**CMSE492 Lab 4. RS Steganalysis**

29.04.2019

**Task (7 items):**

1. Implement RS Steganalysis algorithm [1] in any programming language/operating system available in CMPE-134
2. Test your implementation using Seminar 29.04.2019 examples: “Consıder two masks, 1x4 mask M1=(0,1,1,0), and 2x2 mask, M2=(1,0; 0, 1). Consider the grayscale image, I, below having 2 rows and 8 columns:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Col1 | Col2 | Col3 | Col4 | Col5 | Col6 | Col7 | Col8 |
| Row1 | 10 | 205 | 101 | 11 | 12 | 23 | 28 | 121 |
| Row2 | 255 | 235 | 193 | 221 | 207 | 121 | 170 | 211 |

Specify 4 pixel groups, G11,..,G14, for M1, and 4 pixel groups, G21, .., G24, for M2

Classify the groups as RM1, SM1, R-M1, S-M1, UM1, U-M1, and RM2, SM2, R-M2, S-M2, UM2, U-M2

Using 5% threshold, decide on the image, I, whether it has (not) a payload according to M1 and according to M2”

1. Test your implementation on 6 host 512x512 images from [1, 2]: Lena, Scene, Zelda, Peppers , Jet, and Mandrill-Baboon. Secret bit stream generate using a pseudo-random number generator (PRNG) provided in a programming language you use; set seed of the PRNG to some predefined value, e.g,, “345” so that each time you run the PRNG it will generate the same sequence of bits. Stream bit size shall vary from 0 to 100% of the host image pixel number according to [1, Figure 1; 2, Figure 8], with step 10%. **Use mask M1=(0,1,1,0) only in your implementation.**
2. **For each of the six host images calculate RM1, R-M1, SM1, S-M1, and RM2, R-M2, SM2, S-M2, for each embedding capacity from 0%, 10%, .., 100%, using 1-LSB method**.  **Draw 12 RS-diagrams (plots in the format of [1, Figure 1]). Make sure that for each host image one and the same secret bit stream is generated.**
3. **Using threshold, T=5%, and the RS-diagrams obtained in Item 4, stego-analyze the six images under consideration (decide whether they have/have not payload). Compare results of your stego-analysis for the masks M1 and M2.**
4. **Defend the Lab on May 15, 2019, Wednesday, 16.30-18.20, CMPE-134 (hand in your report to Evaluator, run your program, and explain your work done).**
5. Report shall have
	1. Cover page (University, Department, Course, Semester, Year, City, Country, Lab subject, Team members, Lecturer, Lab assistant)
	2. Outline
	3. Problem definition (see items 1-5 above)
	4. RS steganalysis method description
	5. Description of RS steganalysis method implementation in your programming language/operating system
		1. Description of the host images you use and their sources
		2. Description of the secret bit stream generation including how you make possible repetition of the same secret bit stream set generation for each host image and each mask used.
		3. Description of the secret bit stream generation of the required size: 0%, 10%,…,100%
		4. Description of RM, R-M, SM, S-M calculation (in percent)
		5. Description of testing on the seminar examples (see item 2 above).
		6. Description of an 1LSB embedding method
		7. Description of how you plot RS-diagrams
		8. Description of the results obtained (12 RS-diagrams)
		9. Results of steganalysis of the six host images using the 12 RS-diagrams (provide screenshots of the RS-diagrams).
	6. Conclusion
	7. References
	8. Appendices with the code developed
	9. CD with all Lab related materials (report, images used, test results, sources, executables). CD shall be runnable (it is possible to install your program from the CD, run it on your examples, and view results you got).

**References**

1. J. Fridrich, M. Goljan, and R. Du, Detecting LSB steganography in color and gray scale images, IEEE Multimedia, Oct.-Dec., 2001, 22-28.
2. D.-C. Wu, W.-H. Tsai, A steganographic method for images by pixel-value differencing, Pattern Recognition Letters 24 (2003) 1613–1626, <http://cmpe.emu.edu.tr/en/CourseLoad.aspx?id=CMSE492&page=lecturenotes>

**Grading policy: report – 50%, explanations – 50%**