1. Introduction:

Steganography is the art of hiding and transmitting data through apparently innocent carriers in an effort to conceal the existence of the secret data. The provided tool JPEG FILE HIDER is a Jpeg-based steganography tool which can be used to hide a secret file within Jpeg image. Several of image-based steganography tools are available on the internet but most of them don't provide the required robustness against detection which could be performed using steganalysis tools. In the Magic Triangle shown in Fig1, we can see the contradiction between requirements in the field of information hiding, regarding steganography there are two main Conflicting requirements, insertion capacity and undetectability, it is not possible to attain high undetectability and high insertion capacity at the same time, there is a trade-off between the capacity available to hide data and the robustness against detection. Compared with other image-based steganography tools, JPEG FILE HIDER doesn't provide a high insertion capacity, but it provides a high level of robustness against detection.

JPEG FILE HIDER applies a non typical DCT-based algorithm, the applied algorithm doesn't provide a high insertion capacity but it has a high resistance against statistic attacks, the secret data will be hidden within the DCT coefficients domain using an imperceptible way. Secure steganographic algorithms should not cause any perceptible distortion and have to attain statistical invisibility as well. It should be very difficult to prove the presence of hidden data, in other words the presence of steganography should be undetectable.

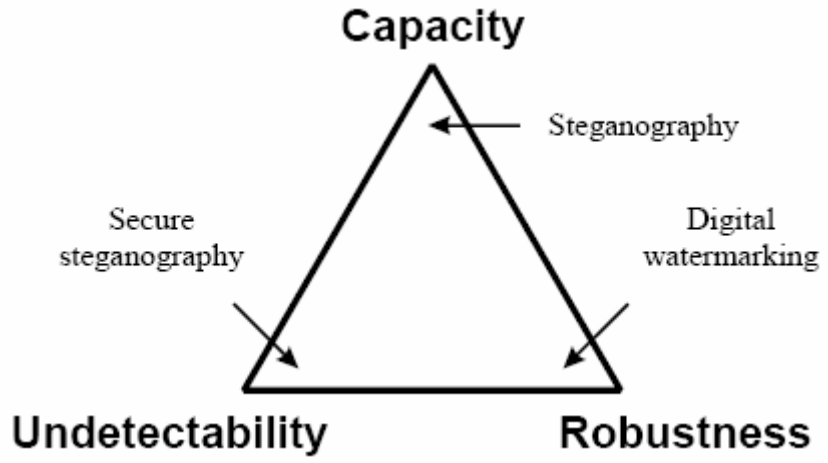

Figure (1): Conflicting requirements**.**

JPEG FILE HIDER is a Matlab-based application, it is built in Matlab development environment; therefore an environment called Matlab Compiler Runtime (MCR) needs to be installed prior to execute JPEG FILE HIDER EXE file. An MCR installer is available and it guides the user through the installation of the MCR.

2. Files contained in the provided packages (JFHMCRv1200_pkg and JFHv1200_pkg):

- JPEGFileHider.exe
- MCRInstaller.exe for MCR version 7.17, (Not contained in package JFHv1200_pkg)
- Readme PDF file
- Readme TXT file

3. Installation and execution:

The Matlab Compiler Runtime (MCR) version 7.17 should be installed prior to run JPEG FILE HIDER program. If the user doesn't have MCR installed on his system, the execution of the self extracting EXE package JFHMCRv1200_pkg will guide the user first through the installation of the MCR then extracting JPEG FILE HIDER and Readme files. Once the MCR is installed, the user can run JPEG FILE HIDER by double clicking its icon directly as a portable executable file.

NOTE: User will need administrator rights to run MCRInstaller.

4. How JPEG FILE HIDER works:

When the MCR has been installed on the system, the user can run JPEG FILE HIDER directly by double clicking its icon as a portable executable file. Only the starting execution of JPEG FILE HIDER takes a long time because it has to unpack all libraries from the archive, after that subsequent runs launch faster, therefore once the starting GUI is shown the rest of runs and functions will run faster. The first GUI, as shown in Fig2, welcomes the user and gives him the ability to choose either the hiding section or the extracting section. As a channel communication model, the user in the sender side chooses the hiding section GUI to hide a secret file within a Jpeg file, while in the receiver side the user chooses the extracting section GUI to extract the hidden sent file from the received image. It is not only restricted to communicate with others; the user can use it on his system to keep private and secret information away from intruders.

In the hiding section GUI, as shown in Fig3, the user can select the Jpeg file in which the user wants to hide a secret file; JPEG FILE HIDER will test the selected Jpeg file in order to accept or reject the selected Jpeg, if accepted the available maximum container capacity will be shown. The acceptance of a Jpeg file relates to the color components of the selected image and to quantized DCT coefficients. The user can select any secret file, whatever its format, to be hidden within the selected Jpeg file, but a comparison between the maximum hiding capacity available and the secret file size will be performed to decide if the hiding operation could be accomplished or not. During the hiding process and after naming and locating the produced steganographed Jpeg file, the user will be asked to provide a key which must be at lest 12 characters, JPEG FILE HIDER uses the key to locate the corresponding DCT locations in which the secret data will be hidden. The entered key is not saved or even check summed within the produced steganographed Jpeg file, therefore if the used key is lost the hidden file will never be extracted.

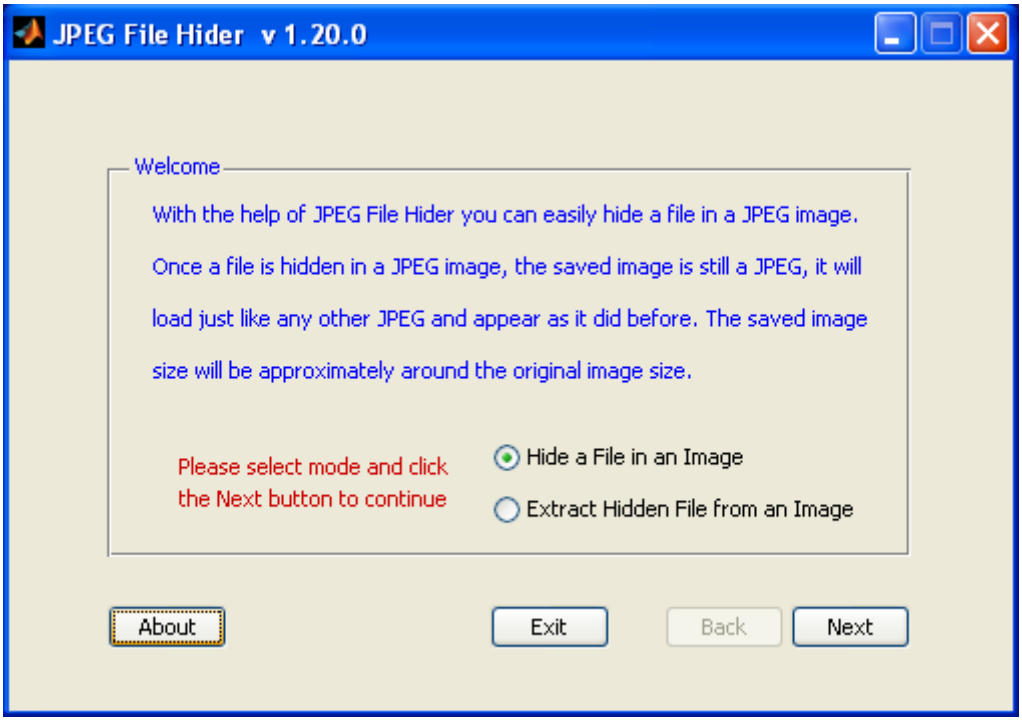

Figure (2): Starting GUI.

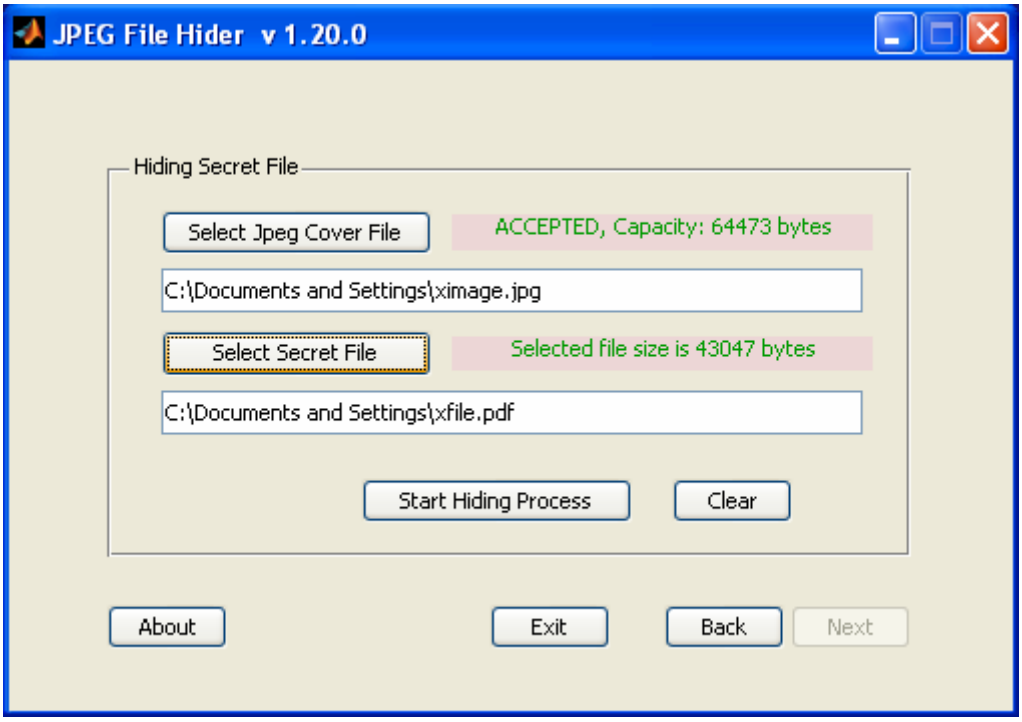

Figure (3): Hiding Section GUI.

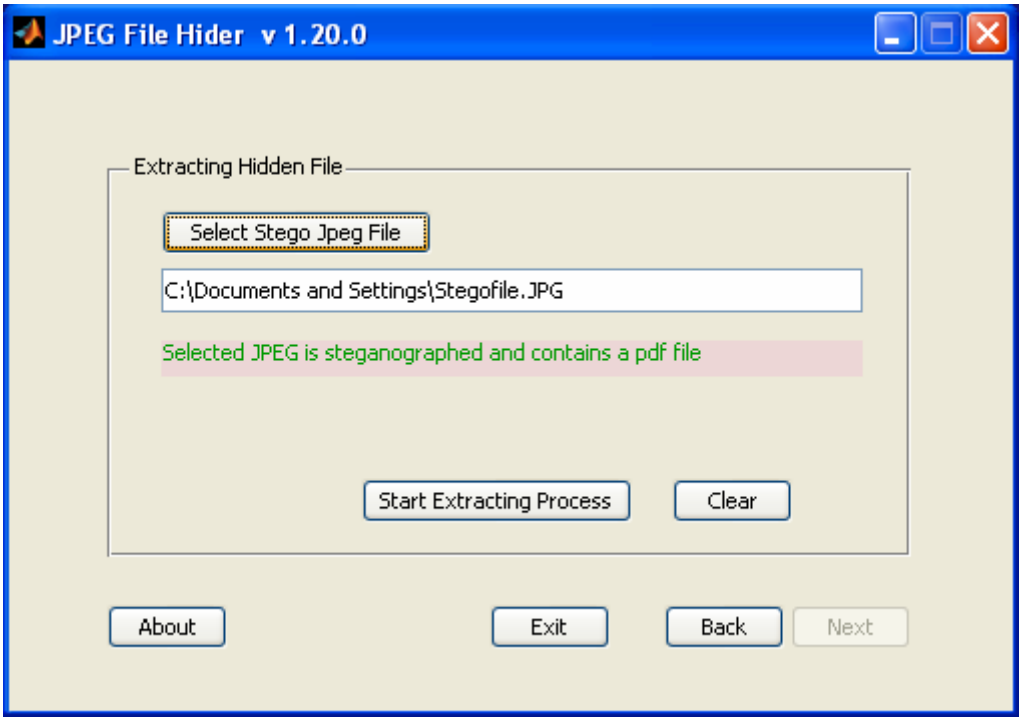

Figure (4): Extracting Section GUI.

In the extracting section GUI, as shown above in Fig4, the user can select the steganographed Jpeg file. The selected Jpeg file will be tested to decide if it has been steganographed by JPEG FILE HIDER, if steganographed, JPEG FILE HIDER will shows the extension of embedded secret file. During the extraction process, the user starts by entering the key, then the user will be asked to name and locate the extracted hidden file. IT IS VERY IMPORTANT TO NOTICE THAT the extraction process is based on the entered key; therefore if the entered key doesn't match the key which was used during hiding process, the extracted file WILL BE UNREADABLE. The key which was used during hiding process IS NOT SAVED OR EVEN CHECK SUMMED in a secret place inside Jpeg file, therefore if the key is lost, the hidden file will never be extracted.

5. Notices:

Actual version of JPEG FILE HIDER doesn't encrypt the selected secret file before embedding it within the Jpeg file. Adding encryption layer in addition to steganography layer will provide an absolute security; therefore XDATASecurity considers adding encryption in addition to steganography in a future version of JPEG FILE HIDER. Actual version of JPEG FILE HIDER is experimental and free and could be redistributed without permission from XDATASecurity. XDATASecurity team hopes that users provide him with any problems or bugs they encounter during their use of JPEG FILE HIDER or any suggestions they consider.

