

## Assignment Three

ITN556 - Advance Topics in Cryptology

**Title:** Study of MARS and RC6 Block Ciphers

Unit Co-coordinator,  
Prof. Ed Dawson.

By,  
Vijayakrishnan Pasupathinathan (N2810743)  
[v.krishnanp@student.qut.edu.au](mailto:v.krishnanp@student.qut.edu.au)

### Abstract

This document describes block ciphers **MARS** and **RC6<sup>TM</sup>**. These block ciphers were submitted to NIST to be considered for AES. This paper discusses cipher structure and algorithm as well as the criteria's that were taken into consideration for design of the ciphers and also computational efficiency of these ciphers. The document also looks into type of attacks that were carried out on these ciphers.

## 1. Introduction

Symmetric key Block ciphers are fundamental to cryptographic systems. They are used in a wide range of applications from e-mail messages to ATM's to secure distribution of Internet contents. The previous standard for block ciphers was DES, which was developed by IBM. The limitations on block size and key size in DES made it necessary to design a new block ciphers to replace the aging DES. In Jan 1997 the National Institute of Standards and Technology (NIST) called for papers to develop Advance Encryption standard (AES) which would replace DES. [1]

MARS and RC6 were submitted to NIST. IBM developed MARS as its cipher for consideration to AES [2] and RSA developed RC6 as its candidate. [3] This document looks at these two ciphers.

## 2. MARS

MARS is a symmetric key block cipher with a block size of 128 bits and a variable key size from 128 to 400 bits. These were chosen to meet the requirements of AES. The designers estimated that MARS would offer more security than triple-DES. MARS is designed to be used in computers of today. It uses a mixed cipher structure; the top and bottom rounds were designed differently than the bottom ones. All operation in MARS is applied to 32-bit words.

### 2.1 Structure of MARS

The MARS structure can be considered as six different layers through which a plaintext block must pass to become a cipher text block: [4]

1. Pre-Whitening Layer: The plaintext has 128 bits of key material added to its words modulo  $2^{32}$ .
2. Forward Mixing Layer: Eight rounds of un-keyed mixing operations making extensive use of the MARS S-box.
3. Forward Core Layer: Eight rounds of keyed unbalanced Feistel cipher, using a combination of S-box lookups, multiplications, data-dependent rotations, additions, and xors to resist cryptanalytic attack.
4. Backward Core Layer: Eight rounds of keyed unbalanced Feistel cipher, using a combination of S-box lookups, multiplications, data-dependent rotations, additions, and xors to resist cryptanalytic attack.
5. Backward Mixing Layer: Eight rounds of un-keyed mixing operations making extensive use of the MARS S-box.
6. Post-Whitening Layer: The block has 128 bits of key material subtracted from its words modulo  $2^{32}$ .

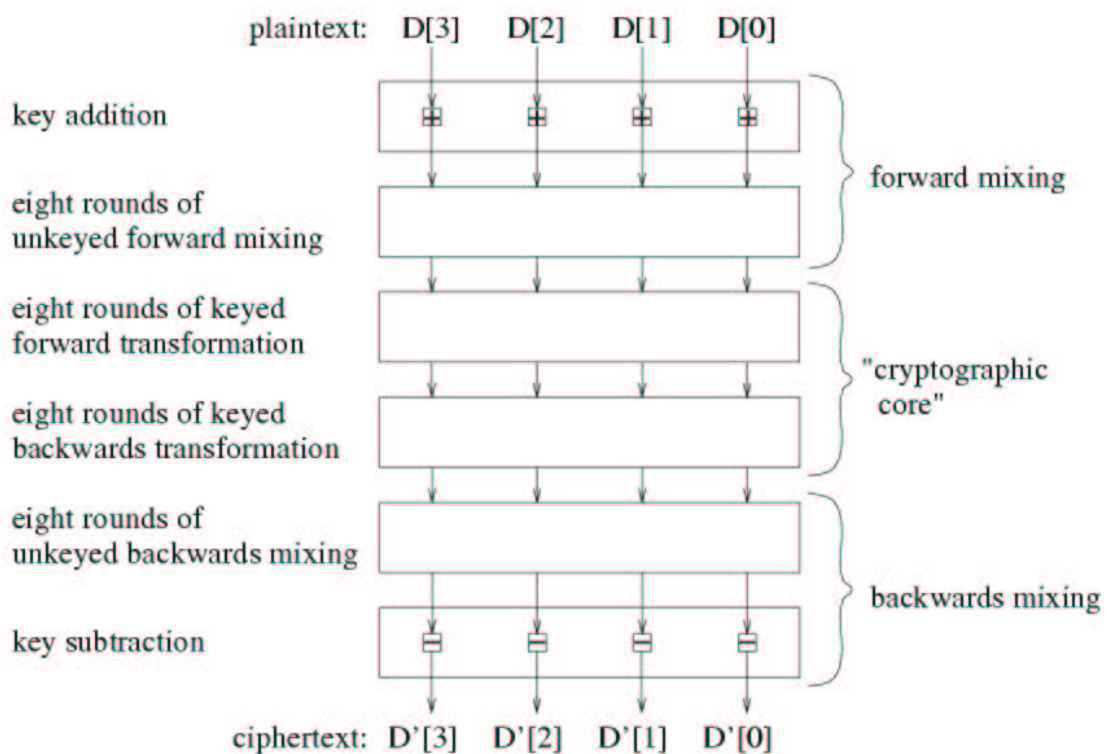


Figure 1: High level structure of the cipher [2]

## 2.2 Mars Cipher Operations

MARS algorithm uses a big variety of different operations:

**Additions, subtractions and xors:** These operations are used to mix data and key values together. **Table look-up:** Similar to the S-boxes in DES, in MARS cipher a table look-up is used. It uses a single table of 512 32-bit words, simply called S-box. **Fixed rotations, Data-dependent rotations:** Data dependent rotations may lead to differential weaknesses. This problem is solved in MARS by combining these rotations with multiplications. **Multiplications:** All multiplications in MARS are modulo  $2^{32}$  which suit most modern computer architectures. MARS algorithms uses 16 multiplications per block..

## 2.3 Mars: Encryption

**Forward Mixing:** The structure of the forward mixing stage is showed in figure 2. First the data words are added with 4 key words. Then 8 rounds of unkeyed type-3 Feistel mixing are performed. Type-3 Feistel mixing means there is always one word that is used

to modify the three other words. The modification is done without key information by using S-boxes. The output of the S-boxes is added or xored with the other words. Additional to the S-box mixing several shifts are performed.

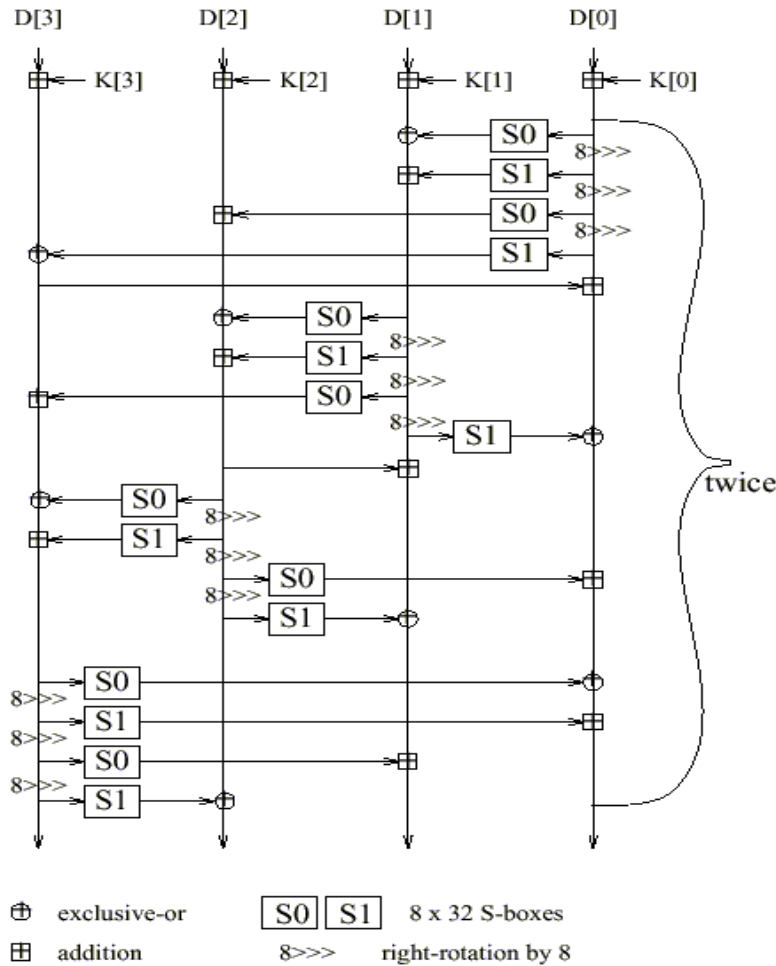


Figure 2: Forward Mixing [2]

### *Pseudo Code*

```

for i = 0 to 3 do
  D[i] = D[i] + K[i]
  for i = 0 to 7 do
    D[1] = D[1]  $\oplus$  S0[ low byte of D[0] ]
    D[1] = D[1] + S1[ 2nd byte of D[0] ]
    D[2] = D[2] + S0[ 3rd byte of D[0] ]
    D[3] = D[3]  $\oplus$  S1[ high byte of D[0] ]
    D[0] = D[0] > 24
  if i = 0 or 4 then
  
```

```

    D[0] = D[0] + D[3]
    if i = 1 or 5 then
        D[0] = D[0] + D[1]
    (D[3];D[2];D[1];D[0]) ← (D[0];D[3];D[2];D[1])
end-for

```

### Cryptographic core

The cryptographic core (figure 3) is a type-3 Feistel network. But here the algorithm uses a keyed E-function (figure 4) instead of the unkeyed S-boxes as in the forward mixing stage. The output of the E-function is also added or xored with the other words. There are total 16 rounds in the core. Eight forward plus eight backward rounds.

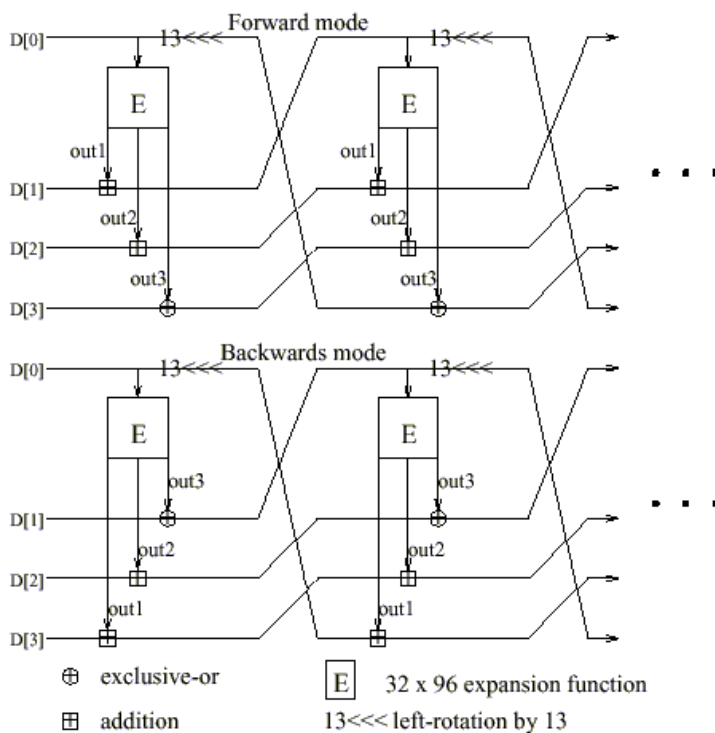


Figure 4: Type-3 Feistel network of the main keyed transformation

### E- function:

The E-function (Figure 4) is a combination of different operations that mix two key word with the input word. It also contains an S-box lookup. The k' (odd) means that this are special made-up key words with special properties. The E-function is one of the best-designed parts of the MARS algorithm. The different functions are combined in a way that maximizes the advantage of each.

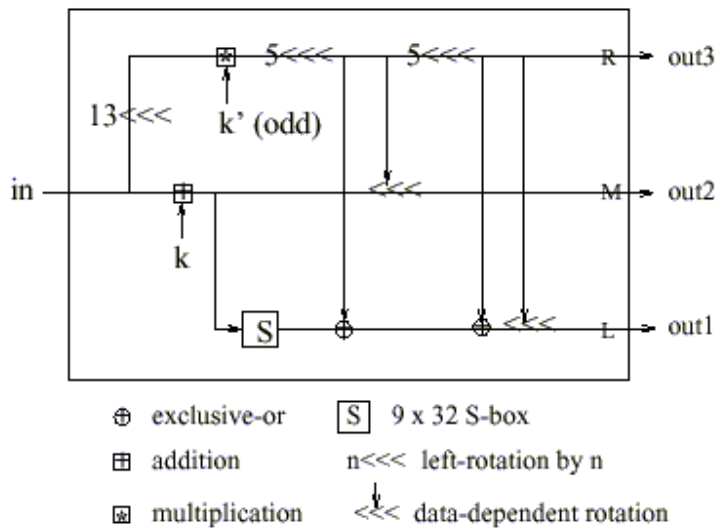


Figure 4: The E-function of the main keyed transformation

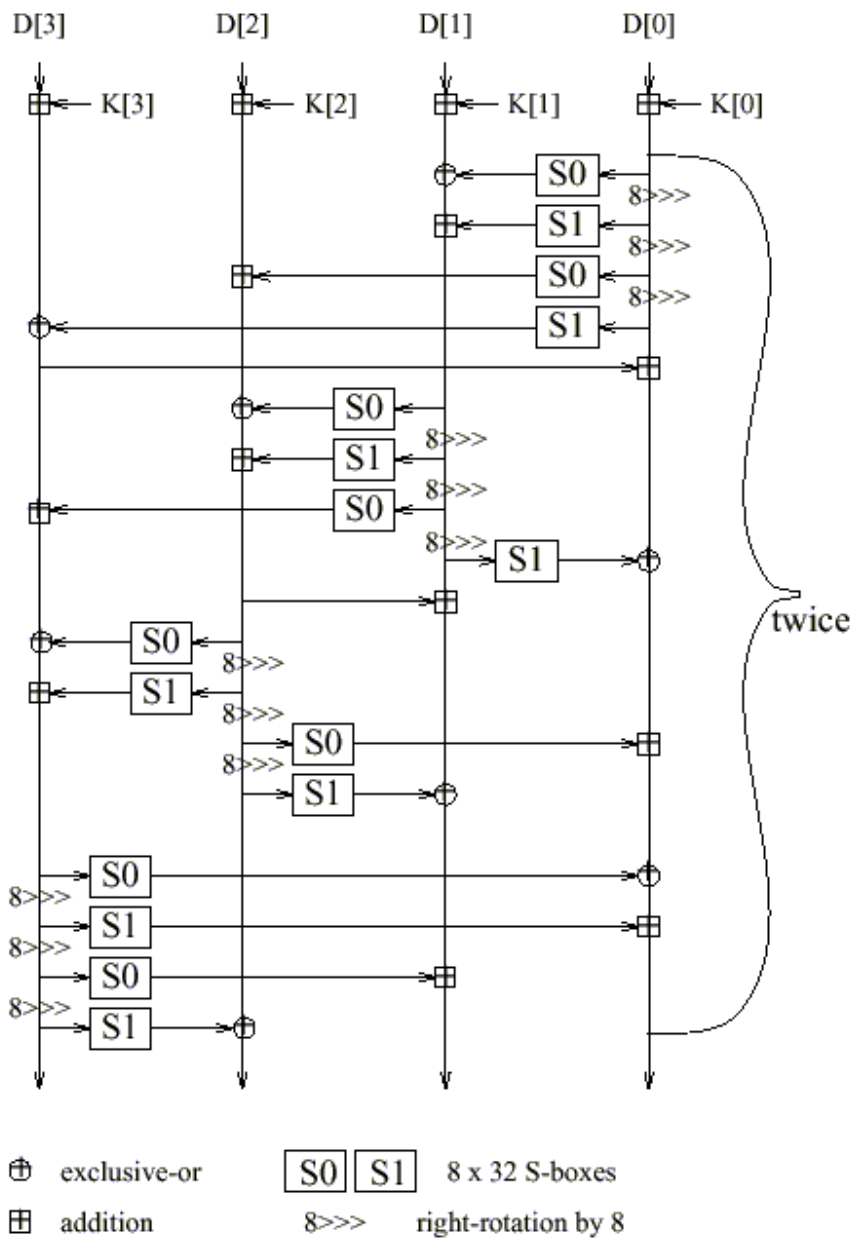
### *Pseudocode*

```

for i = 0 to 15 do
  (out1; out2; out3) = E-function(D[0];K[2i + 4];K[2i + 5])
  D[0] = D[0] <<< 13
  D[2] = D[2] + out2
  if i < 8 then
    D[1] = D[1] + out1
    D[3] = D[3] ⊕ out3
  else
    D[3] = D[3] + out1
    D[1] = D[1] ⊕ out3
  end-if
  (D[3];D[2];D[1];D[0]) ← (D[0];D[3];D[2];D[1])
end-for
  
```

### **Backward mixing**

The structure of the backward mixing showed in figure 5 is very similar to the one of the forward mixing. It consists also of 8 rounds of a unkeyed type-3 Feistel network, followed by a key subtraction step.



**Pseudo Code**

```

for i = 0 to 7 do
  if i = 2 or 6 then
    D[0] = D[0] ,D[3]
    if i =3 or 7 then
      D[0] = D[0] ,D[1]
      D[1] = D[1] ⊕ S1 [low byte of D[0] ]
      D[2] = D[2] - S0[ high byte of D[0] ]
      D[3] = D[3] - S1[ 3rd byte of D[0] ]
      D[3] = D[3] ⊕ S0[ 2nd byte of D[0] [
  
```

```

        D[0] = D[0] <<< 24
        [D[3];D[2];D[1];D[0]] ← [D[0];D[3];D[2];D[1]]
    endfor
    for i = 0 to 3 do
        D[i] = D[i] , K[36 + i]
    
```

### **Key Expansion**

MARS algorithm has a variable key length. The interval of the key length is either from 128 to 448 bit or 128 to 1248 bit with some restrictions. Internal the algorithm works with 40 key words, which is equal to 1280 bit. But 32 of these bits are constant 1, therefore the actual internal key is just 1248 bit. There are some further restrictions to that key. The same key words with the two bits constant 1 should not contain 10 consecutive 0's or 1's. The reasons for this criterion are that these key words are used for the multiplication in the E-function and key words without these properties would lead to weak keys for differential attacks. In the submission for AES IBM suggests using just 128 to 448 bit keys that are expanded to a 1280 bit key with the mentioned properties. The key expansion routine uses the same operations (xor, shift and table look-up) as the encryption / decryption.

### 3. RC6

RC6 is a block cipher, which was submitted to NIST, to be considered as a replacement of DES. It is an extension of RC5. To meet the requirements of the AES, a block cipher must handle 128-bit input/output blocks. RC6 is designed to use 4 32-bit registers. RC6 is more accurately specified as RC6-w/r/b where the word size is  $w$  bits, encryption consists of a nonnegative number of round  $r$ , and  $b$  denotes the length of the encryption key in bytes.

#### 3.1 RC6 cipher Operations

RC6-w/r/b operates on units of four  $w$ -bit words using the following six basic operations. The base-two logarithm of  $w$  will be denoted by  $\lg w$ .

- $a + b$  integer addition modulo  $2^w$
- $a - b$  integer subtraction modulo  $2^w$
- $a \oplus b$  bit wise exclusive-or of  $w$ -bit words
- $a \times b$  integer multiplication modulo  $2^w$
- $a \lll b$  rotate the  $w$ -bit word  $a$  to the left by the amount given by the least significant  $\lg w$  bits of  $b$
- $a \ggg b$  rotate the  $w$ -bit word  $a$  to the right by the amount given by the least significant  $\lg w$  bits of  $b$

#### 3.2 RC6 Encryption

RC6 works with four  $w$ -bit registers  $A$ ;  $B$ ;  $C$ ;  $D$  which contain the initial input plaintext as well as the output ciphertext at the end of encryption. The first byte of plaintext or ciphertext is placed in the least-significant byte of  $A$ ; the last byte of plaintext or ciphertext is placed into the most-significant byte of  $D$ .

$(A; B; C; D) = (B; C; D; A)$  to mean the parallel assignment of values on the right to registers on the left.

##### *Pseudo Code*

```
B = B + S[0]
D = D + S[1]
for i = 1 to r do
{
    t = (B × (2B + 1)) <<<lg w
    u = (D × (2D + 1)) <<<lg w
    A = ((A ⊕ t) <<<u) + S[2i]
```

$$C = ((C \oplus u) \lll t) + S[2i + 1]$$

$$(A; B; C; D) = (B; C; D; A)$$

$$\}$$

$$A = A + S[2r + 2]$$

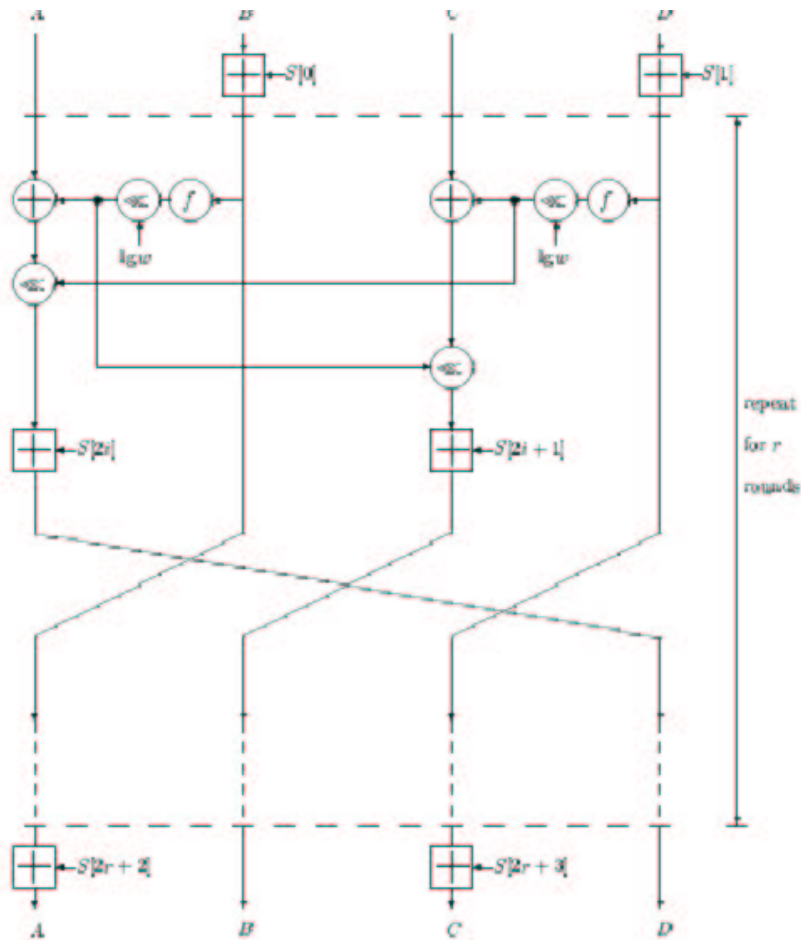
$$C = C + S[2r + 3]$$


Figure 5: Encryption with RC6 [3]

### 3.3 Key schedule

The user supplies a key of  $b$  bytes, where  $0 < b < 255$ . From this key,  $2r + 4$  words ( $w$  bits each) are derived and stored in the array  $S[0 \dots 2r + 3]$ . This array is used in both encryption and decryption.

## 4 Mars v/s RC6

### 4.1 Based on Performance

The table below shows the time taken by the encryption process in RC6 and MARS ciphers on three different platforms

	<b>MARS</b>	<b>RC6</b>
<b>PII 200Mhz (32bit word)</b>	9.97 MB/s 306c	13.68MB/s 223c
<b>Ultra SparcIII (64bit word)</b>	5.09MB/s 810c	3.54MB/s 1164c
<b>6805 , 8 bit word</b>	358 Kc	106 Kc

Mhz → Mega Hertz

MB/s → MegaBytes per second

c → CPU cycles

Kc → Kilo cycles

### 4.2 Based on Security

**RC6:** Best attack on RC6 [3] appears to be exhaustive search for user supplied encryption key. It is also noted that the data required to mount differential and linear cryptanalysis exceeds the available data and there are no known examples what might be termed a weak key.

**MARS:** Two kinds of attack have been carried out on MARS[4]

*Meet in the middle attack:* Full mixing plus 5 core rounds (21 rounds) which requires  $2^{232}$  half encryptions and  $2^{236}$  bytes of memory, 8 known plaintext

*With reduced round:* 8 rounds –  $2^{68}$  partial decryptions  $2^{29}$  bytes of memeory,  $2^{25}$  chosen plaintext

The best result is to break 21 out of 32 rounds.

## 5. Conclusion

Both MARS and RC6 are efficient block ciphers. There are no weaknesses in both cipher algorithms. Mars has a larger availability of key length than RC6 and can be easily expandable to more than 128 bits. Both were excellent candidate for AES. The reason Rijndael might have been chosen for AES over these two ciphers, might be because of performance, support for hardware and flexibility.

## Reference:

- [1] "ANNOUNCING DEVELOPMENT OF A FEDERAL INFORMATION PROCESSING STANDARD FOR ADVANCED ENCRYPTION STANDARD", NIST, January 1997.  
[http://csrc.nist.gov/encryption/aes/pre-round1/aes\\_9701.txt](http://csrc.nist.gov/encryption/aes/pre-round1/aes_9701.txt)
- [2] "MARS A CANDIDATE FOR AES", IBM, September 1999.
- [3] "RC6 A BLOCK CIPHERS", RSA, August 1998.
- [4] "MARS Attacks! Preliminary Cryptanalysis of Reduced-Round MARS Variants", John Kelsey and Bruce Schneier.
- [5] "MARS encryption algorithm" Reto Galli