CMSE492 Seminar on EMD 28.03.2019

1. Consider EMD method [1] with n=3
2. Consider the grayscale image, I, below having 2 rows and 8 columns:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Col1 | Col2 | Col3 | Col4 | Col5 | Col6 | Col7 | Col8 |
| Row1 | 0 | 100 | 105 | 10 | 21 | 31 | 255 | 121 |
| Row2 | 251 | 136 | 19 | 22 | 21 | 20 | 159 | 183 |

1. Let secret bit stream , bs, in hexadecimal is as follows: A0 B1 C2 D3 A1 B1 C1 D1 A2 B2 C2 D2 A B3 C3 D3
2. Convert the binary stream, s, into (2n+1)-ary digit stream, ds, using (1) from [1] assuming L=16.
3. Embed ds into I getting stego image, SI
4. Extract secret data from SI, and convert them into binary. Check that the data extracted match the data embedded
5. Your homework and participation in the seminar will be graded (50% +50%)

References

1. X. Zhang and S. Wang, Efficient Steganographic Embedding by Exploiting Modification Direction, IEEE COMMUNICATIONS LETTERS, VOL. 10, NO. 11, NOVEMBER 2006, 781-783, <https://staff.emu.edu.tr/alexanderchefranov/Documents/CMSE492/ZhangIEEECL2006.pdf>

Solution

-Arity is A=2\*n+1=2\*3+1=7

The number K of 7-ary digits shall be minimal such that $(2n+1)^{K}=7^{K}\geq 2^{L}$, and $K\geq \frac{L}{log\_{2}7}=\frac{16}{log\_{2}7}=5.7$, hence, K=6

The first block of L=16 bits is a block of 4 hexadecimal digits, b1= A0 B116 = 10\*16^3+0\*16^2+11\*16+1 = 40960+176+1 = 4113710

Let’s convert decimal b1=41137 to K=6 7-ary digits dividing it consecutively by 7, using quotient for next didvision, and recording residues

|  |  |  |  |
| --- | --- | --- | --- |
| Dividend | remainder | quotient | Divisor |
| 41137=7\*5876+5 | 5 | 5876 | 7 |
| 5876=7\*839+3 | 3 | 839 |  |
| 839=7\*119+6 | 6 | 119 |  |
| 119 | 0 | 17 |  |
| 17 | 3 | 2 |  |
| 2 | 2 | 0 |  |

Hence,

b1=2306357=2\*7^5+3\*7^4+0\*7^3+6\*7^2+3\*7+5=33614+7203+0+294+21+5=41137 (1)

To embed 6 digits, we need 6 groups of 6 pixels, i.e. 18 pixels. We are given 16 pixels, hence, only 5 first digits of b1, 23063, can be embedded

1. Embed d=2 into (p1, p2, p3)=(0,100,105)

We use extraction function

f(0,100,105)=0+100\*2+105\*3 mod 7= 0+2\*2+0\*3 mod 7 = 4 mod 7 = 4=f

s=d-f=2-4 =-2 mod 7 = 5>n=3, hence pixel -5 mod 7 = 2 is decreased by 1 getting

(p1’, p2’, p3’)=(0,99,105)

Extraction: f(0,99,105)=0+99\*2+105\*3 mod 7 = 0+1\*2+0\*3=2 mod 7 = 2=d, OK

1. Embed d=3 into (p4, p5, p6)=(10,21,31)

We use extraction function

f(10,21,31)=10+21\*2+31\*3 mod 7= 3+0\*2+3\*3 mod 7 = 12 mod 7 = 5=f

s=d-f=3-5 =-2 mod 7 = 5>n=3, hence pixel -5 mod 7 = 2 is decreased by 1 getting

(p4’, p5’, p6’)=(10,20,31)

Extraction: f(10,20,31)=10+20\*2+31\*3 mod 7 = 3+6\*2+3\*3=24 mod 7 = 3=d, OK

1. Embed d=0 into (p7, p8, p9)=(255,121,251)

We use extraction function

f(255,121,251)=255+121\*2+251\*3 mod 7= 3+2\*2+6\*3 mod 7 = 25 mod 7 = 4=f

s=d-f=0-4 =-4 mod 7 = 3<=n=3, hence pixel 3 is increased by 1 getting

(p7’, p8’, p9’)=(255,121,252)

Extraction: f(255,121,252)=255+121\*2+252\*3 mod 7 = 3+6\*2+3\*3=24 mod 7 = 3=d, OK

1. Embed d=6 into (p10, p11, p12)=(136,19,22)

We use extraction function

f(136,19,22)=136+19\*2+22\*3 mod 7= 3+5\*2+1\*3 mod 7 = 16 mod 7 = 2=f

s=d-f=6-2 =4 mod 7 = 2>n=3, hence pixel -4 mod 7 =3 is decreased by 1 getting

(p10’, p11’, p12’)=(136,19,21)

Extraction: f(136,20,22)=136+19\*2+21\*3 mod 7 = 3+5\*2+0\*3=13 mod 7 = 6=d, OK

1. Embed d=3 into (p13, p14, p15)=(21,20,159)

We use extraction function

f(21,20,159)=21+20\*2+159\*3 mod 7= 0+6\*2+5\*3 mod 7 = 27 mod 7 = 6=f

s=d-f=3-6 =-3 mod 7 = 4>n=3, hence pixel -4 mod 7 =3 is decreased by 1 getting

(p13’, p14’, p15’)=( 21,20,158)

Extraction: f(21,20,158)=21+20\*2+158\*3 mod 7 = 0+6\*2+4\*3=24 mod 7 = 3=d, OK

So, thus, the data extracted are 23063 in 7-ary system. Since the number of digits embedded and extracted is less than L=16 used for conversion the binary stream into 7-ary stream, we can’t convert back these digits to binary. If there were 2 more pixels, then the last digit 5 could be embedded and extracted. Then, with K=6 7-ary digits, we can convert them to decimal as in (1), getting 41137. Then, we convert this decimal to hexadecimal, and, representing each hexadecimal as 4 bits, we come to the respective binary stream. Convert 41137 to 16-ary number:

|  |  |  |  |
| --- | --- | --- | --- |
| Dividend | remainder | quotient | Divisor |
| 41137=16\*2571+1 | 1 | 2571 | 16 |
| 2571=16\*160+11 | 11 =B | 160 |  |
| 160 | 0 | 10 |  |
| 10 | 10 =A | 0 |  |

Thus, 4113710=A0B116= 1010 0000 1011 00012