Introduction
Analogue video from Israeli UAVs has been intercepted in both clear (i.e. unencrypted) and scrambled (i.e. encrypted) formats. Processing clear video using M2Extra is relatively straightforward, the method being described in Anarchist training Module 4. For scrambled video the capability exists to exploit the content using a combination of image processing tools and scripts on Mutiny Jaguar.

Background
Interception of scrambled analogue video signals at Anarchist has a long history, with the earliest examples dating back to 1998.

The Signal
The appearance of the encrypted signal in the frequency domain is virtually indistinguishable from the clear video signal. A comparison of the Post-D data for an example of clear video, and an encrypted example from a few seconds later (figs 1a & 1b) show that, apart from a subtle modification to the envelope, the signals appear very similar. The most noticeable effect is an increase in energy at lower frequencies, consistent with the detail in the image being smoothed out by the scrambling process.

Scrambled Imagery
As can be seen by examining an example of a frame of scrambled video (Fig.2) the video frame is unchanged by the scrambling method. In addition to the image seen in clear video there is also two lines of digital information encoded in the teletext area at the top of the screen. This is presumed to be information relating to the scrambling, e.g. a cryptographic 'key' to enable the original image to be reconstructed.

Investigation of the data has determined the method by which the video is scrambled. The method used is a ‘cut & slide’ technique whereby each line is cut at a location and the two halves are transmitted in the opposite order. This technique was originally used by Sky TV to protect their analogue transmissions before they switched to digital, the system being known as VideoCrypt.
Exploiting Scrambled Video Images

Having determined the technique used to scramble the video imagery a number of known attacks are available from open source material. One technique in particular offers a brute force way of reconstructing the image, without requiring any knowledge of the generating algorithm. This technique, and the source code needed to employ it is freely available on the internet.

The computing power needed to descramble the images in near real time is considerable without the use of dedicated hardware such as a video capture card that can record uncompressed images. It is still possible to descramble individual frames to determine the image content without too much effort.

Method

The method involves capturing a video frame in bitmap (.BMP) format using M2Extra. The video data should be processed as described in the M2Extra video processing guide. When the quality of the video image is good a snapshot can be made of the data in the Event Processing window. Pause the processing and use the left mouse button to zoom in to the scrambled image to exclude the frame.

From the file menu in the top left hand corner of the Event Processing window select Snap . . . (CTRL-S), and choose .BMP as the format.

Start Martes in a terminal window with the command martes, and launch the Image Magick tool from a terminal window with the command magick_display.
Select the bitmap image from the file menu, right click on the image and select Save ... The image format can be set by pressing the Format button at the bottom of the window. There are a huge number of different formats to select from. The format required is PPM format (portable pixmap).

In a terminal window at a command line prompt type

\texttt{antisky -bc input.ppm output.ppm}

This will now have descrambled the image using the program \texttt{antisky}. The descrambled image can be viewed using the \texttt{Image Magick} tool and converted to a more convenient format if desired.

The initial results from running antisky with the default settings as above may not produce particularly good results depending on the image being descrambled. This is because there may be part of the non-scrambled frame of the image included in the descrambling which corrupts the results. To improve the descrambling there are two more option which force the program to ignore the left and right hand sides of the image. Using the \texttt{-l} and \texttt{-r} (for left and right) flags and experimenting with different values may produce better results. The descrambled image first obtained with the default settings is shown in Fig. 4, whilst the image obtained with the command

\texttt{antisky -bc -l15 -r3 input.ppm output.ppm}

is shown in Fig. 5 and is considerably clearer.

There is no quick way of discovering the optimum settings for Antisky other than stepping through the parameter space of values and selecting the one that gives the best results.

As can be seen from Figs 4 & 5, for a good quality signal and optimal settings near perfect image construction can be achieved.