Identifying Optimal Encryption Effort to Improve Battery Life in Mobile Wireless Devices

Christopher Mansour and Danai Chasaki
Department of Electrical and Computer Engineering, Villanova University, Pennsylvania, United States
cmansour@villanova.edu; danai.chasaki@villanova.edu

ABSTRACT
Multimedia streaming applications are increasing rapidly due to the widespread use of wireless communications channels. Such applications have to be designed with both security and battery life in mind. Secure video applications are the most common culprit of battery drainage. In this article, an application layer algorithm that runs on the transmitting device is presented in order to adjust the amount of encryption performed depending on the wireless channel's conditions. This algorithm trades-off between security and power in wireless video communications. It achieves the desired security requirements while saving on power at the same time. The algorithm is validated and verified through a series of extensive simulations.

KEYWORDS: MULTIMEDIA, SECURITY, WIRELESS

CONCLUSION
Real-time multimedia streaming is nowadays growing rapidly and the demand on real-time multimedia applications is increasing tremendously. This will require an increase in computational resources and thus, higher power consumption. At the same time, such applications require the implementation of security techniques in order to protect the transmitted video streams. This is due to the fact that most of today's consumer devices use wireless channels for real-time multimedia streaming. However, applying security algorithms is a resource intensive task and thus power consuming as well. The analysis and experiments provided in this article lead to the design of an algorithm that gets feedback from the channel concerning its quality and provides it to the sender. As a result, the sender has the ability to specify the exact degree of encryption that is essential to achieve the mandatory security requirements. By following this approach, the sender is able to save on power while still meeting the video quality and security requirements. Consumers will be able to use their devices more efficiently than before. The design proposed in this article can be considered as a building block to the process of developing more sophisticated protocols that can run at different layers on the sender's side, as well as on the receiver's side. Such protocols can serve as self-learning protocols which contribute to the development of a smarter network, which is able to evaluate the conditions of the communication media and determine the exact amount of the required computation.

Quick Response Code

Access this article online

Paper URL: http://ijrws.org/index.php/ijrws/issue/view/15
ISSN: 2320-3617

Citation
I. INTRODUCTION

The new smartphone and tablet era has brought along massive exchange and streaming of video content in wireless networks [1]. The demand for multimedia applications such as Spotify, Netflix, and YouTube is increasing tremendously and have gained a great acceptance among smartphone users. Recent studies [2] report that there are more than six billion hours of YouTube video views every month out of which 25% come from a wireless device. Other studies show that over 50% of the peak bandwidth used in North America in the last mile of the subscriber network was used for paid subscriptions of video-on-demand services [3]. This increasing demand is accompanied with a lot of security issues. Security, in the terms of authentication and privacy, is very important for the commercial environments in order to protect companies’ intellectual property and users’ private data.

From a consumer perspective, security as well as battery life and performance are critical. On the one hand, the transmitted information should be encrypted so that unauthorized users, eavesdroppers, are unable to reconstruct it. On the other hand, depending on the strength and type of the cryptographic algorithm used during the information exchange, there is a corresponding power consumption overhead which decreases the battery life. N. Ruangchaijatupon and P. Krishnamurthy in [4] show that 75% of the battery power of a handheld device will be drained if 13.6 KB of data are encrypted using a 32-bit key Blowfish algorithm.

In the case of wireless communications, mobile wireless devices are usually constrained in terms of battery [5] and computational resources compared to traditional general-purpose processors. Moreover, wirelessly transmitted data needs to be encrypted if the existence of a passive attacker eavesdropping on the wireless link cannot be denied. Traditionally, all wireless packets are being encrypted regardless of the quality of the channel link. However, this article shows that if the wireless channel conditions are taken into account, time and power consumption can be reduced at the sender's side while ensuring strict security guarantees at the legitimate receiver.

Depending on the quality of the wireless link, the sender has the ability to achieve the same degree of security while encrypting a smaller percentage of the video frames. The proposed algorithm provides the application with the optimal degree of encryption that needs to be applied to a given video stream. An eavesdropper sniffing on a lossy channel would receive a highly distorted video after all; thus a small percentage of encryption can be enough to make the video useless to the attacker. In order to assess the overall channel's conditions, the sender would estimate the channel's SINR.

The main contribution of the proposed work is the design of an application layer algorithm that runs at the sender's side. The algorithm is based on the channel's feedback to determine the video frame encryption rate that is sufficient to meet the security requirements of the particular application.

The remainder of this article is organized as follows. Section II of this article introduces some of the related work in the field of multimedia streaming. The encryption overhead in video communications is described in Section III. The proposed analytical model is discussed in Section IV. The proposed algorithm is explained in Section V. The simulation and results are presented in Section VI and Section VII summarizes and concludes this article.

II. SECURITY IN MULTIMEDIA STREAMING

Video encoding standards, such as MPEG-4 or H.264/AVC [6], [7] provide guidelines on the encoding of video clips for transmission over communication systems. Video flows consist of different frames, namely I, P, and B frames [8]. The size of these frames and their succession depend on the video source and the encoder parameters. I, P and B frames provide different levels of encoding and thus, some protection against transmission losses. The periodic sequence of these frames is called a Group of Pictures (GOP), which is used as a fixed sequence by the encoding process of most of the real-time applications [8]. In each GOP, the first frame is an I frame which contains the entire image and can be decoded independently of other frames within the same GOP. Following the I frame, comes a sequence of P frames and B frames. The P frames are the predicted frames; they contain intra-coded parts and motion vectors calculated depending on the previous I or P frames. The B frames are coded depending on previous and successive frames. The P and B frames use the I frame as a reference to encode information.

The critical difficulty of the decoder is handling the missing frames and that is due to the fact that the frame losses affects the quality of the received video [9]. Typically, at the receiver's side, the missing frame is substituted with the last decoded frame [9]. Using this method, the difference between the substitute frame and the original one can easily be determined, and thus, the video quality can be evaluated.

The problem of developing analytical models for the communication process over wireless channels has been addressed extensively in the literature [9]. Single and multi-hop wireless network models take into account queues, back-off stage dependence of collision probabilities, and the correlation between departure and arrival processes of adjacent nodes in order to optimize end-to-end delay and throughput. In [10], the authors propose few techniques that balance the energy consumption and packet delay in wireless visual sensor networks.

Concerning security, [11] proposed a privacy preserving continuous multimedia streaming in mobile ad-hoc networks (MANETS) in order to enhance the degree of privacy in MANETS in addition to link duration and overall system performance. Moreover, several encryption techniques have been developed and applied on wireless communications channels. These techniques improved the performance of the naïve algorithm which encrypts every single byte of the video packets. Examples of such algorithms include the selective encryption algorithm which encrypts parts of the video being
transmitted, the Zig-Zag algorithm [12] which compresses and encrypts without affecting the image and video quality after decryption, the Video Encryption Algorithm (VEA) [13] which uses a secret key randomly changing the sign bits of encoded differential values of I, B, and P pictures, and other pure permutation techniques [12].

In [14], a partial encryption scheme was proposed in order to hide important MPEG-4 contents. In [15], the authors proposed several different approaches to achieve multimedia privacy by combining multimedia encryption and entropy coding to a single process. They proposed the use of Multiple Huffman Tables (MHT) alternately in a secret order which is created using a pseudorandom number generator randomizing the process. According to this study, reasonably higher levels of security and lossless media compression efficiency have been achieved. However, the cryptographic strength in terms of the key length may make the proposed MHT approach susceptible to channel errors.

The approach presented in this article, takes the idea of selective encryption one step further. This approach does not randomly select the number of frames to encrypt rather it is a cross-layer approach to help make the decision, taking the advantage of the priori information about the wireless channel conditions for a given video clip transmission. In the experimentation phase, an essential look at the trade-off between encryption effort and video quality is performed. The video quality is linked to the video distortion taking into consideration the video encoding and decoding details.

III. ENCRYPTION OF VIDEO TRANSMISSION

Public key cryptography is not suitable for secure real time video communications since its operations are computationally very intensive and thus too slow. Advanced Encryption Standard (AES) is widely used to send encrypted video for playback in numerous devices. For example, MPEG DASH and HLS encryption utilize AES to protect content from unauthorized streaming and illegal redistribution. Moreover, the iTunes Music Store is composed of XML-based pages, most of them encrypted using 128-bit AES.

Power consumption is particularly important for battery-powered devices like smartphones and tablets. Most of the applications favored by users involve downloading and streaming of data. However, these applications are the most common culprits of battery drain while data is being pulled from cellular or Wi-Fi networks. At the same time, data is encrypted during video transmission to protect privacy or intellectual property. This involves an additional power consumption which would be added to that of the video communications.

The energy consumption of encryption algorithms depends on several factors including the hardware platform, the cryptographic algorithm used, and the key size. This consumption is subject to an increase every year since the size of the keys used increases in order to achieve the same degree of security.

Table 1. Energy Consumption for 128-bits block AES

<table>
<thead>
<tr>
<th>Machine</th>
<th>CPU Cycles</th>
<th>Energy Consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>StrongARM SA-1110 [16]</td>
<td>639</td>
<td>1.17 µJ</td>
</tr>
<tr>
<td>IBM’s Cell Broadband Engine [17]</td>
<td>198.4</td>
<td>0.02 µJ</td>
</tr>
</tbody>
</table>

Table 1, provides the CPU cycles needed for AES-128 implementations on different hardware platforms. The energy consumption of the AES on a 128-bit block of data can be calculated by using the CPU cycles, the average current drawn on each cycle, and the voltage at which each CPU operates. Intel’s StrongARM, for example, operating at a voltage of 1.45V consumes approximately 180mA per cycle. It runs at 133 MHz, and since AES encryption of a 128-bit data takes approximately 639 cycles [16], the overall energy consumption for that block amounts to 1.17 µJ.

Table 1, shows that AES encryption requires a large number of CPU cycles to encrypt each block of data. It is particularly important for video streaming applications, which encrypt massive amounts of data, to determine the exact number of data blocks to be encrypted. As a result, this will lead to a decrease in the energy consumption.

In the following section, the mathematical framework that helps the sender determine the percentage of video frames that need to be encrypted is discussed. Overall, even if a small percentage of video packets are eventually not encrypted, the proposed algorithm saves an equivalent percentage of CPU cycles and energy cost, and consequently battery life.

IV. ANALYSIS

In this section, the channel's conditions between the sender and the receiver are first estimated, and then the video distortion an eavesdropper would experience is analyzed. This takes into consideration the packet losses due to interference and the invisible packets due to encryption. Later in the experiments, the effect of the location of the eavesdropper, different percentage of encryption of H and B frames, and different degrees of encryption in many scenarios including single-hop and multi-hop environments are studied. The analysis is based in the analytical framework developed by the authors in [9], where they compute the expected distortion of the video in the presence of interference and hence, map the packet losses due to interference to the video quality. In this article, the model is adapted to account for additional losses due to the inability of the eavesdropper to decrypt the percentage of the encrypted packets. The analytical process is described as follows:

- The expected value of the Signal-to-Interference-and-Noise-Ratio (SINR) is computed.
- The SINR is mapped to the packet success rate metric and hence to packet losses.
- Packet losses are mapped to video transmission distortion.
- Distortion is mapped to Peak Signal-to-Noise-Ratio (PSNR), an objective video quality metric.
A. Signal-to-Interference-Noise -Ratio

In regards to the channel access system, when considering a wireless ad-hoc network with one-hop sender/receiver pairs, the time is assumed to be divided into equal-length time slots. The length of each time slot is equal to the time it takes to transmit the largest allowable packet in the system. Let \( P_n \) denote the transmission probability of a node \( n \).

Passive distributed attacker \( E \) eavesdropping at the wireless medium is assumed, while sender \( S(x,y,t) \) is transmitting a video clip to receiver \( R(x_0,y_0) \). In order to make conclusions about the video quality a potential eavesdropper would receive, the overall SINR of the channel should first be estimated.

The SINR is defined as:

\[
SINR = \frac{P_{se}}{N + \sum_{k \in l} P_{se}}
\]

where \( N \) is the noise power and \( I \) is the set of nodes interfering with the eavesdropper.

\( P_{se} \) is the received power at the eavesdropper of the signal sent by the sender \( S \). \( P_{se} \) is the received power at the eavesdropper of a signal transmitted by any node \( k \). The received power \( P_{se} \) can be calculated as:

\[
P_{se} = \frac{P_t s G_s G_e}{d_{se}^2} \cdot \left( \frac{\lambda}{4\pi} \right) \cdot \frac{1}{d_{se}^2}
\]

where \( P_t \) is the transmission power. All the nodes are assumed to use the same transmission power \( P \), i.e. \( P_t = P \) for all \( s \). \( G_s \) and \( G_e \) are the transmission and reception gains, respectively. \( \lambda \) is the wavelength of the signal, \( d_{se} \) is the distance between the sender \( S \) and the eavesdropper \( E \) and \( L \) is the system loss.

To compute the expected value of SINR we assume that \( R \) is the maximum communication range. If a uniform node distribution is assumed, the probability density function of the distance is given by:

\[
f_d(t) = \frac{2t}{R^2}, \quad 0 < t < R
\]

The expected value of the SINR can then be computed as [9]:

\[
E[SINR] = \sum_{l=0}^{m} E[SINR|l \text{ interferers}] \cdot P\{l \text{ interferers}\}
\]

where we assume that the maximum number of neighbors \( m \), and \( l \) is the number of interfering neighbors.

The number of interferers follows a binomial distribution with parameters \( m \) and \( p_o \)

\[
P\{l \text{ interferers}\} = \binom{m}{l} p_o^l (1-p_o)^{m-l}, l = 0,1,\ldots,m
\]

where \( p_o \) is the transmission probability of a node as explained earlier.

Following the analysis steps in [9] leads to:

\[
E[SINR] = \sum_{l=0}^{m} \left( \frac{m}{l} \right) p_o^l (1-p_o)^{m-l} \times \left( \frac{2}{R^2} \right)^{l+1} \overbrace{\int_{0}^{\infty} \int_{0}^{\infty} \int_{0}^{\infty} \int_{0}^{\infty} dt_0 \cdot dt_1 \cdot dt_2 \cdot dt_3 dx}^{l-times}
\]

wherel \( U(x) = \left( \frac{x N L (4\pi)^2}{PG_s G_e \lambda^2} + x \sum_{k=1}^{l} \frac{1}{2} \right)^2 \). The last equation can be solved numerically to compute the expected value of the SINR.

B. Packet Success Rate

The eavesdropper will experience packet loss, that are dependent on the following:

- The number of concurrent video transmissions in the wireless channel.
- The percentage of video packets that the sender decides to encrypt. This is actually due to the fact the eavesdropper lacks the possession of the encryption key to be used in order to decrypt the encrypted packets.

The packet success rate \( P_t \) is computed as a function of the expected value of SINR according to the following [9]:

\[
P_t(S) = \left[1-P_{e1}(S)\right]^B
\]

where

\[
P_{e1}(S) = 1 - \left( \frac{1}{\sqrt{2\pi}} \int_{-\sqrt{2\pi}}^{\sqrt{2\pi}} 2\phi(x+\sqrt{2S})-\int_{-\sqrt{2\pi}}^{\sqrt{2\pi}} \phi(x+\sqrt{2S})-1)^{(N-1)} \times \exp\left(-x^2/2\right) dx \right)
\]

If \( r \) is assumed to be the rate at which the sender encrypts the video packets, where \( 0 < r < 1 \), then the packet success rate at the eavesdropper becomes:

\[
P'_t(S) = (1-r) \cdot \left[1-P_{e1}(S)\right]^B
\]

C. Estimation of Distortion

At this stage, having computed the packet success rate by taking into account both the interference and the packet losses, the distortion model of [9] can directly be used. \( P_t \) and \( P'_t \) as given by (9) can directly be substituted. The model is a Markov chain model that maps missing packets to video distortion. The video distortion will be computed as the mean squared error (MSE) of the difference between the missing frame and the substitute frame [18].
D. Peak-Signal-to-Noise-Ratio

In all the results presented in this work, the PSNR is used. This is due to the fact that it is an objective video quality measure [19]. The relationship between distortion and PSNR (in dB) is given by the following:

\[
PSNR = 10 \log_{10} \frac{255}{\sqrt{\text{Distortion}}} \quad (10)
\]

Using (10), the PSNR metric would be calculated throughout the experimentation process in the article.

V. ALGORITHM DESIGN

This section describes the design of the proposed algorithm which runs on the sender’s side. The algorithm implements a method that guides the sender on deciding the necessary percentage of the video packets to be encrypted. The application layer has to provide the sender with the acceptable video quality at a potential eavesdropper’s side as input to the algorithm. This is a metric that depends solely on the level of security of the video transmission. For instance, a military video transmission required a higher level of security compared to that of a normal video on a social media website. According to Table 2., which shows the ITU-R quality and impairment scale, it would be essential for the eavesdropper in some video applications to have a bad video quality, i.e. less than 20 dB. For other applications, even a PSNR of a value greater than 20 dB could be acceptable, in case the goal is to annoy the eavesdropper and compromise the corresponding viewing experience.

Table 2. ITU-R Quality and Impairment Scale

<table>
<thead>
<tr>
<th>PSNR [dB]</th>
<th>MOS</th>
<th>Impairment</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;37</td>
<td>5</td>
<td>(Excellent)</td>
</tr>
<tr>
<td>31-37</td>
<td>4</td>
<td>(Good)</td>
</tr>
<tr>
<td>25-31</td>
<td>3</td>
<td>(Fair)</td>
</tr>
<tr>
<td>20-25</td>
<td>2</td>
<td>(Poor)</td>
</tr>
<tr>
<td>&lt;20</td>
<td>1</td>
<td>(Bad)</td>
</tr>
</tbody>
</table>

Table 3. Encryption Rate Look-Up Table

<table>
<thead>
<tr>
<th>r (%)</th>
<th>PSNR [dB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>PSNR(0)</td>
</tr>
<tr>
<td>(\gamma_1)</td>
<td>PSNR((\gamma_1))</td>
</tr>
<tr>
<td>(\gamma_2)</td>
<td>PSNR((\gamma_2))</td>
</tr>
<tr>
<td>(\gamma_3)</td>
<td>PSNR((\gamma_3))</td>
</tr>
<tr>
<td>100</td>
<td>PSNR(100)</td>
</tr>
</tbody>
</table>

Algorithm 1 Determine the encryption percentage “r” for a given PSNR, requirement

Input: PSNRs
Output: “r”

1. \(\text{SINR} \leftarrow \text{SINR computation according to (6)}\)
2. for \(r = 0 \text{ to } 100\) do
3. \(\text{compute } P_r \text{ from (9)}\)
4. \(\text{compute PSNRs as explained in [9]}\)
5. end for
6. save \((r, \text{PSNR})\) pairs in “Table. 3”
7. use input PSNR, as an index in the second column to lookup the corresponding “r” from “Table. 3”

Algorithm 1 takes as an input the PSNR requirement for the eavesdropper (in dB). After estimating the channel conditions, the algorithm outputs the encryption rate at which the video frames need to be encrypted at the sender’s side. For every possible encryption rate (integers ranging between 0 and 100), the sender calculates the corresponding PSNR at the eavesdropper following the equations discussed in Section IV and constructs a table similar to Table 3. This table is indexed in the end of the algorithm, using the input reference \(\text{PSNR}_s\), to determine the output encryption rate \(r\).

VI. EVALUATION

In this section, the proposed algorithm is validated using extensive simulations. The video distortion is measured and quantized. This distortion is due to the packet loss from the interference on the channel and due to the loss of packets as a result of the absence of the encryption key to be used in the decryption process. It is obvious that the quality of the video stream is inversely proportional to the amount of packets lost.

A. Experimental Framework

In order to perform the analysis, the Network Simulator NS-3 [20] was used on a UNIX based machine. The framework implemented consists of several nodes connected to each other in a wireless connection. A virtual bridge is created on the UNIX machine in order to implement a link between the machine and the framework. The quality of the video transmitted over a real communication network is evaluated using EvalVid [21], which is a popular toolset for that purpose. This tool allows the assessment of the quality of the video by calculating the PSNR and the Mean Opinion Square (MOS) Metrics [22]. The PSNR is a differential metric, which is similar to the Signal-to-Noise-Ratio (SNR) in communication systems. The PSNR metric can be mapped to the MOS metric. The latter is a five-degree scale that depicts the human perception of the video quality and ranges from bad to excellent. It should be noted that, the lower the PSNR is, the lower the MOS is and thus the higher the distortion in the video will be. Table 2. shows the ITU-R quality and impairment scale. It also maps possible PSNR values to MOS values and their impairment equivalent [21]. According to Table 2., an excellent video quality is guaranteed when the average PSNR exceeds 37 dB. However, in order to get an acceptable video quality, the average PSNR should be at least 26 dB.
An initial video file composed of raw data (YUV frames) is retrieved and used in the simulation. Within the EvalVid toolset, the YUV video is transformed to the MP4 format using the ffmpeg codec and then to the ISO MPEG4 format using MP4Box codec. Mp4trace tool is used in order to transmit the resulting MPEG4 video to the NS-3 framework. In the framework, the channel’s conditions are varied by changing the bit-error rate and changing the amount of traffic being exchanged. The additional traffic comes from different nodes communicating with each other. These conditions are considered to be the interference that we take into consideration throughout the article and will alter the quality and speed of the video being exchanged.

In addition to interference, AES encryption is implemented at the sender node in order to study the effect of different degrees of encryption on the quality of the video as seen by the eavesdropper. The degree of encryption resembles the percentage of frames to be encrypted at the sender, also scaled from 0 to 1. The encryption is applied in different scenarios; encrypting I frames and selectively encrypting other frames (P and B frames), and only encrypting B frames. It should be expected that when encrypting I frames; the quality of the video should be degraded due to the fact that I frames carry a large amount of information. In the experiments, EvalVid was modified so that real packets are sent to the NS3 framework, and real packets encryption is applied. Each node in the NS3 framework creates a dump file using the trace utility implemented in NS3. Using these dump files, and comparing the video frames or packets captured by the eavesdropper, the required metrics are calculated and the effects of the encryption and interference is quantized using the etmp4 and psnr tools in EvalVid.

The first step in each of the evaluations is the calculation of the reference PSNR. The reference PSNR is the PSNR of the coded and decoded video without the transmission errors, i.e. without any packet or frame loss. The next step is the reconstruction of the video as seen by the eavesdropper and later the evaluation of the quality and performance metrics using the aforementioned tools. The tools use the dump files generated by the nodes in the framework and produce log files that contain the loss description, delay description, and the data rate at the sender and receiver. In addition to that, these tools produce the average PSNR value and its corresponding standard deviation.

The video encoding parameters used are as follows: the GOP size is 15 frames; each frame size is CIF (352 x 288) with a rate of 30 fps, and the maximum transmission unit (MTU) is equal to 1000.

The aforementioned steps and evaluations are executed in different scenarios that cover different environments for video transmission. These scenarios include, an environment with nodes in a constant position or in mobility. Multi-hop environment in which there are multiple hops between the sender and the receiver. Different levels of encryption are used in each of the aforementioned environments.

B. Constant Position

In this experiment [23], the sender, the receiver and the eavesdropper are placed in a constant position. The first step was to study the interference effect on the video quality as seen by the eavesdropper sniffing on the channel. Fig. 1 shows the variation of the average PSNR and the overall percentage loss of frames as a response to the percentage of interference on the channel.

Fig. 1. Average PSNR and overall percentage loss with respect to the percentage of interference on the channel

Fig. 1. shows that, as the interference on the channel increases, the percentage of packet losses in the channel increases, and thus, the average PSNR of this video will eventually decrease. By analyzing the trend of the average PSNR and that of the overall percentage loss, it is clear that with the increase of the interference on the channel, the quality of the video deteriorates. According to Table 2, for a video to be fair or slightly annoying the average PSNR should be at least 26 dB. However, Fig.1. shows that with a percentage of interference being greater than 55%, the average PSNR decreases below the accepted value and thus the quality of the video is not acceptable. Therefore, an eavesdropper sniffing on the channel with high level of interference will not be able to reconstruct and watch the video, and the quality of the video at the receiver will be bad.

In order to further evaluate the effect of interference on the video quality, the MOS metric was evaluated and the results are provided in Fig. 2. The results shown in Fig. 2. prove that as the percentage of interference on the channel increases, the percentage of frames with an acceptable MOS value decreases. Furthermore, with an interference greater than 50%, the percentage of poor and bad frames (MOS values 1 and 2) totals to more than 60% of the total frames. This means that the frames are of an annoying quality and the receiver or eavesdropper sniffing on the channel will not be able to reconstruct and watch the video.
The results shown in Fig. 1. and Fig. 2. show that, with the increase of the percentage of interference on the channel, the video quality deteriorates. Therefore, an eavesdropper sniffing on a channel with high interference will experience an annoying video quality.

However, in real-time multimedia applications, encryption is also introduced at the sender in order to achieve security. The following experiment studies the effect of both encryption and interference on the quality of the video captured at the eavesdropper. Different rates of encryption were applied at the sender on the video before transmission to the receiver. It should be noted that an eavesdropper does not possess the key used for encryption and thus the packet loss at his point could be the result of both, lack of the key as well as the interference on the channel. Furthermore, encrypting the video frames or packets affects the quality of the video at the eavesdropper but not at the receiver. This is because the receiver does have the encryption key in order to decrypt the received packets and the packet loss at the receiver are the results of interference only. The encryption process will only cause the delay of the packets received by the receiver.

Fig. 3. reflects the results obtained after introducing encryption. It shows the average PSNR for different interference percentages as a function of different encryption rates applied to the video packets at the sender. It is clear that for various percentages of interference on the channel, as the rate of encryption increases, the average PSNR value decreases and thus the quality of the video at the eavesdropper becomes worse. Therefore, both the interference and encryption affect negatively the quality of the video at the eavesdropper.

Additionally, the analysis of Fig. 3. shows that for each interference percentage, there is a cutoff point at which the average PSNR of the video is below the accepted amount (26 dB) and therefore there should be no need for an additional encryption overhead.

Depending on the results shown in Fig. 3., Table 4. is constructed in order to specify the exact amount of encryption to be applied for a certain PSNR requirement. The values in this table shows the maximum percentage of encryption to be applied to guarantee that the quality of the video is annoying if reconstructed by an eavesdropper. By deciding the exact amount of encryption required, the sender in this case would save power and energy consumption.

Table 4. Percentage of Encryption to be Applied in order to Meet Certain PSNR Specifications

<table>
<thead>
<tr>
<th>Interference (%)</th>
<th>PSNR [dB]</th>
<th>28 – 26</th>
<th>26 – 24</th>
<th>Below 26 dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10%–25%</td>
<td>30%–40%</td>
<td>45%–50%</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>10%–25%</td>
<td>30%–35%</td>
<td>40%–45%</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>0%</td>
<td>10%–30%</td>
<td>35%–40%</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>0%</td>
<td>10%–30%</td>
<td>35%–40%</td>
<td></td>
</tr>
</tbody>
</table>

An example which shows the effect of interference and encryption of the video being transmitted is provided in Fig. 4. Fig. 4a. shows a snapshot of the video to be transmitted by the sender, a snapshot of the video constructed by an eavesdropper, and that received by the receiver in the case where the video is sent without any encryption. It is noticed that both the receiver and the eavesdropper have the same experience. The quality in this case is related to the packets lost due to interference on the channel. However, Fig. 4b. shows a snapshot of the video to be sent, snapshot of that captured by an eavesdropper, and that received by the receiver in the case of an encrypted video.
Fig. 4. Video as seen by the eavesdropper and receiver: (a) in the case were no encryption was applied, (b) in the case were 80% encryption was applied

Fig. 4b. shows that the video captured by the eavesdropper in the case where encryption is applied is useless since the latter does not possess the encryption key to decrypt the encrypted video whereas the receiver will still receive the video and reconstruct it. The only losses at the receiver's side are the result of interference on the channel.

Furthermore, in order to analyze the effect of encrypting B frames on the quality of the transmitted video, the sender was programmed to encrypt the B frames only. Fig. 5. shows the variation of the mean PSNR and the overall percentage loss if only a certain percentage of B Frames were encrypted. Fig. 5. shows that, the encryption of the B frames only will have no additional effect on the video. Both the average PSNR and the overall percentage loss become worse when the encryption percentage is increased.

Fig. 5. Average PSNR and overall percentage loss of frames as a function of different encryption rates applied on B frames

In order to further assess the video quality of the above scenario, the MOS metric was also calculated when the B-frames were the only frames being encrypted by the sender. Fig. 6. shows the MOS variation as a function of the different degrees of encryption applied on the B frames.

Fig. 6. MOS metric values with respect to the degree of encryption of B-frames

The results obtained from experimenting with B frames can also be used in order to construct a table similar to Table 4. The latter can be used to save on power and delay which can be added as a result of encryption.

In order to ensure that the proposed approach works accurately, the effect of the location of the eavesdropper with respect to the sender should be analyzed. In the following experiments, the location of the eavesdropper with respect to the sender was varied and the data was collected at each location of the ten locations that were chosen in a way to cover most of the scenarios. The same process was done in order to reconstruct the video and assess its quality by evaluating the mean PSNR and MOS values. The results were as shown in Fig. 7. and Fig. 8.

Fig. 7. Mean PSNR and overall percentage loss as a function of different locations of the eavesdropper

Fig. 7. shows that as the distance between the eavesdropper and the sender increases, the video quality becomes worse. This is clear by if the trends of the average PSNR and overall percentage loss was observed. The same
Identifying Optimal Encryption Effort to Improve Battery Life in Mobile Wireless Devices

The results provided in Fig. 8. also prove that with the location of the eavesdropper being far from the sender, the quality of the video becomes worse. Therefore, for an eavesdropper being closer to the receiver or the sender, the quality of the video captured is better; this will cause an additional encryption and thus an additional power consumption and delay overhead. Hence, if the application layer was able to determine such conditions from the corresponding channel, including the percentage of interference and the location of the eavesdropper, it could be able to determine the exact degree of encryption required saving power and delay overhead on the consumers’ device. A similar table to Table 4. can be constructed and used by the sender to save power consumption while assuring a certain security requirement.

C. Mobility Implementation

In this experimental phase, the nodes are considered to be in motion (mobile) and not in a constant position. The framework is edited and mobility properties are added to the nodes with a certain perimeter and speed attributes. The same process was followed, as in the case of constant node positioning in order to estimate the average PSNR and MOS metrics.

The first step was to evaluate the effect of different percentage of interference in the channel on the quality of the video captured by the eavesdropper. Fig. 9. shows the variation of the average PSNR and overall percentage loss of frames as a function of the percentage of interference on the channel. Fig. 10. shows the MOS evaluation when the nodes are mobile. A similar behavior to that in the case of constant node positioning earlier was noticed. As the interference increases, the average PSNR decreases and the overall percentage loss of the frames increases. It is observed that, with a percentage of interference exceeding 75% the average PSNR drops below 26 dB and thus the video captured by the eavesdropper is useless. The same behavior is observed in the MOS graph where the percentage of bad frames increase with the interference percentage.

D. Multi-Hop Implementation

In the following series of experiments [24], several hops are implemented on the route between the source and the destination. The received video packets pass through multiple intermediate nodes in order to reach the destination.

The first step is to study and illustrate the effect of interference on the video quality seen by an eavesdropper
sniffing on the channel. The variation of the average PSNR and the overall percentage loss of frame is the same as that obtained from earlier experiments. The average PSNR decreases and the overall percentage loss increases with the increase of the interference percentage on the channel. The same behavior is also obtained for the MOS metric. The percentage of bad frames increases with the increase of the degree of interference on the channel. Moreover, applying different encryption rates on the video packets before their transmission to the receiver results in a similar behavior to that obtained earlier and the results are reflected in Fig. 11.

![Average PSNR for Different Interference Percentages](image)

Fig. 11. Average PSNR for various interference percentages with respect to the rate of encryption in the scenario of multi-hop nodes

Fig. 11. shows that for each interference percentage, as the percentage of encrypted frames by the sender increases, the average PSNR value decreases.

Analysis of such results leads to the construction of Table 4, which serves as a look-up table including the cutoff points for each PSNR requirement. The sender could use such a table in order to specify the exact encryption rate required so that an eavesdropper sniffing on the channel would lack the ability of reconstructing the video. At the same time, by applying this encryption rate, processing overhead is decreased resulting in a decrease in the delay and power consumption.

It is noticed that in the case where multiple hops are presented between the sender and the receiver, the interference percentage has a greater effect on the average PSNR and the overall percentage loss. Moreover, for an interference percentage greater than 30% the average PSNR value decreases below 26 dB and thus the eavesdropper would obtain a useless video. Hence, there is no need for any additional encryption. Therefore, if the look up table was constructed the sender can save a lot of delay and power consumption.

Finally, if the scenarios presented in this article were implemented on the IBM’s Cell Broadband Engine, an additional 617.50 µJ energy consumption will be added to the energy consumption of the transmission process. However, if the protocol discussed in this article was implemented, at least half of the additional energy consumption will be saved. This is proved in Table 5, which shows the energy consumption that is saved in different application scenarios with respect to the quality of the channel. It is observed that in the case of a military application where high security requirements should be guaranteed, if the channel is of a high quality then 123.50 µJ would be saved. This is because in this case 80% encryption rate is enough to ensure that the average PSNR of the video at the eavesdropper is below 14 dB. Therefore, instead of wasting additional 617.50 µJ to perform a complete encryption, selective encryption with a rate of 80% will save on energy. Additionally, if the channel is of medium quality, 50% encryption rate can be enough to guarantee a useless video at the eavesdropper. The latter results save 308.75 µJ. However, in the case where a commercial application exists with medium security requirements, if the channel is of a high quality then 25% encryption would guarantee a video quality of an average PSNR below 26 dB at the eavesdropper. This will save 463.12 µJ. Moreover, in the same scenario, encryption rates of 15% and 5% are enough to guarantee the security requirements in the case of a channel with average quality and bad quality respectively. This will save 524.85 µJ and 586.62 µJ respectively. The same behavior is observed in the third scenario corresponding to applications with low security requirements.

<table>
<thead>
<tr>
<th>Applications</th>
<th>High</th>
<th>Average</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>Military Application</td>
<td>123.50 µJ</td>
<td>308.75 µJ</td>
<td>401.37 µJ</td>
</tr>
<tr>
<td>Commercial Application</td>
<td>463.12 µJ</td>
<td>524.85 µJ</td>
<td>586.62 µJ</td>
</tr>
<tr>
<td>Low Security Application</td>
<td>555.75 µJ</td>
<td>586.62 µJ</td>
<td>617.50 µJ</td>
</tr>
</tbody>
</table>

VII. CONCLUSION

Real-time multimedia streaming is nowadays growing rapidly and the demand on real-time multimedia applications is increasing tremendously. This will require an increase in computational resources and thus, higher power consumption. At the same time, such applications require the implementation of security techniques in order to protect the transmitted video streams. This is due to the fact that most of today’s consumer devices use wireless channels for real-time multimedia streaming. However, applying security algorithms is a resource intensive task and thus power consuming as well. The analysis and experiments provided in this article lead to the design of an algorithm that gets feedback from the channel concerning its quality and provides it to the sender. As a result, the sender has the ability to specify the exact degree of encryption that is essential to achieve the mandatory security requirements. By following this approach, the sender is able to save on power while still meeting the video quality and security requirements. Consumers will be able to use their devices more efficiently than before. The design proposed in this article can be
Identifying Optimal Encryption Effort to Improve Battery Life in Mobile Wireless Devices

REFERENCES


