Proceeding of International Conference on Electrical Engineering, Computer Science and Informatics (EECSI 2014), Yogyakarta, Indonesia, 20-21 August 2014

Secure Cryptographic Algorithm for a Fault Tolerant model in Unmanned Aerial Vehicles

Sujatha Rajkumar[1], Ramakrishnan Malaichamy[2]
[1]*Velammal Engineering College, India,* [2]*Anna University, India*
Email: ece.sujatha@velammal.edu.in, ramkrishod@velammal.edu.in

Abstract.AES (Advanced Encryption Standard), provides the highest level of security by utilizing the strongest 128 bit Algorithm to encrypt and authenticate the data. AES commercial security algorithm has proved to be effective in Unmanned Aerial vehicles which is used for military purpose such as enemy tracking , environmental monitoring meteorology, map making etc .The demand to protect the sensitive and valuable data transmitted from UAV (Unmanned Aerial Vehicles) to ground has increased in Defense and hence the need to use onboard encryption . In order to avoid data corruption due to single even upsets (SEU's) a novel fault tolerant model of AES is presented which is based on the Hamming error correction code. For this work a problem was chosen that first addresses the encryption of UAV imaging data using the efficient AES CBC mode .A detailed analysis of the effect of single even upsets (SEUs) on imaging data during on-board encryption is carried out. The impact of faults in the data occurring during transmission to ground due to noisy channels is analyzed .The performance for the above fault tolerant model is measured using power and throughput.

Keywords: Onboard Encryption, UAV, Cipher block chain, Single Even Upset, Hamming code

1Introduction

UAV used for military applications takes images of the Earth with smart and sophisticated imaging sensors. Multispectral images of the Earth captured by optical on-board cameras can be used in monitoring the environment and disasters, vegetation control, map marking, urban planning, etc. The real problem now is towards UAV, as they require smaller budgets to build and launch and also involve less maintenance costs. The imaging payload units of UAV comprise imagers, memory, and high-rate data transceivers. UAV are equipped with on-board encryption to protect the data transmitted to the ground station. Encryption is used to protect data from unauthorized users.. But now UAV manufacturers are realizing the importance of on-board encryption to protect valuable data.At present, more and more UAV's are equipped with on-board encryption to protect the data transmitted.

Advanced Encryption Standard (also known as Rijndael) took its place, various AES implementations have been proposed both in software and hardware.

The Rijndael algorithm approved as the Advanced Encryption Standard (AES) by the US National Institute of Standards and Technology (NIST) is a block cipher, which encrypts one block of data at a time [1]. To encrypt multiple blocks, modes of operation have been defined by NIST. AES is being adopted by many organizations across the world. Because of its simplicity, flexibility, easiness of implementation, and high throughput AES is used in many different applications ranging from smart cards to big servers. In fact, hardware implementations of AES are well suited to resource-constrained embedded applications like satellites. There are various hardware implementations of the AES algorithm on platforms like application specific integrated circuits (ASICs) and field programmable gate arrays (FPGAs) that achieve a significant throughput ranging from a few Mbit/s to Gbit/s. Thus the requirements of UAV for high-rate data transmission are met by existing AES implementations. However, in addition to high throughput, immunity of the encryption process against faults is very important .UAV operate in a harsh radiation environment and consequently any electronic system used on board, including the encryption processor, is susceptible to radiation-induced faults [2]. Most of the faults that occur in on-board electronic devices are radiation-induced bit flips called single event upsets (SEUs[3].If faulty data is transmitted to the ground station, request for data retransmission has to wait until the next UAV revisit period, with revisit time varying from a couple of hours to weeks. In order to prevent faulty data transmissions, there is a need for an error-free encryption scheme on board[4]. UAV data can further get corrupted during transmission to ground due to noise in the transmission Channel. The impact of radiation on semiconductor devices on board depends on orbit altitude, orientation, and time [5]. Reliability is the most important issue in avionics design. SEUs must be detected and corrected on board before sending the data to ground. The triple modular redundancy (TMR) technique is one of the most widely used redundancy-based SEU mitigation techniques in satellites [6]. However, with the TMR technique the area and power overheads triplicate in comparison with the original module.[7] This paper addresses reliability issues of the AES algorithm. A detailed analysis of the impact of faults during on-board encryption and during transmission for the AES-CBC mode is presented.

2 Advanced Encryption Standard– Algorithm and Modes Of Operation

The AES is a symmetric key algorithm, in which both the sender and the receiver use a single key for encryption and decryption[8]. AES defines the data block length to 128 bits, and the key lengths to 128,192, or 256 bits. It is an iterative algorithm and each iteration is called a round. The total number of rounds, Nr, is 10, 12, or 14 when the key length is 128, 192, or 256 bits, respectively. Each round in AES, except the final round, consists of four transformations: Sub Bytes, Shift Rows, Mix Columns, and Add Round Key. The final round does not have the Mix Columns transformation. The decryption flow is simply the reverse of the encryption flow and each operation is the inverse of the corresponding one in the encryption process. The round transformation of AES and its steps operate on some intermediate results, called state. The state can be visualized as a rectangular matrix with four rows. The number of columns in the state is denoted by Nb and is equal to the block length in bits divided by 32. For a 128 bit data block (16bytes) the value of Nb is 4, hence the state is treated as a 4×4 matrix and each element in the matrix represents a byte. For the sake of simplicity, in the rest of the paper, both the data block and the key lengths are considered as 128 bit long. However all the discussions and the results hold true for 192 bit and 256 bit keys as well.

3 Transformations Of AES Algorithm

All the four transformations of the AES round work on the principles of finite fields. The number of the elements in a finite field is called the order of the field[9]. A finite field of order p^n is generally denoted as GF (P^n), where GF stands for Galois field, p is the characteristic of the finite field, and n is the number of bits used to represent the field elements. The basic operations of AES are defined over the elements of the field GF (2^8). The specification of AES has adopted polynomial representation of the byte elements of GF (2^8) with coefficients over the field GF (2). A polynomial representation of byte a(x) with coefficients over the field GF (2) is represented as follows:

$$a(x) = \sum a_i x^i = a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x^1 + a_0 \quad (1)$$

Where $a_i = \{0,1\}$.

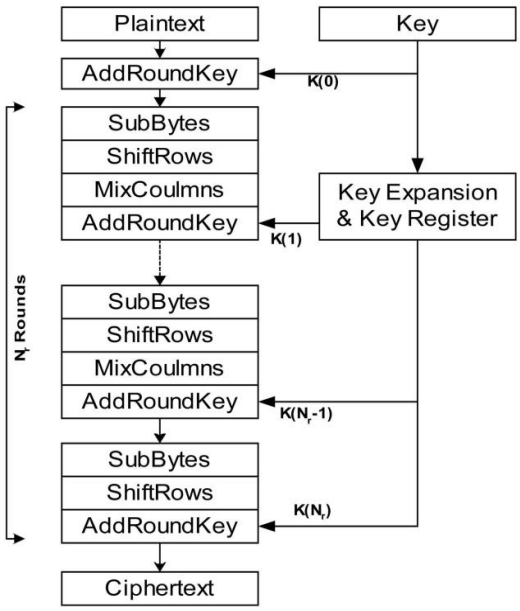

Fig .1. Block Diagram of AES Algorithm

The binary representation of the polynomial given by (1) is as follows:

$$a(x) \rightarrow a_7a_6a_5a_4a_3a_2a_1a_0 \quad (2)$$

For example, a byte element represented in binary form as 1100 0001 is written in polynomial form as $x^7 + x^6 + 1$.

The operation of addition using byte elements is defined as addition of the corresponding polynomials. In case of polynomials over GF (2) addition is just an exclusive OR (XOR) operation. For byte multiplication the following irreducible polynomial is used in AES:

$$m(x) = x^8 + x^4 + x^3 + x + 1 \quad (3)$$

Multiplication in GF (2^8) is denoted as $\otimes$. The multiplication of two polynomials a(x) and b(x) is defined as the algebraic product of the polynomials modulo the irreducible polynomial m(x) as below:

$$c(x)=a(x)-b(x)= a(x)b(x) \bmod m(x) \quad (4)$$

The SubBytes transformation is a nonlinear byte substitution, operating on each byte of the state matrix independently. Each byte of the state is first inverted in GF (2^8) and then processed through an affine transformation. The SubBytes transformation can be calculated on the fly for each state byte. Alternatively, for the sake of computational simplicity, Sub Bytes can be computed in advance and stored in a look-up table (LUT) of 2^8 = 256 elements called S-Box. Shift Rows

cyclically left shifts the last three rows of the state by 1, 2, and 3 bytes, respectively.

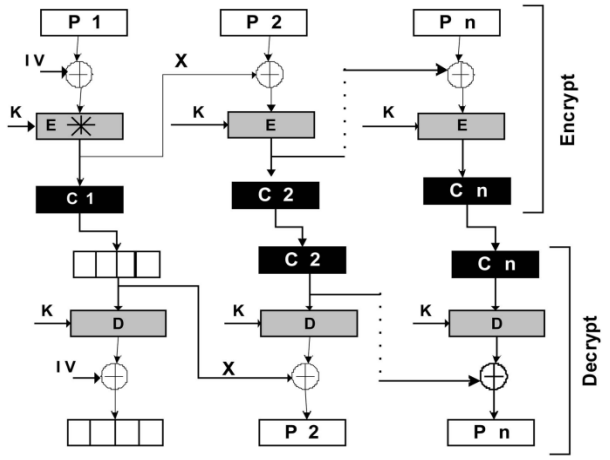

Fig.2. Fault Propagation during encryption in AES-CBC mode

Mix Columns transforms every column in the state by multiplying it with a predefined polynomial .

4 Fault propagation analysis

Due to fault propagation even a single bit error during encryption of one block of AES can result in corruption of 50% of the bits in the final encrypted data on average. In addition, when using the AES feedback modes faults occurring in one block can propagate to other blocks because of the feedback[10]. In this section propagation to subsequent blocks of single bit faults occurring both during encryption and during transmission is investigated.

4.1 Fault Propagation in AES-CBC Mode

Cipher Block Chaining Mode: The CBC mode, illustrated in Fig. 3, is the mode in which the plain data block is XOR-ed with the cipher data of the previous block before it is encrypted. The first block is XOR-ed with an initial vector (IV), which is a random number. In Fig.2,P1,P2…. Pn represent the plain data,C1,C2….Cn represent the cipher data and K is the key used in both encryption and decryption. The plain data blocks, the cipher data blocks, and the key are of 128 bit length each. The "E" and "D" blocks in Fig.2 denote an encryption and decryption function using the AES algorithm, respectively.

The effect of an SEU during encryption in the CBC mode is illustrated in Fig.2, where the SEU occurrence is marked by the star symbol * and the corrupted data blocks are represented by black boxes. If SEU occurs while encrypting the plain block P1,the cipher block C1 will be corrupted and hence the decrypted block P1 will also be corrupted. However, this corrupted data is not propagated to the subsequent blocks despite the feedback. The reason for this is that the corrupted cipher block C1 is XOR-ed twice (with the plain block P2 before encryption and with the cipher block C2 after decryption) as shown in Fig.2.Performing the XOR operation two times with this corrupted cipher block C1 neutralizes the fault and prevents propagation of faults to subsequent blocks as shown below:

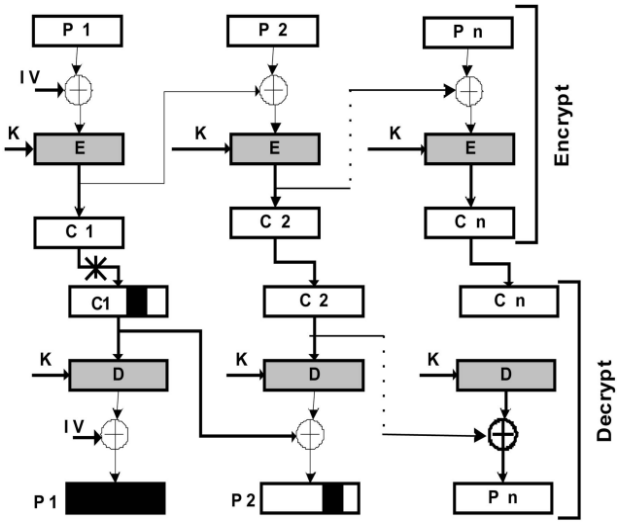

Fig.3.Fault Propagation during transmission in CBC mode

In contrast, a fault occurring in an encrypted block during transmission propagates to the next block, as shown in Fig.3,where the transmission fault is shown by the star symbol , during the transmission of the cipher block C1. The decrypted block P1 is completely garbled and the subsequent decrypted block P2 will have bit errors at the same position as the original erroneous block C1.The decrypted blocks following the second block will not be affected by the fault. Hence the CBC mode is self synchronizing.

5 Fault-tolerant model of the AES Algorithm

This section presents a novel fault-tolerant model for the AES algorithm, which is immune to radiation-induced SEUs occurring during encryption and can be used in hardware implementation in an on-board small OE satellites. The model is based on a self-repairing EDAC scheme, which is built in the AES algorithmic flow and utilizes the Hamming error correcting code. The proposed Hamming code based fault-tolerant model of AES can be adapted to all the five modes of AES to correct SEUs on board. Even though the calculation of the Hamming code is carried out within the AES, it does not alter any of the transformations of the algorithm and does not affect in any way the operation of AES. The disadvantage of this method is that the implementation of the codes based EDAC will require an additional encoding stage to encode the

plain data blocks which will inevitably add an overhead to on-board resources and processing.

5.1 *Model Description*

The proposed fault-tolerant model is based on the singe error correcting Hamming code (12,8), the simplest of the available error correcting codes. The Hamming code (12,8) detects and corrects a single bit fault in a byte and it is a good choice for satellite applications, as most frequently occurring faults in on-board electronics are bit flips induced by radiation .However, the AES correction model can be extended to correct multiple bit faults by using other error correcting codes such as the modified Hamming code (16,8), the Read-Solomon codes, etc. The procedure to calculate the parity check bits is discussed below.

Calculation of the Hamming Code**:** The parity check bits of each byte of the S-Box LUTs are pre calculated. These Hamming code bits can be formally expressed as below:

$$h(\mathbf{S}_{RD}[a]) \rightarrow \mathbf{h}RD[a]$$

$$h((\mathbf{S}_{RD}[a] \otimes 2) \rightarrow \mathbf{h}_{2RD}[a] \quad (6)$$

$$h((\mathbf{S}_{RD}[a] \otimes 3) \rightarrow \mathbf{h}_{3RD}[a]$$

where a is the state byte and h represents the calculation of the Hamming code.

As can be seen from (6), $\mathbf{h}_{RD}$ is given by the parity check bits of the S-Box LUT $\mathbf{S}_{RD}$, $\mathbf{h}_{2RD}$ is given by the parity check bits of ($\mathbf{S}_{RD} \otimes 02$), and $\mathbf{h}_{3RD}$ is given by the parity check bits of ($\mathbf{S}_{RD} \otimes 03$).The procedure to derive the $\mathbf{h}_{RD}$ parity bits is described below by taking one state byte a, represented by bits (b_7,b6,b5,b4,b3,b2,b1,b0) as an example. The Hamming code of the state byte a is a four-bit parity code, represented by bits (p_3, p_2 ,p_1,p_0),which are derived as follows:

p3→ is parity of bit group b_7,b_6,b_4,b_3,b_1

p2 → is parity of bit group b_7,b_5,b_4,b_2,b_1

p1 → is parity of bit group b_6,b_5,b_4,b_0 (7)

p0 → is parity of bit group b_3,b_2,b_1,b_0

The Hamming bits of all the bytes of table $\mathbf{S}_{RD}$ are Pre-calculated and stored in the form of a memory table referred to as the $\mathbf{h}_{RD}$ table. The Hamming code $\mathbf{h}_{2RD}$ is given by the parity check bits of (**S**RD⊗ 02). The Galois field multiplication of a state byte a with {02} is defined as follows

$$\{02\} \otimes a = \{02\}.a(x) \bmod m(x)$$

$$= x.a(x) \bmod m(x)$$

$$= (x. \sum_{i=0}^{n-1} a_i\, x^i) \bmod m(x)$$

$$= a_{n-1} \sum_{i=0}^{n-1} m_i\, x^i + \sum_{i=0}^{n-2} a_i\, x^{i+1} \quad (8)$$

$$= an\text{-}1\ m0 + (\sum_{i=0}^{n-2} a_{n-1} m_{i+1} + a_i) x^{i+1}$$

where{02} is represented as x in polynomial form, mi represent the coefficients of the irreducible polynomial m defined in the AES algorithm.

The Hamming code of the above product **h**2RD is calculated as follows

$$\mathbf{h}_{2RD} = h(\{02\} \otimes a)$$
$$= h(a_{n-1}m_0 + \sum_{i=0}^{n-2} a_{n-1} m_{i+1} + a_i) \quad (9)$$

Using the irreducible polynomial given by (3) and n = 8, (9) can be rewritten as

$$\mathbf{h}_{2RD} = h(a_7 + \sum_{i=0}^{6} a_7 m_{i+1} + a_i) \quad (10)$$

Unlike the calculation of parity bit of a byte, the calculation of Hamming parity bits depends on the position of the bits in a byte and therefore it is not possible to further simplify (10). Hence the $\mathbf{h}_{2RD}$ parity bits are calculated beforehand, and are stored in the form of a memory table, which is referred to as the $\mathbf{h}_{2RD}$ table. The Hamming table **h**3RD is given by the parity check bits of ($\mathbf{S}_{RD} \otimes \{03\}$). The Galois field multiplication of a state byte a by {03} can be described as follows.

$$\{03\} \otimes a = (\{02\} \oplus \{01\}) \otimes a$$

$$= x.a(x) \bmod m(x) \oplus a(x) \bmod m(x) \quad (11)$$

Similar to the parity function, the Hamming function is also a linear operator. The Hamming code of the above product $\mathbf{h}_{3RD}$ is written as follows.

$$\mathbf{h}_{3RD} = h(\{03\} - a) = \mathbf{h}_{2RD} \oplus \mathbf{h}_{RD} \quad (12)$$

Hence, the $\mathbf{h}_{3RD}$ parity bits can be calculated from the$\mathbf{h}_{RD}$ and $\mathbf{h}_{2RD}$ parity bits and therefore it is not necessary to store them in the form of a parity memory table. Once we have all the parity bits, the next step is to detect and correct the faults by predicting the Hamming code bits using the pre-calculated Hamming code bits.

5.2 Detection and Correction of Fault Using Hamming Code Bits

The Hamming code matrix of the Sub Bytes transformation is predicted by referring to the $\mathbf{h}_{RD}$ table. The Hamming code matrix prediction for Shift rows involves a simple cyclic rotation of the Sub Bytes Hamming code bits. The

Hamming code state matrix for Mix Columns is predicted with the help of the h_{RD},$h_{2RD}$and h_{3RD} parity bits and is expressed by the equation below

$$\mathbf{h}_{0,j} = \mathbf{h}_{2RD}[a_{0,j}] \oplus \mathbf{h}_{3RD}[a_{1,j}] \oplus \mathbf{h}_{RD}[a_{2,j}] \oplus \mathbf{h}_{RD}[a_{3,j}]$$

$$\mathbf{h}_{1,j} = \mathbf{h}_{RD}[a_{0,j}] \oplus \mathbf{h}_{2RD}[a_{1,j}] \oplus \mathbf{h}_{3RD}[a_{2,j}] \oplus \mathbf{h}_{RD}[a_{3,j}]$$

$$\mathbf{h}_{2,j} = \mathbf{h}_{RD}[a_{0,j}] \oplus \mathbf{h}_{RD}[a_{1,j}] \oplus \mathbf{h}_{2RD}[a2,j] \oplus \mathbf{h}_{3RD}[a_{3,j}]$$

$$\mathbf{h}_{3,j} = \mathbf{h}_{3RD}[a_{0,j}] \oplus \mathbf{h}_{RD}[a_{1,j}] \oplus \mathbf{h}RD[a_{2,j}] \oplus \mathbf{h}_{2RD}[a_{3,j}]$$

$$0 \le j \le 4 \qquad (13)$$

By substituting (12) in (13), the Hamming code matrix for MixColumns can be predicted with just two tables, $\mathbf{h}_{RD}$ and $\mathbf{h}_{2RD}$.As shown in Fig. 4, for each transformation, the Hamming code is predicted using the input data state to the transformation by referring to the parity check bit tables and also the parity check bits are calculated from the output of the transformation. The predicted and calculated check bits are compared as discussed below.

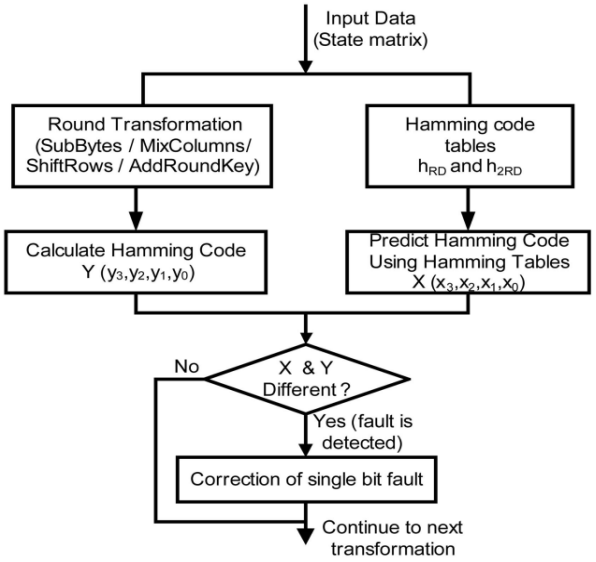

Fig. 4. Fault detection and correction

flow chart

Let the predicted check bits of the transformation input be represented by (x_3,x_2,x_1,x_0) and the calculated check bits of the transformation output be represented by (y_3,y_2,y_1,y_0). The location of the faulty bit is detected by comparing the predicted and calculated Hamming check bits .Once the faulty bit position is identified, the fault correction is performed by simply flipping the particular bit. The encryption is then continued without any interruption. Here we assume that the Hamming code tables will be protected from SEUs by traditional memory protection techniques in satellite applications like memory scrubbing and refreshing.

5.3 Hardware and Software Details

In UAV high throughput encryption processing is required to comply with high-rate data transmission bandwidth up to a few hundred Mbit/s . In order to meet the requirement for high throughput processing, hardware implementation is considered to be the preferred choice. The Verilog hardware description language (HDL) can be used for coding, and model sim can be is used for the functional, pre synthesis, and post synthesis simulations of the design. The HDL designs will be tested extensively using the KAT and MCT vectors provided by NIST. Synthesis and implementation are carried out using Xilinx ISE, respectively.

6 Conclusion

This paper investigates the reliability of the AES CBC algorithm to encrypt UAV data for defense application . The AES CBC algorithm was discussed in detail and its advantages and disadvantages for encryption of multispectral images are compared. The impact of the propagation of SEU faults occurring during on-board encryption is analyzed. In addition, an analysis of the propagation of faults that occur during transmission due to noise is carried out. In order to avoid data corruption due to SEUs, a fault detection and correction model is proposed based on the Hamming code (12,8). The model provides an SEU self-recovering capability, which is built in the AES data path. Also it consumes a very small portion of the power available to the payload unit. The estimated hardware overhead of the optimal fault-tolerant AES IP core is 49% in terms of area and 152% in terms of power. The model can be extended for detection and correction of multiple bit faults by using other more sophisticated error-correcting codes such Reed-Solomon codes, etc. The proposed fault detection and correction AES model targets the defense application domain (UAV) .

References

[1]T.Mangaiyarkarasi and B.Nandhini, Fault and tolerant-method using AES for images, International Journal of Communications and Engineering, Vol No.5, Issue:01,March 2012

[2]Roohi Banu, Tanya Vladimirova,and Martin N.Sweeting, IEEE Surrey Space Centre,Fault-Tolerant Encryption for Space Applications (2009)

[3] Sun, W., Stephens, P., and Sweeting, M. Micro-mini Satellites for affordable EO constellations–Rapid Eye and DMC .In Proceedings of the IAA Symposium on Small Satellites for Earth Observation, Berlin, Germany, Apr. 2001,IAA-B3-0603.

[4]Directory of Earth Observation Re- sources. http://directory.eoportal.org/pres TopSat.html (last accessed 18th June 2007).

[5] Mariani, R., and Boschi, G.Scrubbing and partitioning for protection of memory systems. In Proceedings of the 11th

IEEE International Symposium on On-Line Testing, July 6—8, 2005, 195—196.
[6] B. Olsen. "Encryption Gets a Boost", *Federal Computer Week*,August2004,URL: http://www.fcw.com/fcw/articles/2004/0823/fast-encryption.asp
[7]Consultative Committee for Space Data Systems Security threats against space missions. Informational Report CCSDS 350.1-G-1, Green Book, NASA, Washington, D.C., Oct. 2006.
[8]Sweet, K. The increasing threat to satellite communications. Online Journal of Space Communication, 6 (Nov. 2003).
[9] "Critical Infrastructure Protection. Commercial Satellite Security Should Be More Fully Addressed", US Government Accountability Office Report, GAO-02-781,August 2002.
[10] K. Poulsen, "Satellites at Risk of Hacks", *Security Focus*,Oct 2002,URL : http://www.securityfocus.com/news /942
[11] NASA/GSFC. "IP-in-Space Security Handbook", September 001.URL:http://ipinspace.gsfc.nasa.gov/documents/
[12] W. Stallings "Cryptography and Network Security – Principles and Practices", 3rd edition, Prentice-Hall, 2002.
[13] B. Schneier, "Applied Cryptography – Protocols, Algorithms and Source Code in C", Second Edition.
[14]Advanced Encryption Standard, Wikipedia,URL:http://en.wikipedia.org/wiki/Advanced_Encryption_Standard
[15] K. Lauter, "The advantages of Elliptic Curve Cryptography for Wireless Security", Kristin Lauter, *IEEE Wireless Communications*, February 2004.
[16] T. Wollinger, J. Guajardo, C. Paar, " Cryptography in Embedded Systems: AnOverview", *Proceedings of the Embedded World 2003 Exhibition and Conference*,pp. 735-744, Design and Electronik, Nurenberg, Germany, February 2003.
[17]H.Weiss,J.Stanier,"SpaceMission Communications Security, GSAW2001URLhttp://sunset.usc.edu/events/GSAW/gsaw2001/SESSION9/Shave.pdf