


Data Security

steganography

Samar Shaaban Abdelfattah



Using Disinformation To Promote The Security Of SMMWB Image Steganography Method

Embedding

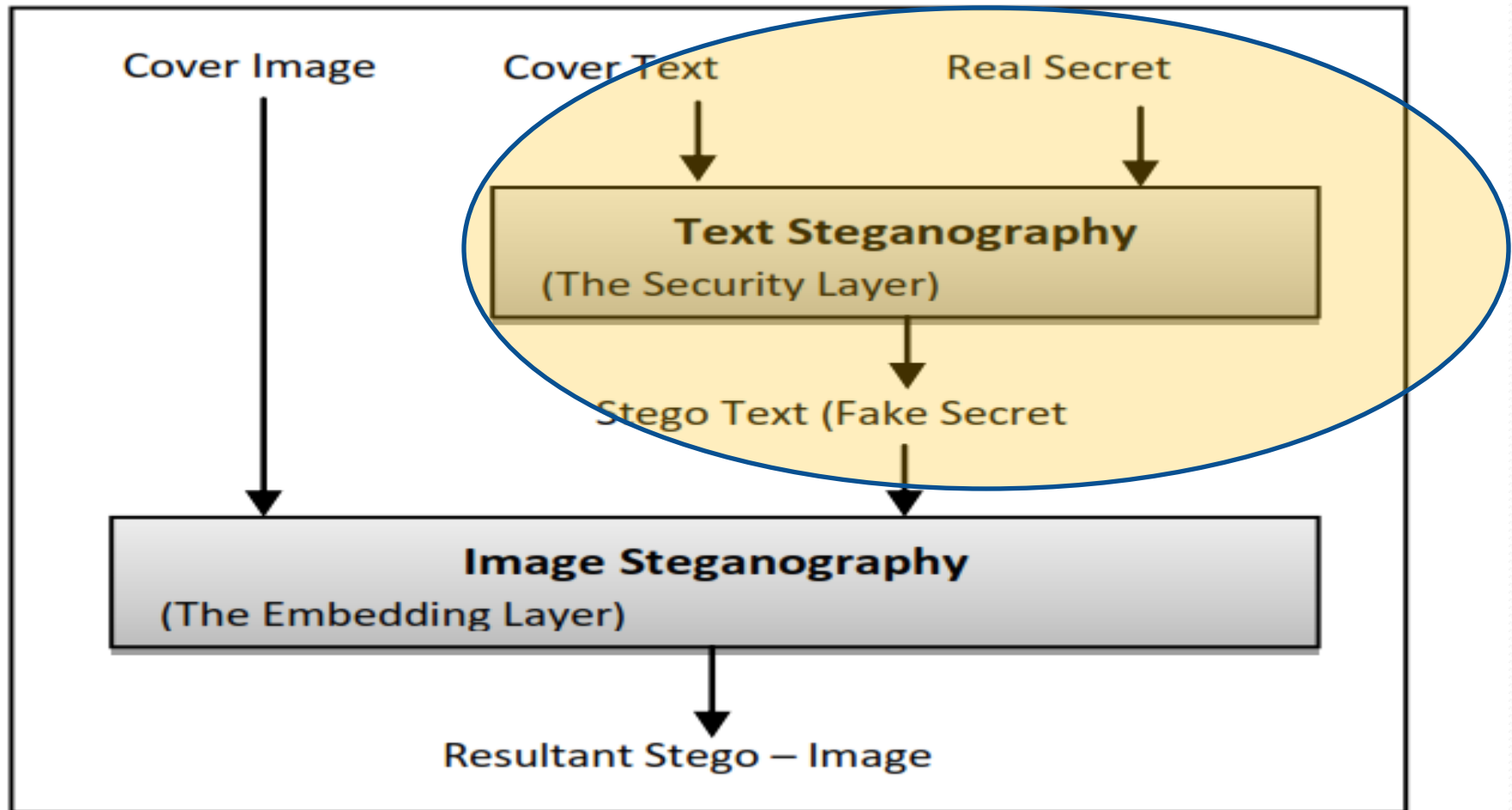


Fig 2: Sender Side

First Phase:

Text Steganography Algorithm

Algorithm: Embedding Using Mixed-Case Font Method

Input : Text File T ; Secret Message M.

Output : Stego Text S.

Steps:-

- 1) Choose a text file T.
- 2) Divide T into letters, $T = \{T_1, T_2, \dots, T_n\}$.
- 3) Get the secret message M.
- 4) Convert secret message M into stream of bits b.
- 5) Divide b into bits, $b = \{b_1, b_2, \dots, b_n\}$.
- 6) Select T_i from T and b_i from b.
- 7) If the b_i is 'one' then change T_i case into capital else change T_i case into small.
- 8) Repeat steps 6,7 till the whole b is hidden.
- 9) The resultant file will be the stego text S.

Screenshot Of My Run First Phase

```
In [3]: plaintext='Password' #secret message  
text_stegnog(plaintext)
```

The string after binary conversion : 101000011000011111001111100111110111110111110100101000100
56

```
In [4]: fake_Mess=text_desteg()  
print("Message After Uncovering File: ",fake_Mess)
```

Message After Uncovering File: Password

Embedding

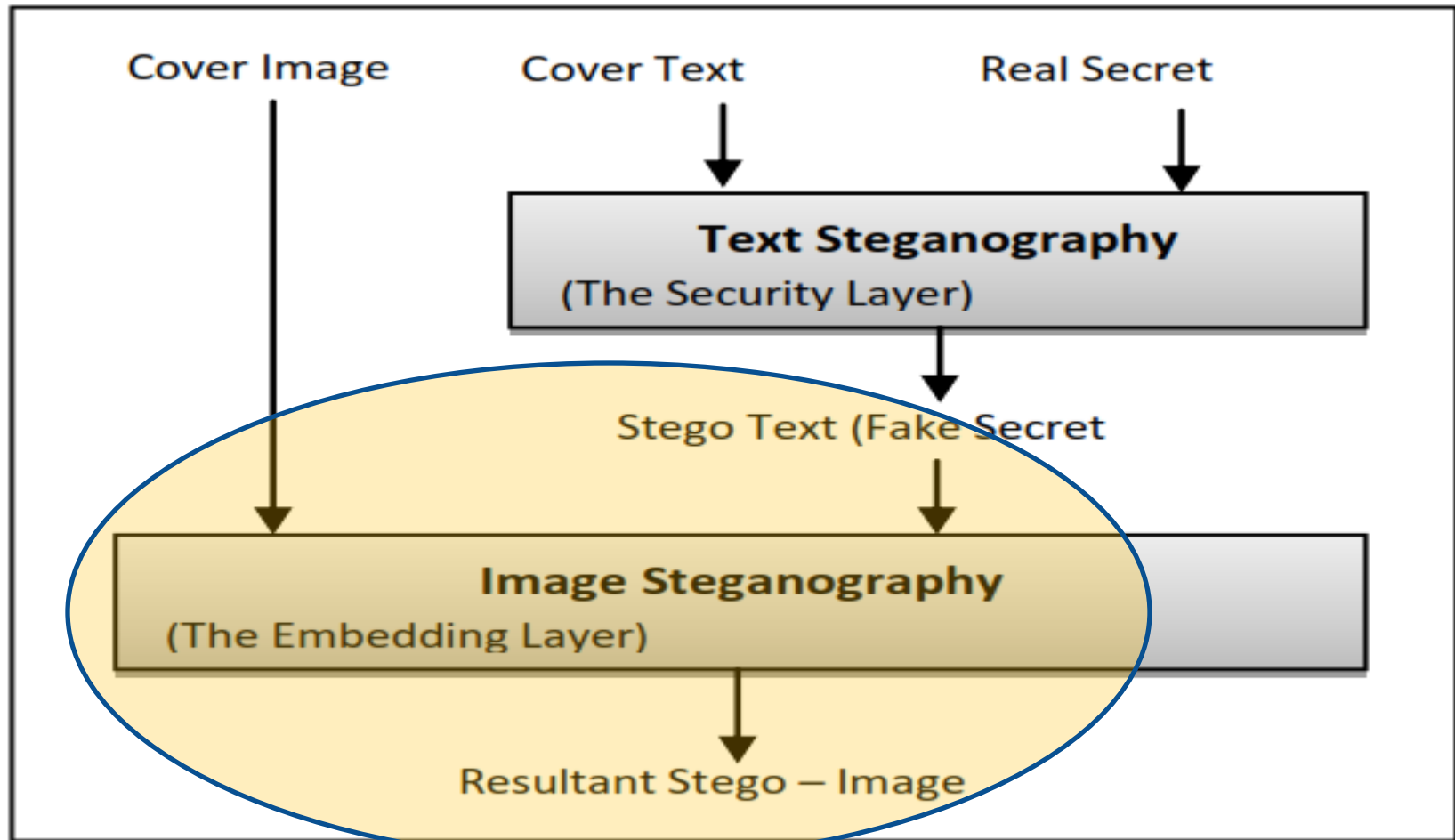


Fig 2: Sender Side

Second Phase:

Image Steganography Algorithm Using LSBraile

5.1 **LSBraile Embedding Algorithm: Message Embedding Using LSBraile Method**

Input : Cover Image C ; Secret Message M.

Output : StegoImage S.

Steps:

- 1) Split C into 3 channels Red (R), Green (G), Blue (B).
- 2) Split M into characters; $M = \{ m_1, m_2, m_3, \dots, m_n \}$.
- 3) Take m_i from M
- 4) Convert m_i into Braille 6 – bits representation.
- 5) Take 6 pixels from B.
- 6) Apply LSB on m_i 's 6 – bits and the 6 pixels of B.
- 7) Repeat steps from 3 to 6 until the whole M has been embedded in C.
- 8) Merge the 3 channels R, G, B again to construct the StegoImage S.

Braille Vs Traditional LSB

- Consider an 8-bit bitmap image where each pixel is stored as a byte representing a grayscale value.
- Suppose the first eight pixels of the original image have the following gray scale values

11010010 01001010 10010111 10001100 00010101 01010111 00100110 01000011

- To hide the letter C whose binary value is 10000011, we would replace the LSBs of these pixels to have the following new gray scale values:

11010011 01001010 10010110 10001100 00010100 01010110 00100111 01000011

- using Braille method of reading and writing for blind people.
- Braille system uses six raised dots in a systematic arrangement with two columns of three dots.

Decryption

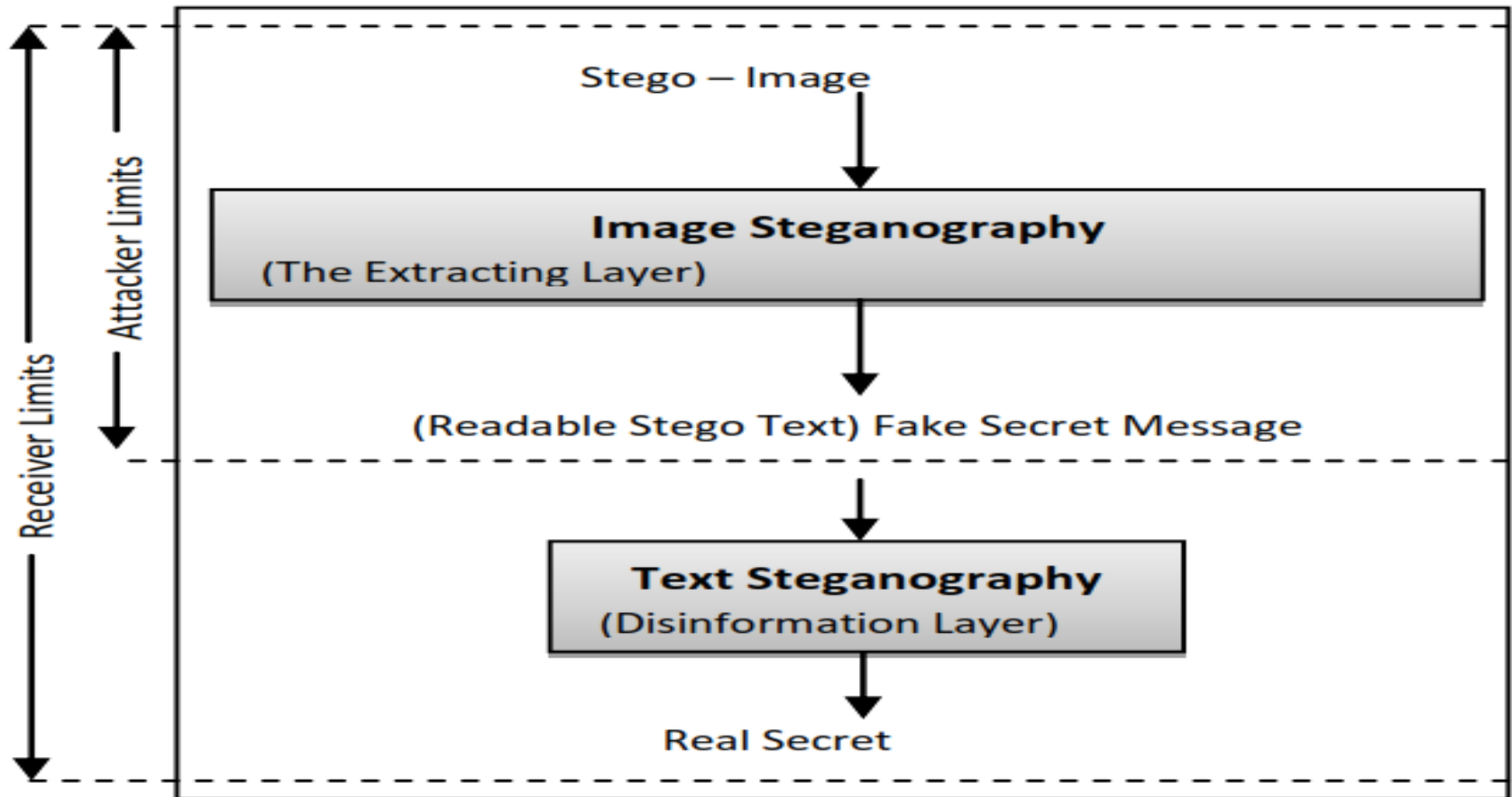


Fig 3: Attacker and Receiver Sides

Complete Run of the system

```
In [7]: secure_msg='passwordSamar'
cover_text='test.txt'
cover_img = Image.open('flag fayoum.jpg')

encrypted_image=image_stegnog(cover_img,cover_text,text_stegnog(secure_msg))
```

Encryption Phase

The string after binary conversion : 111000011000011110011111001111101111101111110010110010010100111100001110110111000011110010

```
In [8]: enc_img = Image.open('encrypted_image.png')
fake_message=image_desteg(enc_img)
```

The hidden message is :

YOUdonoTDoyouDONOtDOANYMoreBLACKSHOEInWHICHIHavELIvedLikEaFootFORTHinTYYEARSFOOrandWITteba

Decryption Phase

```
In [9]: print('the secret message is:', text_desteg(fake_message))

the secret message is: passwordSamar
```

```
In [11]: cover_img.close()
enc_img.close()
```