AIRBORNE SURVEILLANCE SYSTEM. IMAGE ENCRYPTION MODULE

Ciprian RĂCUCIU*, **Nicolae JULA***, **Dan GRECU***, **Marian PEARSICĂ****

*Military Technical Academy, Bucharest, Romania
**"Henri Coandă" Air Force Academy, Brasov

***Abstract**: This paper presents an encryption system developed to be used on an airborne surveillance system. The encryption system is composed from six modules: image capture, encryption, two radio link modules, decryption and display module. The first three modules are used on airborne hardware platform and the last three modules are used on the base station platform. The airborne hardware platform is the CV700C motherboard and the base station is an Intel microprocessor based notebook. For encryption, Rijndael algorithm was used.*

***Keywords**: gear noises, airborne system, encryption module, original image, standard deviation.*

1. INTRODUCTION

An UAV[1] is an airborne system, without a pilot, witch flies by means of a remote control or by using an autopilot installed on board and carries sensors or weapons. It can be used only once or reused many times. Comparing an UAV with a classical airplane, most of the times, the UAV is small and light and the load is composed from sensors used in reconnaissance and surveillance missions, or in target acquisition missions. Now, there are new mission for UAV, like combat, intelligence, and civil applications. Because any transmission from a military UAV must be secured it's a must to use an encryption method to protect de confidentiality of the transmitted data.

Because of the restrictions applied to weight, dimensions and power consumption of any UAV module, it was chosen the CV700C motherboard manufactured by Lex Inc. witch is 200 mm long and 150 mm wide. For an increased endurance to vibrations, the hard disk was replaced with a Compact Flash memory card. To maximize the encryption speed and to reduce the footprint the Windows XP Embedded operating system was used. The program was written in C++ programming language using the DirectShow API[2] witch has the whole support for buffering and frame dropping.

2. THE SYSTEM

The system is composed of:

- Digital video camera with USB interface;
- CV700C motherboard;
- RF modules;
- Notebook.

Table 1 Weight and price of used components

Device	Weight [g]	Price [€]
CV700C	500g	320€
Gigabyte 802.11b/g	15g	20€
Canion video	20g	10€

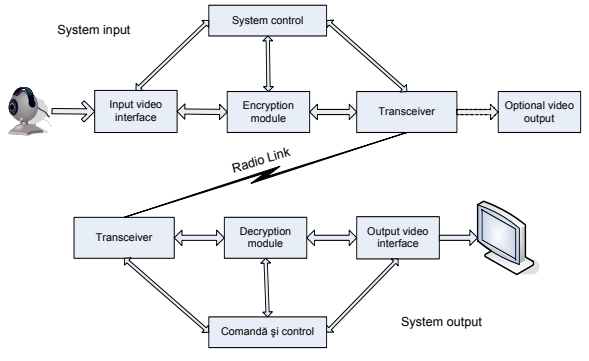

Fig. 1 The system

]

The motherboard is based on VIA C7 microprocessor witch operates at a frequency of

[1] Unmanned Aerial Vehicle
[2] Application Interface

1GHz. The microprocessor is built for applications with low power consumption requirements. On full load, the consumption of the microprocessor is 11 W and of the entire board is 25 W. To the motherboard are connected a video camera witch takes pictures at a rate of 25 frames/second and a 802.11 wireless radio module witch operates in the public spectrum. On this motherboard runs the airborne encryption subsystem.

The other component of the system is the base station witch runs on an Intel based microprocessor notebook. It receives data from the integrated 802.11 wireless module and feeds the data to the decryption module and displays the decrypted images on the monitor. The airborne encryption module can be controlled via radio link and there can be changed the encryption keys and the encryption modes.

2.1. DIRECTSHOW API

To benefit from the full power of Windows operating system, the DirectShow framework was used. Because DirectShow is a filter based framework, there were developed several filters:

- Encryption and decryption filter;
- RF interface modules (client and server).

The video input and output filters are contained in the DirectShow API. The main program initialises the filters and controls different parameters like:

- Image resolution;
- Bit depth;
- Encryption mode;
- Encryption key.

In this case, a resolution of 320x240 pixels and a bit depth of 24 bits/pixel were used. So the entire image contained 230400 bytes without the header used to pass the image parameters between filters, witch added extra data.

2.2. THE ENCRYPTION MODULE

The encryption module is based on the Rijndael algorithm in witch the block and key size are limited at 128 bits. This limitation appeared to meet the encryption speed requirements. The algorithm is composed from four major operations made to the data block:

- SubBytes;
- ShiftRows;
- MixCollumns;
- AddRoundKey.

The SubBytes transformation is defined as a non-linear byte substitution witch operates on the State. In SubBytes, the calculation of the multiplicative inverse can be efficiently done using a "table lookup" method: a small table of 28 = 256 pairs of bytes can be built once and used forever.

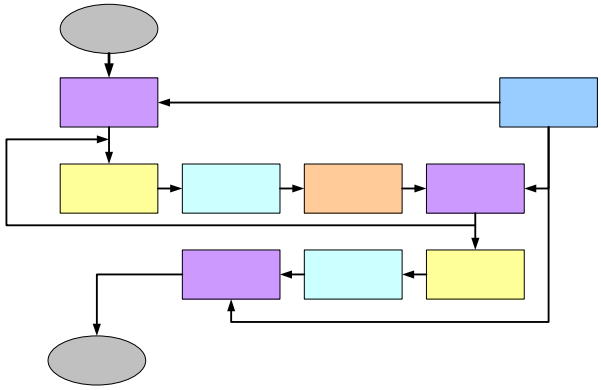

Fig. 2 Rijndael algorithm block scheme

In ShiftRows, the rows of the state are cyclically shifted over different offsets. The first row is not shifted, the second row is shifted over one byte, the third row is shifted over two bytes and, finally, the fourth row is shifted over three bytes.

The MixColumns transformation operates on the State column-by-column, treating each column as a four-term polynomial. The columns are considered as polynomials over GF (2^8) and multiplied modulo x^4+1 with the fixed polynomial: $3x^3+x^2+x+2$.

The AddRoundKey operation consists in the simple addition of the round key with the State. The Round Key is generated by means of the key schedule.

The algorithm uses the cipher key to generate an extended key. This key is used by the AddRoundKey operation.

This entire process is called Key Schedule.

- Key Schedule consist in two components:
- Key Expansion;
- Round Key selection.

The algorithm was implemented using three modes of operation: ECB[3], CBC[4], CFB[5]. In this

application the ECB mode was implemented for testing purposes because of the security problems caused by the image redundancy. This mode can only be used in tandem with a compression module.

2.3. RF MODULES

The protocols used for feeding the data to the wireless device were TCP and IP in stream mode. The TCP protocol handled packet retransmission and the 802.11 RF module handled error corrections. The radio link could not be used at it's full potential because of minimum delay required and the synchronization that was taking place in background.

3. RESULTS

In the three modes of operation (ECB, CBC, CFB) the achieved standard deviation is between 49.36 and 49.41. From the statistical point of view de difference is very small. Although ECB mode is the fastest, there are problems in using this mode because it doesn't use the process of chaining and that is why ECB mode has problems; a continuous area of the same colour has the same encrypted result. This can be resolved using an image compression module witch eliminates the redundancy.

The key length was limited at 128 bits because of the limitations imposed by the processing power and the power consumption. Although the key is small, there are 340282366920938463463374607431768211456 possible combinations.

If the key is created using a good pseudo-random generator, the key length shouldn't be a problem.

Table 2 Performance of the encryption modes used

Encryption mode	Average link load	Average frames/second
ECB	39%↔2,304 Mbps	10
CBC	40%↔2,073 Mbps	9
CFB	40%↔2,073 Mbps	9

Table 3 Standard deviation of the images in different encryption modes

Encryption mode	Std. Dev.
Original image	35,52
ECB	49,39
CBC	49,41
CFB	48,36

The histograms show that most of the values are gathered near the median.

3.1. STATIC IMAGE ENCRYPTION EXAMPLE

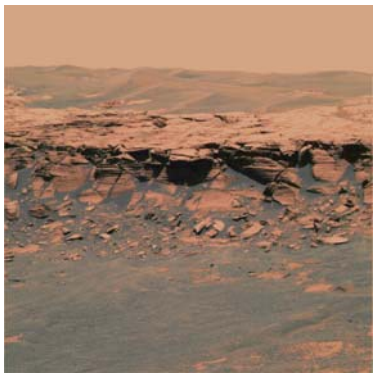

Fig. 3 Original image

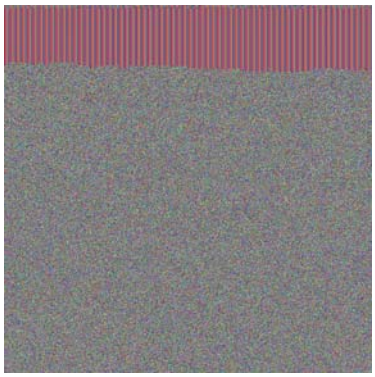

Fig. 4 Encrypted image in ECB mode

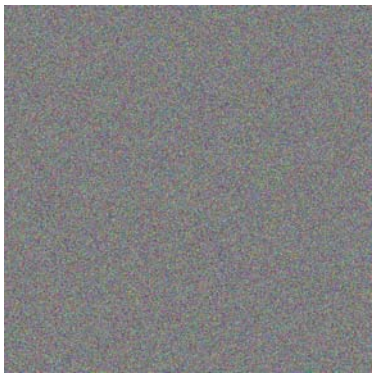

Fig. 5 Encrypted image in CBC mode

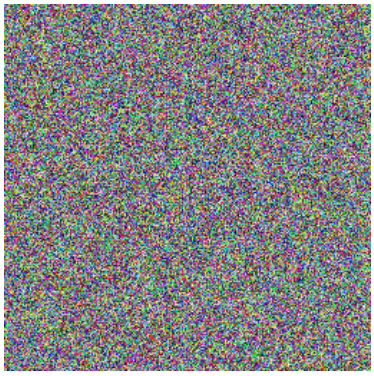

Fig. 6 Encrypted image in CFB mode

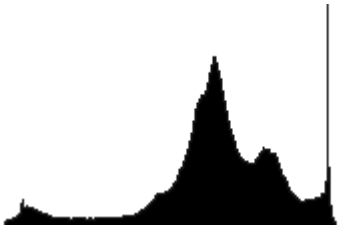

Fig. 7 Histogram of original image

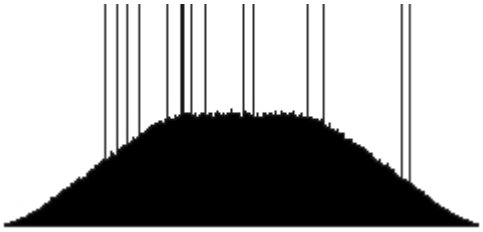

Fig. 8 Histogram of encrypted image in ECB mode

Fig. 9 Histogram of encrypted image in CBC mode

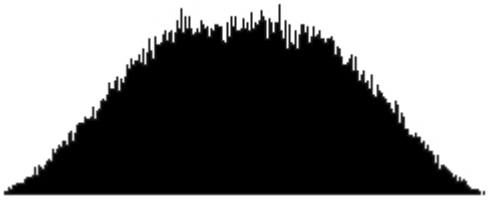

Fig. 10 Histogram of encrypted image in CFB mode

3.2. REAL-TIME ENCRYPTION EXAMPLE

Fig. 11 Decrypted image

Fig.12 Encrypted image using ECB mode

Fig. 13 Encrypted image using CBC mode

Fig. 14 Encrypted image using CFB mode

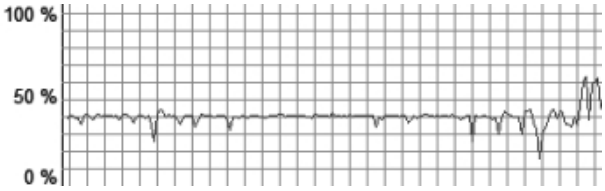

Fig. 15 Link load

REFERENCES

1. Pesce, M.D., *Programming Microsoft DirectShow for digital video and television*, Microsoft Press, 2003;
2. Uhl, A., Pommer, A., *Image and video encryption - From Digital Rights Management to Secured Personal Communication*, Springer, 2004;
3. Daemen, J., Rijmen, V., *The Design of Rijndael*, Springer, 2002;
4. * * * Microsoft Platform SDK Help;
5. Menezes, P. van Oorschot, Vanstone, S., *Handbook of Applied Cryptography*, CRC Press, 1996.