Spread Spectrum Steganography

Matthew Seaworth, Sean Soderman, Dylan Soderman

August 4, 2014

1 Project Description

We are implementing a spread spectrum image steganography algorithm with three main components. The spread spectrum algorithm is slightly modified using a uniform distribution that is then spaced to increase the distance between bitvalues to the maximum possible distance between the two. Then afterwards it is transformed into a normal distribution and added to the cover image. On the other end, the image is processed to extract the original cover image from the steganography image. The extraction of the cover image is not exact and introduces bit errors. The noise is then duplicated on the other end with an identical seed and the closest stream is picked to extract the original message. At both ends error correcting is used before and after the encoding and decoding to help reduce errors.

2 Analysis

What I can tell so far is that the embedding algorithm provides low perceptibility. I can't comment on robustness because our extraction scheme isn't complete. However, even when a maximal number of bits are hidden the image is only slightly distorted. I'd say the artifact is a light fuzziness.

3 General Approach / Algorithm

The Spread Spectrum process contains two phases: encoding and decoding. Encoding is moslty straight forward. Starting with a shared key each bit has a corresponding psuedo-random long double generated on the domain from zero to one. If the bit is zero the normal stream is used, but if the bit is one the prime stream is used. Prime in this usage just means alternate. The prime stream is half of the range away from the generated bit but also within the domain. Meaning, if the default stream is below the middle point half the range is added whereas if it is above the middle point half the range is subtracted. These values are then fed into an inverse gaussian cumulative distribution function. This function is unbounded at both zero and one so the entire range is translated into a new domain clipped enough to allow reasonable values of the function through. Once it has gone through this function the stream is now in the form of a random value from a normal distribution. The Spread Spectrum decoding process is just a reverse of the encoding using a recovered cover image and the stego image to guess at the value of the bit at that location. The recovered cover image is generated by using an alpha-trimmed mean filter to correct the noise data inserted into the image. Additionally the message undergoes convolutional code to account for the imperfect nature of the alpha-trimmed mean filter combined with the decoding process around extreme values. The convolutional code once extracted undergoes a decoding using Viterbi decoding which produces the original image.

The program takes inputs of a picture a message, a seed for encoding, a picture an output file and a seed for decoding. The picture given to the decoding function should be the picture produced from the encoding function. The decoding function does not require the original picture to operate as long as the error correcting code is sufficient to deal with the errors produced by the alpha-trimmed mean filter and decoding.

4 Technical Difficulties

The viterbi portion of the error correcting code was far harder to implement than we thought it would be. We underestimated it and it took a lot of time to do. As a result we haven't quite implemented it, and consequently our extraction of the embedded message doesn't quite work. Fortunately, this stands as the only bug in our algorithm. Unless we get the decoding working and see that the encoding doesn't work, this remains the only problem in our stegosystem.

5 Future work

Get the extracting algorithm working and implement a higher rate error correcting code for better robustness.

6 Examples

These are in another document. The bee has the women in the robes (Barbara) hidden within it, Barbara has the baboon hidden within her, and lena has the car image hidden within her.

7 Statistical Analysis Results

With our current implementation, the capacity is half a bit per byte (since each message bit is convolutionally encoded to be two bits). This lands us a percentage of 6.25% capacity. The only time the technique because humanly perceivable is when a file much larger than the cover image is embedded, however without the cover image beside it to make a comparison the typical viewer wouldn't think anything was hidden at all.

References

[1] Marvel, L. M., C. R. Boncelet, and C. T. Retter. *Spread Spectrum Image Steganography.* *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society* 8.8 (1999): 1075-83.

[2] John Burkardt *NORMAL Normal Random Number Generators* *http://people.sc.fsu.edu/ jburkardt/c_ src/normal/normal.html* 14 August 2004

[3] Natarajan V., (Kanchipuram (near Madras), India) *Normal Inverse CDF* *http://home.online.no/ pjacklam/notes/invnorm/impl/natarajan/normsinv.h*