# STEGANALYSIS OF IMAGES USING QRC

**Description:**

The Steganalysis of images using QRC is used to handle the security of the user information and it provides the ease of access for the files. This uses cryptography and steganography in combined form in order to encrypt the user file and hide them behind an image.

System gathers the files & key from the user that are to be encrypted and secured.

**Technology details:**

C#, .NET IDE: Microsoft Visual Studio.

**Existing system:**

* The existing system of steganography can be used only to encrypt one type of file and it cannot accept all other types.
* The text alone can be hidden inside the image which may provide less security and privacy.
* The steganalysis can be a useful and advanced technique which might prevent intrusion or hacking.

**Proposed system:**

* The steganalysis technique can be implemented to encrypt any form of file and hide its key behind an image.
* This image can be converted to QR Code which can be decrypted by any user-friendly device like desktop, Mobile, scanners etc..,
* It can support on any operating system like Windows, Mac OS, blackberry, android etc.,

**Algorithm used:**

**RIJNDAEL ALGORITHM**

The Rijndael algorithm is a new generation symmetric block cipher that supports key sizes of 128, 192 and 256 bits, with data handled in 128-bit blocks - however, in excess of AES design criteria, the block sizes can mirror those of the keys. Rijndael uses a variable number of rounds, depending on key/block sizes, as follows:

* 9 rounds if the key/block size is 128 bits
* 11 rounds if the key/block size is 192 bits
* 13 rounds if the key/block size is 256 bits

Rijndael is a substitution linear transformation cipher, not requiring a Feistel network. It uses triple discreet invertible uniform transformations (layers). Specifically, these are: Linear Mix Transform; Non-linear Transform and Key Addition Transform. Even before the first round, a simple key addition layer is performed, which adds to security. Thereafter, there are Nr-1 rounds and then the final round. The transformations form a State when started but before completion of the entire process.

The State can be thought of as an array, structured with 4 rows and the column number being the block length divided by bit length (for example, divided by 32). The cipher key similarly is an array with 4 rows, but the key length divided by 32 to give the number of columns. The blocks can be interpreted as uni-dimensional arrays of 4-byte vectors.

The exact transformations occur as follows: the byte sub transformation is nonlinear and operates on each of the State bytes independently - the invertible S-box (substitution table) is made up of 2 transformations. The shift row transformation sees the State shifted over variable offsets. The shift offset values are dependent on the block length of the State. The mix column transformation sees the State columns take on polynomial characteristics over a Galois Field values (28), multiplied x4 + 1 (modulo) with a fixed polynomial. Finally, the round key transform is XOR ed to the State. The key schedule helps the cipher key determine the round keys through key expansion and round selection.

Overall, the structure of Rijndael displays a high degree of modular design, which should make modification to counter any attack developed in the future much simpler than with past algorithm designs.

**Input:**

* The input will be the files which the user desires to encrypt & the image behind which the files are to be encrypted.
* The password consisting of 8 characters must be given as a key to perform encryption.
* The generated QR must be saved in the form of a png image.

**Output:**

* The output will be the QR Code which consists of the encrypted image (with file) and the URL behind it.
* This QR code can be scanned to view the URL of the image & the URL can be redirected to the original image.
* The decryption can be carried out by specifying the path of the encrypted file and the QR (png).

**Process involved:**

**Encryption & QR generation:**

1. The steganalysis application is opened.
2. The admin alone can login by specifying the username and password.
3. On the top left corner the admin can select the “encryption” option.
4. The path of the file to be encrypted is specified in the “input file” text box.
5. The image behind which the key of the encrypted file is to be hidden is given by clicking the “select image” button.
6. The key consisting of 8 characters is specified in the form of password.
7. The “output location” textbox specifies the location of the file.
8. The “encryption” button is clicked to encrypt the file.
9. On clicking the button the QR is automatically generated for the specified image.
10. It prompts the user to save the QR image in png format with any name.
11. The above QR can be scanned to view the link of the encrypted image.

**Decryption:**

1. The decryption can be done only with the encrypted form of the file.
2. The path of the encrypted file is specified in the “input file” text box.
3. The QR which is already saved during the encryption is inserted by clicking the “open image” button.
4. The URL contained by the QR can be retrieved by clicking the “get URL” button.
5. The retrieved URL will be automatically generated in the “Image URL” text box.
6. The “Get Image from URL” button can be clicked to view the original image.
7. The “decryption” button is then clicked to decrypt the original file.

**Note**:

To check whether the encryption is done properly, the user can open and check the encrypted file before decryption. When the user attempts to open the encrypted file, they will not be able to open it. The file can be opened only after decryption.