The Rijndael Block Cipher (AES Proposal): A Comparison with DES

C. Sanchez-Avila[†] & R. Sanchez-Reillo[‡]

[†]Dpto. de Matemática Aplicada, E.T.S.I. Telecomunicacion, Universidad Politecnica de Madrid, 28040 Madrid, Spain.

[‡] Dpto. de Ingenieria Electrica, Electronica y Automatica, Universidad Carlos III de Madrid, 28911 Madrid, Spain.

Abstract - In October 2000, and after three years of competition between 15 candidate algorithms, the National Standards and Technology (NIST) chose the Rijndael algorithm to be adopted as Advanced Encryption Standard (AES) by the U.S. Department of Commerce, replacing to Data Encryption Algorithm (DES), which has been the standard since 1977.

In this work we analyze the structure and design of new AES, following three criteria: a) resistance against all known attacks; b) speed and code compactness on a wide range of platforms; and c) design simplicity; as well as its similarities and dissimilarities with other symmetric ciphers. On the other side, the principal advantages of new AES with respect to DES and T-DES, as well as its limitations, are investigated. Thus, for example, the fact that the new cipher and its inverse use different components, which practically eliminates the possibility for weak and semi-weak keys, as existing for DES, and the non-linearity of the key expansion, which practically eliminates the possibility of equivalent keys, are two of the principal advantages of new cipher.

Finally, the implementation aspects of Rijndael cipher and its inverse are treated. Thus, although Rijndael is well suited to be implemented efficiently on a wide range of processors and in dedicated hardware, we have concentrated our study on 8-bit processors, typical for current Smart Cards and on 32-bit processors, typical for PCs.

Introduction

In 1997, the National Institute of Standards and Technology (NIST) initiated a process to select a symmetric-key encryption algorithm to be used to protect sensitive information in furtherance of NIST's statutory responsibilities. In 1998, NIST announced the acceptance of fifteen candidate algorithms and requested the assistance of the cryptographic research community in analyzing the candidates. This analysis included an initial examination of the security and efficiency characteristics for each algorithm. NIST reviewed the results of this preliminary research and selected MARS, RC6, Rijndael, Serpent and Twofish as finalists. An interesting performance comparison of these algorithms can be found in [11]. On *October 2000* and having reviewed further public analysis of the finalists, NIST decided to propose Rijndael as the Advanced Encryption Standard (AES).

Rijndael, designed by Joan Daemen (Proton World International Inc.) and Vincent Rijmen (Katholieke Univeriteit Leuven) of Belgium, is a blockcipher with a simple and elegant structure [2]. In fact, Rijndael has an easily understandable mathematical (rich algebraic) structure, that can easily be split in its components. Nevertheless, this property should not be confused with the question whether it can easily be broken.

Design of Rijndael

Rijndael, a variant to the Square blockcipher due to the same authors [4], is not a Feistel cipher like DES. As it is well known, in DES and other Feistel ciphers, in each round one half of the block is subjected to a function which involves subkey material, which needs to be nonlinear and which may not be (and, in fact, is not in DES) invertible. However, the round as a whole is guaranteed to be invertible since the half of the block subjected to that function is not changed. Instead, the other half of the block is changed, by being subjected to an XOR operation with the output of that function. Thus, a round is its own inverse, and decryption is the same as encryption, except that the subkeys are used in reverse order.

Contrary to DES, and other block ciphers, the round transformation of Rijndael does not have the Feistel structure. Instead, the round transformation is composed of three distinct invertible uniform transformations, called layers. The specific choices for the different layers are for a large part based on the application of the Wide Trail Strategy [5], a design method to provide resistance against linear and differential cryptanalysis. In the Wide Trail Strategy, every layer has its own function:

- *The linear mixing layer*: guarantees high diffusion over multiple rounds.
- *The nonlinear layer*: corresponds to the parallel application of S-boxes that have optimum worst-case nonlinearity properties.
- *The key addition layer*: is a simple XOR of the Round Key to the intermediate State.

A key addition layer is applied before the first round. The motivation for this initial key addition is the following. Any layer after the last key addition in the cipher can be simply peeled off without knowledge of the key and therefore does not contribute to the security

0-7803-6636-0/01/$10.00 ©2001 IEEE

of the cipher (e.g. the initial and final permutation in the DES). On the other side, in order to make the cipher and its inverse more similar in structure, the linear mixing layer of the last round is different from the mixing layer in the other rounds. It can be shown that this does not improve or reduce the security of the cipher in any way. This is similar to the absence of the swap operation in the last round of the DES.

Structure and description of Rijndael

Mathematical preliminaries

As we will describe, several operations in Rijndael are defined at byte level, with bytes representing elements in the Galois field $GF(2^8)$. As its is known, the elements of a finite field can be represented in several ways. For any prime power there is a single finite field, hence all representations of finite field $GF(2^8)$ are isomorphic [7]. Despite this equivalence, and considering the impact of the representation on the implementation complexity, the classical polynomial representation has been chosen. Thus, we can write

$$GF(2^8) = \{a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + \\ + a_2x^2 + a_1x + a_0 \ : \ a_i \in Z_2\} \quad (1)$$

Thus, a byte consisting of bits $b_7b_6b_5b_4b_3b_2b_1b_0$ can be considered as a polynomial with coefficient in $\{0,1\}$. For example, the byte 11001010 corresponds with polynomial $x^7 + x^6 + x^3 + x$. In the polynomial representation, multiplication in $GF(2^8)$ corresponds with multiplication of polynomials mod$(F(x)$, being $F(x)$ an irreducible polynomial of degree 8. For Rijndael,

$$F(x) = x^8 + x^4 + x^3 + x + 1 \quad (2)$$

Moreover, in Rijndael, other operations are defined in terms of 4-byte words. But, it is possible to define polynomials with coefficients in $GF(2^8)$. In this way, a 4-byte word corresponds with a polynomial of degree below 4. In this case, multiplication of these polynomials needs a polynomial of degree 4, in order to reduce the product to a polynomial of degree below 4. In Rijndael, this is done with the polynomial

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

$M(x)$ is not an irreducible polynomial over $GF(2^8)$, hence multiplication by a fixed polynomial is not necessarily invertible. In Rijndael, a fixed polynomial that does have an inverse has been chosen.

As the addition in $GF(2^8)$ is the bitwise XOR, the addition of two polynomials with coefficient in this finite field is a simple bitwise XOR. However, multiplication is more complicated. Thus, assuming we have two polynomials with coefficient in $GF(2^8)$,

$$\begin{aligned} p(x) &= p_3x^3 + p_2x^2 + p_1x + p_0 \\ q(x) &= q_3x^3 + q_2x^2 + q_1x + q_0 \end{aligned} \quad (4)$$

the modular product of $p(x)$ and $q(x)$, (i.e. $(p(x) \cdot q(x)) \mathrm{mod} M(x)$), denoted by $r(x) = p(x) \otimes q(x)$ is given by

$$r(x) = r_3x^3 + r_2x^2 + r_1x + r_0 \quad (5)$$

with

$$\begin{aligned} r_0 &= p_0q_0 \oplus p_3q_1 \oplus p_2q_2 \oplus p_1q_3 \\ r_1 &= p_1q_0 \oplus p_0q_1 \oplus p_3q_2 \oplus p_2q_3 \\ r_2 &= p_2q_0 \oplus p_1q_1 \oplus p_0q_2 \oplus p_3q_3 \\ r_3 &= p_3q_0 \oplus p_2q_1 \oplus p_1q_2 \oplus p_0q_3 \end{aligned} \quad (6)$$

or expressed as matrix multiplication

$$\begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} p_0 & p_3 & p_2 & p_1 \\ p_1 & p_0 & p_3 & p_2 \\ p_2 & p_1 & p_0 & p_3 \\ p_3 & p_2 & p_1 & p_0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (7)$$

Structure of Rijndael

Rijndael is an iterated block cipher. It has a variable block length b and a variable key length k, which can be set to 128, 192 or 256 bits. The recommended number nr number of rounds is determined by b and k and varies between 10 and 14, as it is shown in Table 1.

nr	b=4	b=6	b=8
k=4	10	12	14
k=6	12	12	14
k=8	14	14	14

Table 1: Number of rounds (nr) as a function of the block and key length.

In Rijndael, the State (i.e. the intermediate cipher result), S, can be written as a rectangular array of bytes with four rows and Nb columns, being $Nb = LB/32$, where LB is the block length. The cipher key is similarly written as a rectangular with four rows and Nk columns, being $Nk = LK/32$, where LK is the key length. The input and output at its external interface are considered to be one-dimensional arrays of bytes numbered upwards from 0 to $4Nb - 1$. The Cipher Key is also considered to be a one-dimensional array of bytes numbered upwards from 0 to $4Nk - 1$.

Thus, considering B the plaintext block., K the key and nr the number of rounds, we can describe the behavior of AES as follows:

1. Compute subkeys $K_0, K_1, ..., K_n$ from the key K
2. $S = B \oplus K_0$
3. For $i = 1$ to $nr - 1$
   - 3.1 S = ByteSub(S)
   - 3.2 S = ShiftRow(S)
   - 3.3 S = MixColumn(S)
   - 3.4 $S = K_i \oplus S$

4. $S = \text{ByteSub}(S)$

5. $S = \text{ShiftRow}(S)$

6. $S = K_n \oplus S$

The inverse transformation can be described by the following steps:

1. Compute subkeys $K_0, K_1, ..., K_n$ from the key K

2. $S = B \oplus K_n$

3. $S = \text{InvShiftRow}(S)$

4. $S = \text{InvByteSub}(S)$

5. $S = K_n \oplus S$

6. For $i = nr - 1$ to 1

   3.1 $S = K_i \oplus S$

   3.2 $S = \text{InvMixColumn}(S)$

   3.3 $S = \text{InvShiftRow}(S)$

   3.4 $S = \text{InvByteSub}(S)$

7. $S = K_0 \oplus S$

As we can see, in the direct transformation, each round transformation is composed of four different functions, except the final round which involves only three. We briefly describe these functions and their respective inverses.

a) The ByteSub function

The function ByteSub is a nonlinear byte substitution, operating on each byte of S independently by an invertible S-box which is obtained by the composition of two transformations:

1. Each byte is represented as an element of $GF(2^8)$ and substituted by its multiplicative inverse in $GF(2^8)$. The value 0 is mapped onto itself.

2. Then, an affine transformation (over $GF(2^8)$ defined by

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad (8)$$

is applied, being $x_0, x_1, ..., x_7$ the bits of corresponding byte and $y_0, y_1, ..., y_7$ the bits of resultant byte. The function InvByteSub is the aplication of the inverse of the corresponding S-box to each byte of S.

b) The ShiftRow function

In this function, the rows of S are cyclically shifted over different offsets. These depend on the block length Nb as we show in Table 2.

Nb	4	6	8
Row 0	0	0	0
Row 1	1	2	3
Row 2	1	2	3
Row 3	3	3	4

Table 2: Shift offsets for different block lengths.

The InvShiftRow function is a cyclic shift of the rows of S the same number of positions, but on the left.

c) The MixColumn function

In this function, the columns of S are considered as polynomials over $GF(2^8)$ and multiplied mod$(M(x))$, being $M(x)$ the polynomial given in (3), with a fixed ploynomial $c(x)$ given by

$$c(x) = \text{'03'}x^3 + \text{'01'}x^2 + \text{'01'}x + \text{'02'}, \quad (9)$$

where '03', '01' and '02' express hexadecimal values corresponding to $x + 1, 1$ and x, respectively.

In the InvMixColumn function, every column is transformed by multiplying it with the polynomial $d(x)$ defined by

$$c(x) \otimes d(x) = \text{'01'}, \quad (10)$$

and given by

$$d(x) = \text{'0B'}x^3 + \text{'0D'}x^2 + \text{'09'}x + \text{'0E'}, \quad (11)$$

being '0B', '0D', '09' and '0E' the hexadecimal values corresponding to $x^3 + x + 1, x^3 + x^2 + 1, x^3 + 1$ and $x^3 + x^2 + x$, respectively.

d) The Round Key addition

In this operation a simple bitwise XOR is applied between S and K_i (being length(K_i) is equal to the block length Nb). As it is known, this operation is its own inverse.

Subkeys calculation

Considering that total number of Round Key bits is equal to the block length multiplied by $nr + 1$, this computation consists of two components: the Key Expansion and the Round Key Selection, briefly described as follows.

Key expansion

This function depends on the value of Nk: there is a version for $Nk \leq 6$ and a version for $Nk > 6$.

If Nk $\leq$ 6
For $i = 0$ to $Nk - 1$

1. $W(i) = (K(4i), K(4i + 1), K(4i + 2), K(4i + 3))$

For $i = Nk$ to $Nb(nr + 1)$

2. temp = $W(i-1)$

3. If $i \bmod Nk = 0$

   3.1 temp = SubByte(RotByte(temp)) $\oplus$ Rc(i/Nk)

4. $W(i) = W(i-Nk) \oplus$ temp

If Nk > 6
For $i = 0$ to $Nk - 1$

1. $W(i) = (K(4i), K(4i+1), K(4i+2), K(4i+3))$

For $i = Nk$ to $Nb(nr+1)$

2. temp = $W(i-1)$

3. If $i \bmod Nk = 0$

   3.1 temp = SubByte(RotByte(temp)) $\oplus$ Rc(i/Nk)

4. If $i \bmod Nk = 4$

   4.1 temp = SubByte(temp)

5. $W(i) = W(i-Nk) \oplus$ temp

In this description, SubByte(X) returns a 4-byte word in which each byte is the result of applying the S-box to the byte at the corresponding position in the input word. RotByte(X) returns a word in which the bytes are a cyclic left shift (1 position) of those in its input. Rc(j) is a constant independent of Nk and defined by

$$Rc(j) = (R(j), '00', '00', '00') \tag{12}$$

being $R(j)$ is an element in $GF(2^8)$ with a value of x^{j-1}, so that

$$\begin{aligned} &R(1) = 1 \\ &R(2) = x = x \cdot 1 \\ &R(3) = x^2 = x \cdot x \\ &\vdots \\ &R(j) = x \cdot R(j-1). \end{aligned} \tag{13}$$

The key expansion function specifies the derivation of the subkeys in terms of K. In order to be efficient on 8-bit processors, a byte oriented expansion scheme has been adopted [2]. The application of SubByte ensures the nonlinearity of the scheme, without adding much space requirements on an 8-bit processor.

Round Key Selection

Finally, the subkeys are selected from buffer words $W(Nb\,i), ..., W(Nb\,(i+1)$. Thus, for example, if $Nb = 6$ and $Nk = 4$, the subsequent subkeys are given by

$$\begin{aligned} &K_0 = [W(0), W(1), W(2), W(3), W(4), W(5)] \\ &K_1 = [W(6), W(7), W(8), W(9), W(10), W(11)] \\ &\vdots \\ &K_i = [W(Nb\,i), ..., W(Nb\,(i+1)-1)] \end{aligned} \tag{14}$$

For implementation aspects, we refer to [2].

Some advantages and limitations of Rijndael

As main advantages of Rijndael we can mentioned: a) simplicity of design (the cipher does not base its security on obscure and not well understood interactions between arithmetic operations); b) variable block length (the block lentgs of 192 and 256 bits allows the construction of a collision-resistant iterated hash function using Rijndael as the compression function); and c) the possibility of extensions (although the number of rounds is fixed in the specifications, it can be modified as a parameter in case of security problems [2]). Concerning to the implementation aspects: a) Rijndael can be implemented on a Smart Card in a small account of code, using a small account of RAM and taking a small number of cycles; and b) the round transformation is parallel by design, which is an important advantage in future processors and dedicated hardware.

The limitations of new cipher is related with its inverse: a) the cipher and its inverse make use of different code and/or tables; and b) the inverse cipher can only partially re-use the circuitry that implements the cipher.

A comparison between Rijndael and DES

Despite the fact that Rijndael has a very different structure from that of DES [10], and in some ways could be said to more closely resemble SAFER[1], developed by the Cylink Corporation, because the ByteSub step directly alters the bytes to be encrypted, and the MixColumn step causes every byte in a column to affect every other byte there, somewhat as the PHT stage in SAFER involves the whole block, it is still possible to relate the fundamental steps in Rijndael to parts of DES based on the function they perform in contributing to the step of the overall cipher.

Both ciphers have round functions with three layers (though in a different order): an S-Box layer (including the expansion function E in DES), a linear diffusion layer and a subkey XOR layer.

The Round Key addition step in Rijndael clearly corresponds to the XOR of subkey material with the input to the f function in DES.

The MixColumn function in Rijndael is where the different bytes interact with each other, so it corresponds to the XOR of the f function output with the left half of the block in DES.

The ByteSub function contributes the nonlinearity in Rijndael, and so it corresponds to the f function itself in DES.

The ShiftRow function ensures that the different bytes of each row do not only interact with the corresponding byte in other rows. Thus, it corresponds to permutation P whitin DES.

An study concerning to the comparison between some aspects of Rijndael and DES, specially the characteristics of linear diffusion layer in both ciphers, we can

[1] A variant of SAFER, SAFER+, was one of the candidate algorithms in the AES process, but it was not chosen as one of the five finalists.

found in [9] and answered by authors of Rijndael in [3].

Symmetry and weak keys

Despite the large amount of symmetry, care has been taken to eliminate symmetry in the behavior of Rijndael. Thus, the function of key expansion which provides resistance against known attacks, also plays an important role in the elimination of symmetry: a) in the round transformation (this symmetric can be removed by having round constants in the key schedule); and b) between the rounds (the round transformation is the same for all rounds, this equality can be removed by having round-dependent round constants in the key schedule.

The fact that the cipher and its inverse use different components practically eliminates the possibility for weak and semi-weak keys, as existing for DES. The nonlinearity of the key expansion practically eliminates the possibility of equivalent keys.

Differential and linear cryptanalysis

As it is known the difference propagation and input-output correlation in boolean mappings and iterated boolean transformations are exploited in differential cryptanalysis (DC) [1] and in linear cryptanalysis (LC) [8], respectively. Both DC and LC were successfully applied on the block cipher DES. Later, Hellman and Langford published an attack on an 8-round variant of DES that combines the mechanisms of differential and linear cryptanalysis [6].

In [5], a detailed treatment of difference propagation and correlation in Rijndael and its relation with possible DC and LC attacks, are given. Thus, an S-box of a specific round is said to be active with respect to a linear trail if its output selection vector is nonzero for that linear trail; and it is said to be active with respect to a differential trail if its input difference vector is nonzero for that differential trail. Both, for linear and differential trails it can be seen that the weight of a trail is the sum of the active S-boxes. Consequently, in order to eliminate low weight trails, two possible mechanism are suggested:

- Choose an S-box where the maximum propagation ratio and the maximum input-output correlation are as small as possible.
- Design the round transformation in such a way that only trails with many S-boxes occur.

The Wide Trail Strategy, adopted in Rijndael, emphasizes the second mechanism. The round transformation must be designed in such a way that linear or differential steps with only few active S-boxes are followed by linear or differential steps with many active S-boxes. This is closely linked to the concept of diffusion (which denotes the quantitative spreading of information). The only requirements for the S-boxes themselves is that their input-output correlations have a certain minimum correlation weight and their difference propagations have a certain minimum restriction weight. But, the Wide Trail Strategy does not restrict the nonlinear step to juxtaposed transformations, it can equally well be applied to the shift-invariant transformations [5].

This strategy contrasts highly with the approach taken by the majority of cryptographic researchers working in cipher design. This traditional approach is dominated by the structure of DES and fully concentrates on the S-boxes. Typically, the S-boxes are assumed to be located in the f function of a Feistel structure. The S-boxes are considered to be the active elements in the cipher and must be designed following several criteria, such as maximum input-output correlation, maximum propagation ratio and diffusion criteria. These criteria impose conflicting restrictions, and finding S-boxes that have an acceptable score with respect to all them becomes less difficult when their size grows. This has led many researchers to the conclusion that resistance against DC and LC is best realized by adopting large S-box. However, this point of view plainly ignores the potential of high diffusion provides by a well-designed round transformation, like it is shown in Rijndael.

On the other side, we can see that the design of the S-box in Rijndael involves the same $GF(2^8)$ as the MixColumn function might also appear to be a concern. However, the Rijndael S-Box is nearly ideal in resistance to DC, and it is also excellent in avoiding any approximations in $GF(2^8)$ usable in the $GF(2^8)$ equivalent to LC.

The choice of Rijndael over the other finalist algorithms, also believed to be highly secure, was based primarily on its efficiency and low memory requirements. These, together with the fact that existing cryptanalysis of Rijndael are based on reduced-round variants somewhat close to the actual cipher (although the results close to the actual number of rounds are quite impractical to exploit) means that some controversy, even if not the intense controversy surrounding DES, may haunt the new AES as well.

A performance comparison between Rijndael, DES and T-DES

After comparing theoretically Rijndael with DES, it is time to analyse the performance of each of the above mentioned algorithms. Due to the fact that with DES, no operations with keys longer than 128 bits are possible, the following cases have been studied:

- DES with 64-bit key, and data length of, also, 64 bits.
- DES in CBC configuration, in order to compute 128 bits of data with a 64-bit key.
- T-DES with 128-bit key, and data length of 64 bits.
- T-DES in CBC configuration, with data length of 128 bits, and 128-bit key.

- Rijndael algorithm in its simplest form: 128-bit key, 128-bit data length.

Results obtained with longer keys or longer key data block sizes have not been included, because they provide a more powerful functionality, only available in Rijndael, and not with DES or T-DES.

Table 3 and Table 4 show the results obtained cyphering and de-cyphering multiple times (100000 for Table 3, and 100 for Table 4), in order to minimize the effects given by data communication, variable initialization, etc. (all common steps among the algorithms).

	DES 64,64	DES 64,128	T-DES 128,64	T-DES 128,128	Rijndael 128,128
Cyphering	2.8	6.1	8.6	18.2	28.8
De-cyphering	2.7	6.0	8.5	18.4	28.0

Table 3: Table 3. Time, in microseconds, in an AMD K7-700 (per round, using 100000 rounds)

	DES 64,64	DES 64,128	T-DES 128,64	T-DES 128,128	Rijndael 128,128
Cyphering	3.4	6.9	11.2	24.5	35.8
De-cyphering	3.5	7.0	11.1	24.4	36.0

Table 4: Table 4. Time, in miliseconds, in 8051 microcontroller (per round, using 100 rounds)

As it can be seen, Rijndael is a little slower than T-DES, for the same configurarion, but difference is not big enough to overcome the advantages previously exposed about this same algorithm. It can also be seen the difference in computation time needed for an standard PC, and for a microcontroller used with smart cards. The main difference among these two tables comes from the fact that the microcontroller is based on 8 bits, running below 5MHz, and having very restricted memory space.

These results show that Rijndael have a computer cost of the same order than the one needed by T-DES, and that it can also be implemented in commercial smart cards, although obtaining computation times larger than the ones obtained in a commercial PC.

Conclusions

In this paper, the structure and design of Rijndael cipher (new AES) have been analyzed, remarking its main advantages and limitations, as well as its similarities and dissimilarities with DES and T-DES. Thus, the fact that the new cipher and its inverse use different components, which practically eliminates the possibility for weak and semi-weak keys, is one of the principal advantages of this new cipher algorithm, compared to DES. Also, the nonlinearity of the key expansion, which practically eliminates the possibility of equivalent keys, is another big advantage. Finally, a performance comparison among new AES, DES and T-DES for differents microcontrollers has been carried out, showing that new AES have a computer cost of the same order than the one needed by T-DES.

References

[1] E. Biham and A. Shamir, *Differential cryptanalysis of DES-like cryptosystems*, Journal of Cryptology, vol. 4, no. 1, 1991, pp. 3-72.

[2] J. Daemen and V. Rijmen, *AES Proposal: Rijndael*, version 2, 1999. Available from URL: *http://www.esat.kuleuven.ac.be/~rijmen/rijndael.*

[3] J. Daemen and V. Rijmen, *Answer to "New observations on Rijndael"*, version of August 11, 2000.

[4] J. Daemen, L.R. Knudsen and V. Rijmen, The Block Cipher Square, *Fast Software Encryption - FSE'97*, Springer Verlag, Haifa, Israel, pp. 149-165, January 1997.

[5] J. Daemen, *Cipher and hash function design strategies based on linear and differential cryptanalysis*, Doctoral Dissertation, March 1995, K.U. Leuven.

[6] M. Hellman and S. Langford, *Differential-Linear Cryptanalysis*, Advances in Cryptology, Proc. Cryto'94, LNCS-839, Sringer-Verlag, 1994, pp. 26-39.

[7] N. Koblitz, *A Course in Number Theory and Cryptography*, Springer-Verlag New York Inc., 1987.

[8] M. Matsui, *Linear Cryptanalysis method for DES cipher*, Advances in Cryptology, Proc. Eurocrypt'93, LNCS 765, Springer-Verlag, 1994, pp. 386-397.

[9] S. Murphy and M. Robshaw, *New observations on Rijndael*, version of August 7, 2000. Available from URL: *http://isg.rhbnc.ac.uk/mrobshaw.*

[10] National Institute of Standards and Technology, *Data Encryption Standard*, FIPS 46-2, 1993.

[11] B. Schneier and D. Whiting, *A Performance Comparison of the Five AES Finalist*, 15 March 2000.