Steganography Secure Communication

Security cross-cutting issues

Submitted by
Lidong Wei, Preethi Josephina Mudialba, Siddharth Nair

Abstract

58 journalist lost their lives in 2016 alone while 117 journalists are currently imprisoned [1]

With more than 32,000 people killed in 2015 from terrorism attacks around the world, more and more journalists, media advocates and netizens are being assigned to dangerous regions. Organizations like Reporters without Borders and the United Nations are fighting every day for freedom of information and human rights. But the dearth of secure modes of communication exposes these undercover journalists to unnecessary risks. [2]

The aim of our project was to develop an application which will help journalists assigned to dangerous countries, communicate or carry back sensitive information (example: images of a suspected terrorist), across borders without compromising their safety. For the purpose of the project we analysed several steganography and we finally implemented 2 algorithms namely:


We then carried out StegAnalysis on the image obtained from each of these algorithms. The results revealed the algorithm (a) to be the better out of the 2, which we implemented in our final application.

The steganography and encryption algorithms were deployed by using MATLAB and the testing and implementation was carried out using several software application.

The novelty of the application arises from integration of prevailing technologies with innovative new models for secure communication. While there exist several steganography algorithms, the Mosaic based algorithm implemented in our application is less prone to attacks as it neither falls under the spatial domain nor the transform domain, thereby making it harder for an attacker, to detect the presence of steganography. This project demonstrates the use of steganography and encryption as a means of securing communication.
Methodology

Steganography: Secret-Fragment-Visible Mosaic Images

Given two images, secret image and cover image respectively, this algorithm is to hide secret image into cover image to form the so-called secret-fragment-visible mosaic image. There are 4 major processes in the hiding phase, split, sort, colour transfer and reconstruct. The detailed algorithm is described below.

Hiding Phase

**Input:** Secret Image, Cover Image, splitting size  
**Output:** Steg Image

- **Step 1.** Split the two images, secret image S and cover image C, according to the splitting size to form two sets of tile images $S = \{S_1, S_2, ..., S_n\}$ and $C = \{C_1, C_2, ..., C_n\}$, each tile image in the two set is of the splitting size.
- **Step 2.** Compute the average colour value and standard deviation of each tile images, $S_i$ and $C_j$ respectively, in the two sets for RGB three colour channels. Sort the two sets $S$ and $C$ based on the average green value in ascending order. Map each tile image $S_i$ in sorted $S$ to one tile image $C_{ji}$ in sorted $C$ in a 1-on-1 manner.
- **Step 3.** Perform colour transfer according to the equation:
  \[c'' = qc(c-c') + c'\]
  Where $qc$ is the quotient of standard deviation and $c=r, g, or b$. $c$ is the average value of a specific tile image $S_i$ in $S$ and $c'$ is the average value of a specific tile image $C_i$ in $C$.
- **Step 4.** Fit each transferred tile image $S_i$ in $S$ into the index of its corresponding tile image $C_{ji}$ in the original cover image to form mosaic image.
- **Step 5.** Concatenate the related information in binary format of each tile image $S_i$ in $S$ to form a recovery bit stream. The recovery bit stream Using two pixels of the mosaic image as one embedding unit to embed the recovery bit stream. First pixel is used as flag bit and second pixel is used as available bit to embed information. Embed the recovery bit stream into mosaic image using Reversible Contrast Mapping[14] to form the steg image.

Recovery Phase

**Input:** Steg Image  
**Output:** Secret Image

- **Step 1.** Extract information from the steg image using the inverse version of Reversible Contrast Mapping. Convert the binary values to decimal values for each tile image.
- **Step 2.** Using the extracted means and standard deviations of each tile image perform inverse colour transfer based on the equation:
\[ \text{ci} = \frac{1}{qc}(\text{ci''-c'}) + c \]

- Step 3. Using the extracted index to retrieve the original position in the secret image of each tile image.
- Step 4. Reconstruct the image according to the index to get the secret image.

**Encryption**

As an added layer of security, we experimented with encrypting the Secret Image to be hidden. This would make it almost impossible for the attacker to recover the original image, because even if they were able to correctly identify the Steg image containing the hidden information, they would not be able to recover the original image by just breaking the Steg, because all they would get is the encrypted image. To show this proof of concept, we integrated a standard image encryption technique based on key generation technique along with our existing steganography algorithm code [5].

The downside to this approach is that as the steganography algorithm is not completely lossless, when the recovered encrypted image is decrypted, there is some amount of colour loss, although the general structure can still be identified. The information recovered is sufficient for Facial recognition algorithms to identify the individual in the photo. Thus, the current implementation can still be successfully used by journalist, military personal etc to transfer information about suspected, terrorists, especially across borders. However, in the future the data lost can be minimized to obtain a clear recovered image.

**References**


