Steganography in Live Video

Jonathan Browning
School of Engineering & Computing Sciences, Durham University

What is Steganography?

Steganography is the practice of embedding secret information inside of a cover object, such as an image, or in our case a live video stream, for the purpose of covert communication. Unlike cryptography, where an eavesdropper is easily able to detect that covert communication is taking place, steganography is designed to keep the covert communication undetected, with the eavesdropper believing that the innocuous cover object is the only communication that has taken place.
An example is shown in the diagram to the right. Alice wishes to communicate with Bob covertly over an insecure channel. Eve can intercept everything that is transmitted over the channel. To avoid suspicion, Alice hides her message inside an innocuous cover object (making it a *steganogram*), before sending this over the insecure channel. Eve sees the steganogram, but believes it is innocuous. Bob receives the steganogram, and is able to extract the secret message.

Research Questions

- **Is LSB embedding is an appropriate steganographic technique for use with live video streaming?**
- **Can the technique be developed to provide resilience against lossy video compression?**

Embedding Techniques

N-th least-significant bits (N-th LSB) encoding

Pixel values are stored as 8-bit integers. Simple least-significant bit (LSB) encoding replaces the least-significant bit of each pixel value with a bit from the message. This can be generalized to replace the N-th LSB with the message bit, with N between 1 and 8.

N LSBs encoding

N LSBs encoding also replaces the N-th LSB of each pixel value with a message bit. However, all the bits less significant than the N-th LSB are altered to maximise the change in pixel value that would be required for the N-th LSB to flip, as this would corrupt the message. This aims to provide robustness against lossy video compression.

Repetition codes

To provide further robustness against lossy video compression, the message is repeated many times, and the client takes a 'majority vote' on each bit of the message, to decrease the probability of errors in the decoded message.
An alternating repetition code was also implemented, whereby every other repetition is inverted. This aims to reduce any bias between the percentages of 0s and 1s that are correctly decoded.

Example

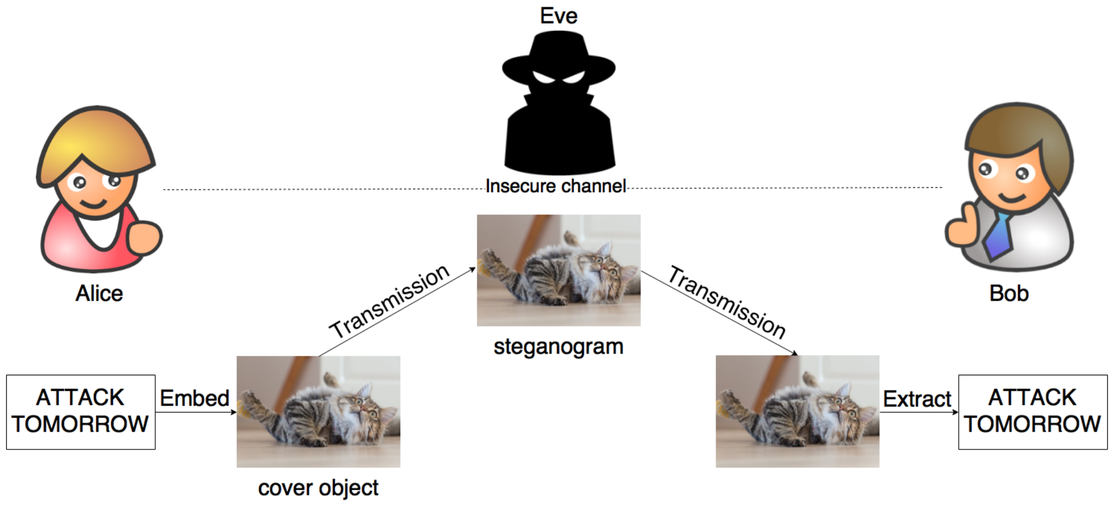


Figure 1: A simple example of steganographic communication between Alice and Bob.

Results

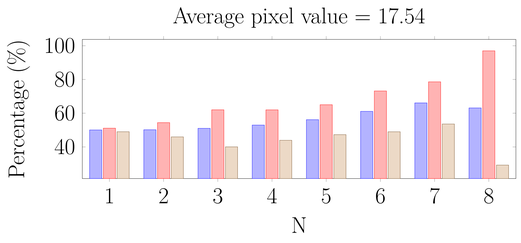


Figure 2: The percentages of message bits (blue), 0s (red) and 1s (biege) correctly decoded from an MJPEG stream using N-th LSB encoding.

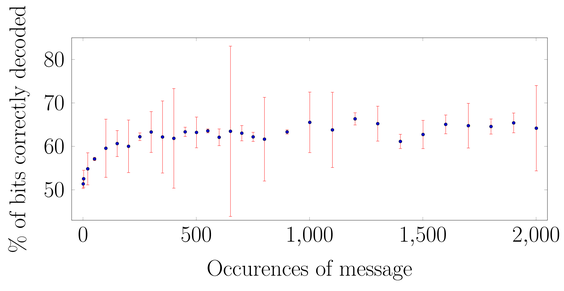


Figure 3: Robustness of 3 LSBs encoding with the simple repetition code. Error bars demonstrate the differences between the percentages of 0s and 1s that are correctly decoded.

One-way communication of text-based messages embedded using the N-th LSB encoding (N=1) was successfully implemented in the form of a Skype-like app, using lossless video transmission. Screenshots of the app are shown in Figure 4 (right). A maximum resolution of 128×72 was achieved. When subjected to lossy MJPEG compression, the messages were corrupted and no longer recognisable.
A random bitstring message was embedded using N-th LSB encoding and transmitted using MJPEG compression. N was varied between 1 and 8, and the percentages of message bits were recorded. These are presented in Figure 2. As N is increased, the percentage of of bits correctly decoded increases, but the quality of the image deteriorates. However, there is a skew between the percentages of 0s and 1s that are correctly decoded. This was found to be dependent on the original pixel values in the webcam feed. N LSBs encoding was found to almost completely eliminate this skew.
The use of a repetition code on top of N LSBs encoding was found to initially increase the percentages of message bits that were successfully decoded, however this improvement flattened off after approximately 200 repetitions, and did not increase much above 65%, still too low to transmit text reliably. The skew between 0s and 1s correctly decoded has increased again, as repetition amplifies any small skews present. The alternating repetition code substantially reduced this skew, but did not provide an overall increase in the percentage of bits correctly decoded.

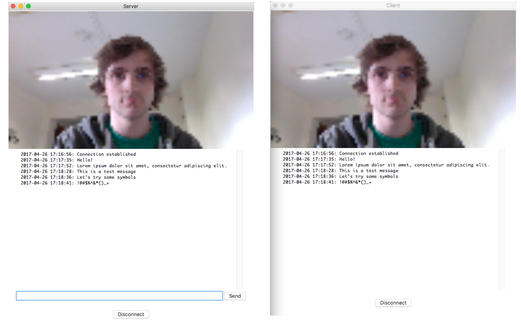
Uses of Steganography

Steganography has uses in a number of areas where the presence of secure communication needs to be remain undetected, including:

- Espionage
- Military communications
- Journalism in authoritarian countries
- Whistleblowing

The advantage of using live video as a cover object is the lack of evidence left behind: unlike images or complete video files, live video streams, such as video conferences, are not typically retained after they have been displayed to the recipient. Therefore, unless an attacker suspects the use of steganography in advance, they will miss the opportunity to undertake steganalysis (attempts to detect the use of steganography) of the video stream.

Prototype App



(a) Server call view **(b)** Client call view

Figure 4: Screenshots of the prototype app during a video stream.

Conclusions

LSB steganography in live video streaming was found to be achievable using lossless video transmission, but subject to severe limitations on resolution due to bandwidth and encoding time required.

LSB steganography was **not** found to be a feasible means of communication when used with lossy MJPEG compression, as while the transmission error was successfully reduced by the use of the repetition code, this was not enough to achieve reliable text-based communication.