Introduction

Nowadays, video and image compression is being widely used. Due to the increasing amount of using multimedia information, the security problem becomes important. The conventional method (compress first and then encrypt) does not meet the requirement of real-time processing due to the need of big amount of computational power.

Lightweight Encryption is a new idea which developed in recent years. This method will carry out encryption at the entropy coding step when compressing the multimedia content.

In this project, lightweight encryption will be implemented into the H.264 and JPEG standard software in order to increase their secure abilities.

Objectives

In this project, I will implement:

1. Secure H.264 video management system
   1.1 Secure Exp-golomb coding and secure UVLC coding in the H.264 encoder and decoder.
   1.2 User interface.
2. Secure JPEG photo management system
   2.1 Secure Huffman coding in the JPEG encoder and decoder.
   2.2 User interface.

Methodology

Secure Exp-golomb coding and secure UVLC coding in H.264

Use a pseudo-random number generator with seed $s$ to produce a key stream $K = k_1 k_2 ...$

when $i = 1$: $N_i$, if $k_i = 1$, then use leading ones $(1 1 1 ... 0 k_1 ... k_N)$ to encode the current data. If $k_i = 0$ then use leading zeros $(0 0 0 ... 1 k_1 ... k_N)$ to encode the current data.

Produce the encrypted bit stream

At the decoder side, after obtaining the seed $s$, it is easy to re-produce the key stream $K$, and decode the bit stream accordingly.

Secure Huffman coding in JPEG

Create a Huffman tree space by using tree mutation for each of the standard Huffman trees.

For each different set of data, carefully select 4 Huffman tables from Huffman tree space. Number them as 0, 1, 2, 3.

Produce a random sequence $k_1 k_2 ... k_N (i = 0, 1, 2, 3)$

Use $k_i$ Huffman table to encode the $i^*_{th}$ encountered symbol.

Key Results

Results of lightweight encryption in H.264

After encoding, people can only use the correct key to decode the video sequence with the secure H.264 decoder. As we can see in the figure, the original decoder cannot decode the encrypted video correctly.

Results of lightweight encryption in JPEG

After encoding, people can only use the correct key to decode the encrypted photo with the secure JPEG decoder. The left figure is generated by the original JPEG encoder and the right figure is generated by the Secure JPEG encoder.