fuzzy sets. Secondly, after detection of the privacy level through the Classifier, the Encryption module encrypts either H.264/AVC or HEVC encoded videos with either the in-house P-XOR or the industry-strength AES cipher. By keeping in mind the need to keep personal videos confidential, the AES cipher operating in CFB mode is applied when where the privacy level is selected to be high. The P-XOR cipher, which is a variant of XOR with two additional permutation rounds, is implemented to provide medium-level privacy or confidentiality to streamed videos. If the privacy level is set to Low, then the encryption module is by-passed. Consequentially, no encryption of videos will be performed at edge servers.

The implementations of P-XOR and AES-CFB for selective encryption for both encoders was tested in the experiments. The security of the encryption key was also considered and Diffie-Hellman key exchange was implemented as part of the encryption module. An encryption key is generated per user's video transmission session. If the same user initiates the next session, then a new key will be generated again for that specific user. Results show the accuracy of the fuzzy Classifier and the considerable visual quality degradation of the tested videos. Comparative distortion analysis and entropy testing upon encrypted test videos validated the relative confidentiality achieved according to the privacy level. The crypto-encoding time and also the compression ratio for H.264/AVC and HEVC were also calculated in the experiments. Until the present time, AES is well-known as the strongest symmetric, block cipher due to its complex set of rounds, which consequently increase the Absolute Encryption Time (AET). However, visual results and the comparative AET verify the suitability of adopting P-XOR for real-time videos in comparison to AES for a sufficient level of confidentiality, as part of a privacy package. It can be concluded that the proposed FTC model will be effective in implementing video streaming within a Peer-to-Peer, enterprise, or public CDN with minimal modifications.

Table 11 Comparative absolute encryption time (s) of the proposed FTC system for PL′ protection

Sr. no.	Video clip	File Size (KB) (YUV)	File size (kB) (100 frames)			Absolute encryption time (s)					
			YUV	H.264	HEVC	H.264			HEVC		
						No Enc.	Enc. with PXOR	Enc. with AES-CFB	No Enc.	Enc. with PXOR	Enc. with AES-CFB
1.	Suzie	5569	3713	20	15	12.71	5.01	8.50	375.33	10.338	16.16
2.	Akiyo	11,138	3713	16	13	11.065	4.21	7.33	266.16	3.81	11.47
3.	MD	44,550	14,850	37	30	39.155	10.25	21.58	1315.836	51.27	85.71
4.	Stephan	13,365	13,365	154	130	71.301	30.01	43.18	1750.53	89.74	109.03
5.	Crew	178,200	59,400	294	199	260.16	43.84	64.44	2206.62	151.09	531.42
6.	Four People	811,350	131,000	271	218	327.43	45.41	103.91	2589.07	190.02	819.74
7.	Vidyo1	895,950	135,000	190	141	337.42	11.16	47.53	2187.51	147.59	560.67
8.	Jockey	1,822,500	303,750	620	350	875.111	147.69	258.57	27,720.30	335.87	1057.89
9.	Life	2,505,943	303,750	1940	1194	1156.32	359.88	921.690	44,170.75	515.05	1415.54

Acknowledgements This research paper was produced as part of a government-funded project [National Research Program for Universities (NRPU-2016)] with no: 6282/Punjab/NRPU/R&D/HEC/2016. We appreciate the support of the Higher Education Commission (HEC) of Pakistan for this project.

References

Anson D (2018) https://os.mbed.com/users/ansond/code/mbedTLSLibrary/docs/tip/aes_8c_source.html. Accessed 6 Jan 2019

Arroyo D, Li C, Li S, Alvarez G, Halang WA (2009) Cryptanalysis of an image encryption scheme based on a new total shuffling algorithm. Chaos Solitons Fractals 41:2613–2616. https://doi.org/10.1016/j.chaos.2008.09.051

Asghar MN, Ghanbari M (2013) An efficient security system for CABAC bin-strings of H.264/SVC. IEEE Trans Circuits Syst Video Technol 23:425–437. https://doi.org/10.1109/tcsvt.2012.2204941

Asghar MN, Ghanbari M, Fleury M, Reed MJ (2015) Sufficient encryption based on entropy coding syntax elements of H.264/SVC. Multimed Tools Appl 74:10215–10241. https://doi.org/10.1007/s11042-014-2160-6

Asghar MN, Fleury M, Makki S (2017) Interoperable conditional access with video selective encryption for portable devices. Multimed Tools Appl 76:13139–13152. https://doi.org/10.1007/s11042-016-3725-3

Ashfaq RAR, Wang X-Z, Huang JZ et al (2017) Fuzziness based semi-supervised learning approach for intrusion detection system. Inf Sci (Ny) 378:484–497. https://doi.org/10.1016/j.ins.2016.04.019

Badva O, Gupta BB, Gupta S (2016) Reviewing the security features in contemporary security policies and models for multiple platforms. In: Handbook of research on modern cryptographic solutions for computer and cyber security. IGI Global, Pennsylvania, pp 479–504. https://doi.org/10.4018/978-1-5225-0105-3

Bentaleb A, Taani B, Begen AC, Timmerer C, Zimmermann R (2018) A survey on bitrate adaptation schemes for streaming media over HTTP. IEEE Trans Commun Surv Tutor. https://doi.org/10.1109/comst.2018.2862938

Biswas B, Patra S (2018) Forecasting problems in cybersecurity: applying econometric techniques to measure IT risk. Computer and cyber security: principles, algorithm, applications, and perspectives. CRC Press, Boca Raton, pp 45–94

Chen MJ, Bovik AC (2011) Fast structural similarity index algorithm. J Real Time Image Process 6:281–287. https://doi.org/10.1007/s11554-010-0170-9

Chen J-B, Liao S-J (2010) A fuzzy-based decision approach for supporting multimedia content request routing in CDN. In: International symposium on parallel and distributed processing with applications. IEEE, pp 46–51

Chen T-C, Huang Y-W, Tsai C-Y et al (2006) Architecture design of context-based adaptive variable-length coding for H.264/AVC. IEEE Trans Circuits Syst II Express Br 53:832–836. https://doi.org/10.1109/tcsii.2006.880014

Cisco (2018) Cisco Global Cloud Index: forecast and methodology, 2016–2021 White Paper—Cisco. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.html. Accessed 5 Jan 2019

Collotta M, Pau G (2015) Bluetooth for Internet of Things: a fuzzy approach to improve power management in smart homes. Comput Electr Eng 44:137–152. https://doi.org/10.1016/j.compeleceng.2015.01.005

Cui S, Asghar MR, Russello G (2018) Multi-CDN: towards privacy in Content Delivery Networks. IEEE Trans Dependable Secur Comput 5971:1–16. https://doi.org/10.1109/tdsc.2018.2833110

Cuka M, Elmazi D, Bylykbashi K, Ikeda M, Barolli L (2019) Implementation and performance evaluation of two fuzzy-based systems for selection of IoT devices in opportunistic networks. J Ambient Intell Hum Comput 10:519–529. https://doi.org/10.1007/s12652-017-0676-0

Fan Q, Yin H, Min G et al (2018) Video delivery networks: challenges, solutions and future directions. Comput Electr Eng 66:332–341. https://doi.org/10.1016/j.compeleceng.2017.04.011

Farajallah M, Hamidouche W, Deforges O, Assad S El (2015) ROI encryption for the HEVC coded video contents. In: 2015 IEEE international conference on image processing (ICIP). IEEE, pp 3096–3100

Furht B, Socek D, Eskicioglu AM (2005) Fundamentals of multimedia encryption techniques. In: Furht B, Kirovski D (eds) Multimedia security handbook. CRC Press, Boca Raton, pp 95–132

Gandotra E, Bansal D, Sofat S (2017) Malware threat assessment using fuzzy logic paradigm. Cybern Syst 48:29–48. https://doi.org/10.1080/01969722.2016.1262704

Gao T, Chen Z (2008) Image encryption based on a new total shuffling algorithm. Chaos Solitons Fractals 38:213–220. https://doi.org/10.1016/j.chaos.2006.11.009

Garg U, Sikka G, Aawsthi LK (2018) A systematic review of attack graph generation and analysis techniques. Computer and cyber security: principles, algorithm, applications, and perspectives. CRC Press, Boca Raton, pp 115–146

Ghanbari M (2003) Standard codecs: image compression to advanced video coding. IET Publications, Murinjapalam

Gupta BB, Agrawal DP, Wang H (eds) (2018) Computer and cyber security: principles, algorithm, applications, and perspectives. CRC Press, Boca Raton

Hernandez-Ramos JL, Moreno MV, Bernabe JB, Carrillo DG, Skarmeta AF (2015) SAFIR: secure access framework for IoT-enabled services on smart buildings. J Comput Syst Sci 81(8):1452–1463

HM(16.18) (2018) HEVC Test Model (HM): main page. https://hevc.hhi.fraunhofer.de/HM-doc/index.html. Accessed 6 Jan 2019

Hsieh M-Y, Hsu Y-C, Lin C-T (2018) Risk assessment in new software development projects at the frontend: a fuzzy logic approach. J Ambient Intell Human Comput 9:295–305. https://doi.org/10.1007/s12652-016-0372-5

Huynh-Thu Q, Ghanbari M (2012) The accuracy of PSNR in predicting video quality for different video scenes and frame rates. Telecommun Syst 49:35–48. https://doi.org/10.1007/s11235-010-9351-x

JSVM (2018) JSVM Reference Software—Fraunhofer Heinrich Hertz Institute. https://www.hhi.fraunhofer.de/en/departments/vca/research-groups/image-video-coding/research-topics/svc-extension-of-h264avc/jsvm-reference-software.html. Accessed 6 Jan 2019

Kan M (2019) Google: Phishing attacks that can beat two-factor are on the rise. PC Mag. https://uk.pcmag.com/google-titan-security-key-bundle/119999/google-phishing-attacks-that-can-beat-two-factor-are-on-the-rise. Accessed 30 Jan 2020

Kollective Technology (2018) Security should never be a concern with enterprise videolKollective Technology. https://kollective.com/resource/security-should-never-be-a-concern-with-enterprise-video/. Accessed 5 Jan 2019

Li S, Da XuL, Zhao S (2015a) The Internet of Things: a survey. Inf Syst Front 17:243–259. https://doi.org/10.1007/s10796-014-9492-7

Li Z, Wu Q, Salamatian K, Xie G (2015b) Video delivery performance of a large-scale VoD system and the implications on content delivery. IEEE Trans Multimed 17:880–892. https://doi.org/10.1109/tmm.2015.2417771

Long M, Peng F, Li H (2018) Separable reversible data hiding and encryption for HEVC video. J Real Time Image Process 14:171–182. https://doi.org/10.1007/s11554-017-0727-y

Lookabaugh T, Sicker DC (2004) Selective encryption for consumer applications. IEEE Commun Mag 42:124–129. https://doi.org/10.1109/mcom.2004.1299355

Lu Y, Li L, Peng H, Yang Y (2016) Robust anonymous two-factor authenticated key exchange scheme for mobile client-server environment. Secur Commun Netw 9(11):1331–1339

Mamdani EH, Assilian S (1975) An experiment in linguistic synthesis with a fuzzy logic controller. Int J Man Mach Stud 7:1–13. https://doi.org/10.1016/s0020-7373(75)80002-2

Marpe D, Schwarz H, Wiegand T (2003) Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard. IEEE Trans Circuits Syst Video Technol 13:620–636. https://doi.org/10.1109/tcsvt.2003.815173

Martino FDI, Sessa S (2018) Multi-level fuzzy transforms image compression. J Ambient Intell Human Comput 10:2745–2756. https://doi.org/10.1007/s12652-018-0971-4

Massoudi A, Lefebvre F, De Vleeschouwer C et al (2008) Overview on selective encryption of image and video: challenges and perspectives. Eurasip J Inf Secur. https://doi.org/10.1155/2008/179290

Memos VA, Psannis KE, Ishibashi Y, Kim B-G, Gupta BB (2018) An efficient algorithm for media-based surveillance system (EAM-SuS) in IoT smart city framework. Future Gen Comput Syst 83:619–628. https://doi.org/10.1016/j.future.2017.04.039

Mudia HM, Chavan PV (2015) Fuzzy logic based image encryption for confidential data transfer using (2, 2) secret sharing scheme-review. In: 2015 IEEE international conference on advances in computer engineering and applications, pp 404–408

National Institute of Standards and Technology (2001) FIPS 197: Advanced Encryption Standard (AES). Federal Information Processing Standards Publication

Noura H, Chehab A, Sleem L, Noura M, Couturier R, Mansour MM (2018) One round cipher algorithm for multimedia IoT devices. Multimed Tools Appl 77:18383–18413

Ohm J, Sullivan GJ (2013) High efficiency video coding: the next frontier in video compression [Standards in a Nutshell]. IEEE Signal Process Mag 30:152–158. https://doi.org/10.1109/msp.2012.2219672

Pedrycz W (1994) Why triangular membership functions? Fuzzy Sets Syst 64:21–30. https://doi.org/10.1016/0165-0114(94)90003-5

Peng Fei, Zhu Xiao-wen, Long Min (2013) An ROI privacy protection scheme for H.264 video based on FMO and Chaos. IEEE Trans Inf Forensics Secur 8:1688–1699. https://doi.org/10.1109/tifs.2013.2259819

Plageras AP, Psannis KE, Stergiou C, Wang H, Gupta BB (2017) Efficient IoT-based sensor BIG Data collection-processing and analysis in smart buildings. Future Gen Comput Syst 82:349–357. https://doi.org/10.1016/j.future.2017.09.082

Qi M, Chen J (2017) An efficient two-party authentication key exchange protocol for mobile environment. Int J Commun Syst. https://doi.org/10.1002/dac.3341

R2018b (2018) Fuzzy Logic Toolbox—MATLAB. https://www.mathworks.com/products/fuzzy-logic/whatsnew.html. Accessed 8 Jan 2019

Rainer JJ, Cobos-Guzman S, Galán R (2018) Decision making algorithm for an autonomous guide-robot using fuzzy logic. J Ambient Intell Human Comput 9:1177–1189. https://doi.org/10.1007/s12652-017-0651-9

Ram C, Panwar S (2017) Performance comparison of high efficiency video coding (HEVC) with H.264 AVC. In: 2017 IEEE 13th international conference on signal-image technology and internet-based systems (SITIS), pp 303–310

Reddit (2018) Thinking about switching your library to HEVC/H.265? AV1 and why you should consider waiting: Plex. https://www.

reddit.com/r/PleX/comments/6y9211/thinking_about_switching_your_library_to_hevch265/. Accessed 6 Jan 2019

Reddy AG, Das AK, Odelu V, Ahmad A, Shin JS (2019) A privacy preserving three-factor authenticated key agreement protocol for client–server environment. J Ambient Intell Human Comput 10:661–680. https://doi.org/10.1007/s12652-018-0716-4

Rescorla E (1999) Diffie-Hellman key agreement method. https://www.ietf.org/rfc/rfc2631.txt. Accessed 6 Jan 2019

Ribino P, Lodato C (2019) A distributed fuzzy system for dangerous events real-time alerting. J Ambient Intell Human Comput 10:4263–4282. https://doi.org/10.1007/s12652-018-1102-y

Rouse M (2015) IoT security (Internet of Things security), IoT Agenda [Online]. http://internetofthingsagenda.techtarget.com/definition/IoT-security-Internet-of-Things-security. Accessed 28 Jan 2020

Roy S, Bose R, Sarddar D (2015) Fuzzy based dynamic load balancing scheme for efficient edge server selection in Cloud-oriented content delivery network using Voronoi diagram. In: 2015 IEEE international advance computing conference (IACC), pp 828–833

Saifurrab C, Mirza S (2016) AES algorithm using advance key implementation in MATLAB. Int Res J Eng Technol 3(9):846–850

Schwarz H, Marpe D, Wiegand T (2007) Overview of the scalable video coding extension of the H.264/AVC standard. IEEE Trans Circuits Syst Video Technol 17:1103–1120. https://doi.org/10.1109/tcsvt.2007.905532

Seufert M, Egger S, Slanina M, Zinner T, Hoßfeld T, Tran-Gia P (2015) A survey on quality of experience of HTTP adaptive streaming. IEEE Commun Surv Tutor 17:469–492. https://doi.org/10.1109/comst.2014.2360940

Shahid Z, Puech W (2014) Visual protection of HEVC video by selective encryption of CABAC Bin-strings. IEEE Trans Multimed 16:24–36. https://doi.org/10.1109/tmm.2013.2281029

Shahid Z, Chaumont M, Puech W (2011) Fast protection of H.264/AVC by selective encryption of CAVLC and CABAC for I and P frames. IEEE Trans Circuits Syst Video Technol 21:565–576. https://doi.org/10.1109/tcsvt.2011.2129090

Shifa A, Asghar MN, Fleury M (2016) Multimedia security perspectives in IoT. J Inf Secur Res 7(4):150–159

Shifa A, Asghar MN, Noor S, Gohar N, Fleury M (2019) Lightweight cipher for H.264 videos in the Internet of Multimedia Things with Encryption Space Ratio diagnostics. Sensors 19(5):1228–1234. https://doi.org/10.3390/s19051228

Shivakumara P, Phan Trung Quy, Tan Chew Lim (2011) A Laplacian approach to multi-oriented text detection in video. IEEE Trans Pattern Anal Mach Intell 33:412–419. https://doi.org/10.1109/tpami.2010.166

Singh K, Verma AK, Aggarwal P (2018) Analysis of various trust computation methods: a step toward secure FANETs. Computer and cyber security: principles, algorithm, applications, and perspectives. CRC Press, Boca Raton, pp 171–194

Stallings W (2010) Network security essentials: applications and standards, 4th edn. In: Pearson. http://www.mypearsonstore.com/bookstore/network-security-essentials-applications-and-standards-9780136108054. Accessed 6 Jan 2019

Stergiou C, Psannis KE, Kim B-G, Gupta B (2018) Secure integration of IoT and cloud computing. Future Gen Comput Syst 78:964–975. https://doi.org/10.1016/j.future.2016.11.031

Stocker V, Smaragdakis G, Lehr W, Bauer S (2017) The growing complexity of content delivery networks: challenges and implications for the Internet ecosystem. Telecommun Policy 41:1003–1016. https://doi.org/10.1016/j.telpol.2017.02.004

Sullivan GJ, Ohm J-R, Han W-J, Wiegand T (2012) Overview of the high efficiency video coding (HEVC) standard. IEEE Trans Circuits Syst Video Technol 22:1649–1668. https://doi.org/10.1109/tcsvt.2012.2221191

Sze V, Budagavi M (2013) A comparison of CABAC throughput for HEVC/H.265 VS. AVC/H.264. In: SiPS 2013 Proceedings. IEEE, pp 165–170

Tewari A, Gupta BB (2017) Cryptanalysis of a novel ultra-lightweight mutual authentication protocol for IoT devices using RFID tags. J Supercomput 73(3):1085–1102. https://doi.org/10.1007/s11227-016-1849-x

Wang X, He G (2011) Cryptanalysis on a novel image encryption method based on total shuffling scheme. Opt Commun. 284:5804–5807. https://doi.org/10.1016/j.optcom.2011.08.053

Wang Y, O'Neill M, Kurugollu F (2013) A tunable encryption scheme and analysis of fast selective encryption for CAVLC and CABAC in H.264/AVC. IEEE Trans Circuits Syst Video Technol 23:1476–1490. https://doi.org/10.1109/tcsvt.2013.2248588

Wang KH, Chen C-M, Fang W, Wu T-Y (2018) On the security of a new ultra-lightweight authentication protocol in IoT environment for RFID tags. J Supercomput 74:65–70. https://doi.org/10.1007/s11227-017-2105-8

Wiegand T, Sullivan GJ, Bjontegaard G, Luthra A (2003) Overview of the H.264/AVC video coding standard. IEEE Trans Circuits Syst Video Technol 13:560–576. https://doi.org/10.1109/tcsvt.2003.815165

Winkler S (2005) Digital video quality: vision, models and metrics. Wiley, Chichester

Xiph.org (2018) Xiph.org :: Derf's Test Media Collection. https://media.xiph.org/video/derf/. Accessed 6 Jan 2019

Zadeh LA (2015) Fuzzy logic—a personal perspective. Fuzzy Sets Syst 1:1–17. https://doi.org/10.1016/j.fss.2015.05.009

Zhang G, Liu Q (2011) A novel image encryption method based on total shuffling scheme. Opt Commun 284:2775–2780. https://doi.org/10.1016/j.optcom.2011.02.039