ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

IJIRSET

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

Study, Simulation and Analysis of Advanced Encryption Standard (AES) Algorithm

B.Sujitha, Dr.B.Karthikeyan

IInd Year – M.E [VLSI], Srinivasan Engineering College, Perambalur, Anna University, Chennai, India

Professor, Dept. of ECE, Srinivasan Engineering College, Perambalur,Anna University, Chennai, India

ABSTRACT: The aim of this project is to attain an encrypted pre configured logic function for FPGA.Such a function should have a security values such as resistant to Differential Power Analysis (DPA) Attacks and it should also be free from cloning or duplication.AES-Rijndael Algorithm which is a block Cipher is used for encryption in some of cryptographic devices such as smart cards, RFIDs can be attacked with the help of the amount of data. The FPGA is processing and the operation it performs over those data by measuring the time taken for the process and also the power consumed during the process. This cryptanalytic technique is called as Side Channel Attacks (SCA).Among such attacks we are focussing on the power consumed by the process and terming it as Differential Power Analysis (DPA) attack.Here our objective is to take preventive steps to make our data and its operation independent of power it consumes such that possibility of DPA attacks can be greatly reduced. Further, The Secure Hashing Algorithm is applied in to an AES, it takes an arbitrary bit string as input and returns a fixed length string as output. The hash value is nearly impossible to derive the original input number without knowing the data used to create the hash value.

Keywords : AES, Rijndael, Secure Hashing Algorithm, Block cipher, encryption.

1. INTRODUCTION

Cryptography is an art of securing information in an insecure environment to provide the confidentiality, integrity, authentication and non-repudiation.Confidentiality is the process of maintaining the secrecy of information.Confidentiality can be achieved by encryption and decryption. Encryption is the process of converting the understandable plain text to an unclear cipher text with some mathematical calculations along with some shifting and rotating operations with or without a key. The vice-versa of encryption is decryption here it transforms the unclear cipher text to an understandable format. Cryptanalysis is the process of breaking encryption/decryption system without knowing the secret key. Motivation of decryption is to hack or alter the information by unauthorized person.

The classical ciphers are easy to break, therefore modern ciphers were introduced in modern era, which uses shift, substitution and some mathematical operations for encryption and decryption.

Modern ciphers are broadly classified into stream a cipher (which encrypts the inputs bit by bit) and block ciphers (which encrypts a block of bits at a time). Here we are focusing on block ciphers.

The block cipher transforms a block of plain text to a block of cipher text. The simple design, easy implementation and resistance to known attacks, characteristics are also favours block ciphers. Many block cipher algorithms were proposed. Among which Data Encryption Standard (DES) was considered to be the most dominant till 1990s. The latter years due to technology development and new ideas in the field of mathematics made the attacks easier on DES. These developments cleared that there is a need of improved advanced standard to provide additional security to the information. Accordingly the National Institute of Standards and Technology (NIST) called for the proposal of new block cipher algorithm. Many groups from various part of the globe submitted their algorithm. By analysing various security parameters and other characteristics, RIJNDAEL algorithm developed by two Belgian cryptographers Joan Daemen and Vincent Rijmen was crowned as new AES algorithm in the year 2001 defeating MARS, RC6, Twofish and Serpent.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

2. AES-RIJNDAEL ALGORITHM

Rijndael is an iterated block cipher that allows for a 128-bit data block, and a variable key block size of 128, 192, or 256 bits. (The original Rijndael submission allowed for the possibility of variable data block sizes as well of 128, 192, and 256 bits, but the final Standard was published with a single data size.) The simple round function of Rijndael replaced the iterative structure of DES and other competitors of AES.

2.1 Preliminaries

Three values are input ahead of time:

Nb –	the data size in words (32 bits)
Nk –	the key size in words
Nr –	the number of rounds to be performed

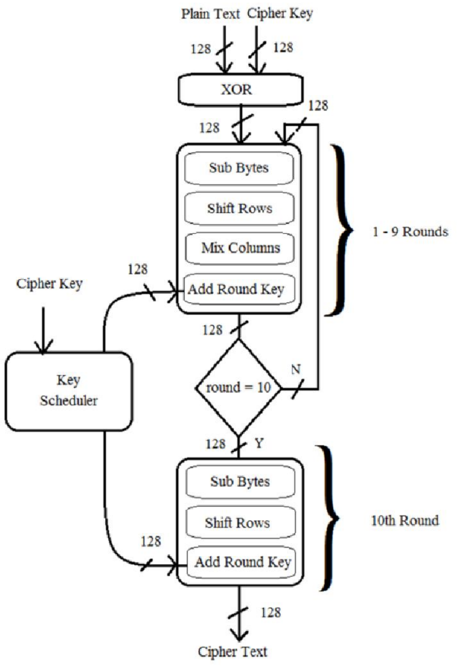

Fig.1.Encryption Process flow of AES Algorithm

The input data of 128 bits is put into an array called State. The State and the KeyState are simply a 4 x Nb and a 4 x Nk array of bytes, input[n] → State[i,j] where i and j correspond to:

i = n mod 4
j = lower bound(n / 4)

The number of rounds performed is a function of the key length as follows.

TABLE I . AES PARAMETER

Key length	Nk:words	Nb:words	Nr:Number of Rounds
128 bits	4	4	10
192 bits	6	4	12
256 bits	8	4	14

The original cipher key that is input to the Rijndael algorithm is used in its own key schedule that can be performed separate from the actual encryption of the cipher text.

2.2 The Round Transformations

The inputs to every round are the State and Roundkey arrays. The Roundkey array is a permutation on the original cipher key.

Each generic round is composed of four functions:

```
Round (State,Roundkey)
    {
    Bytesub (State)
    Shiftrow (State)
    Mixcolumn (State)
    AddRoundkey (State)
    }
```

And the final round is composed of:

```
FinalRound (State, Roundkey)
    {
    Bytesub (State)
    Shiftrow (State)
    AddRoundkey (State)
    }
```

2.2.1 Subbytes Transformations

Subbytes works on bytes or words, so the received 128 bits will be grouped to 16 words of length 8 bits.Each word is mapped to the rows and columns of the content of constant S-box matrix and the value is replaced.Similarly all the 16 words are replaced and arranged in a form of matrix by filling the columns first.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

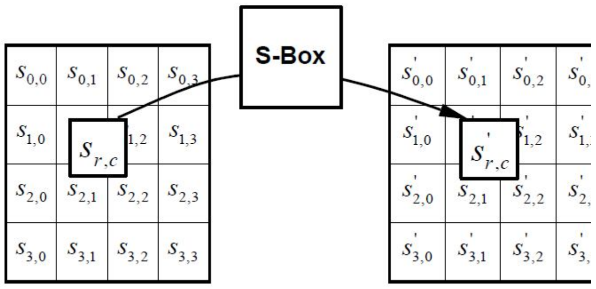

Fig.2.Subbytes( )

2.2.2 Shiftrows Transformations

This transformation is a cyclic shift to the right performed on the last three rows of the State matrix, while the first row is not shifted at all. The shifting sequence is as follow:

row 1 is shifted by one, row 2 is shifted by two, and row 3 is shifted by three.

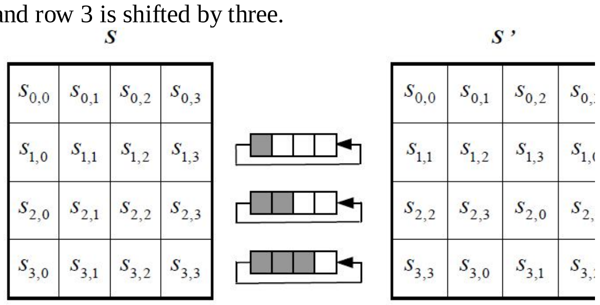

Fig.3.Shiftrows ( )

The inverse of the ShiftRow operation used for decryption is simply another right shift operation that restores the columns to their original positions.

2.2.3 Mixcolumns Transformations

The MixColumn operation treats each column as a four term polynomial. Each column is multiplied modulo $x^4 + 1$ with a specifically chosen four term polynomial *c(x)*:

$$c(x) = \text{'03'}x^3 + \text{'01'}x^2 + \text{'01'}x + \text{'02'} \quad (1)$$

This Multiplication can be written in matrix as follow:

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

The operation of mixcolumn can be easily illustrated in the following figure.

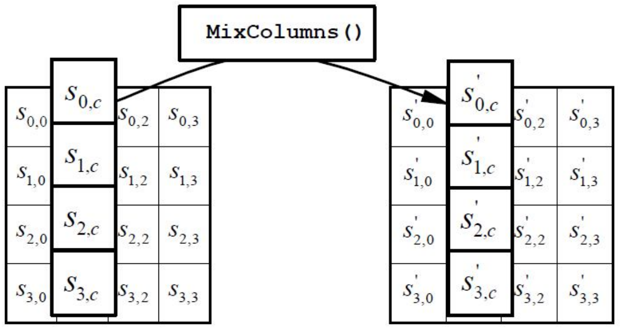

Fig.4.Mixcolumn ( )

The Inverse of MixColumn, which is used for decryption, is similar to the normal version. For decryption, the multiplicative inverse of c(x) is taken which gives us the new polynomial d(x).

$$(\text{'03'}x^3 + \text{'01'}x^2 + \text{'01'}x + \text{'02'}) \bullet d(x) = \text{'01'} \quad (2)$$

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

The polynomial c(x) is co-prime to $x^4 + 1$, and is therefore invertible, which is why c(x) was chosen in the first place. The Inverse MixColumn function simply performs the multiplication modulo $x^4 + 1$ with the new d(x) polynomial.

2.2.4 AddRoundkey Transformations

The AddRoundKey(State, RoundKey) function is a simple bit-wise XOR operation between two 4 x 4 byte matrices. The two matrices are the State and the

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

RoundKey. The RoundKey is a result of the Key Expansion Schedule performed on the original cipher key, and will always produce a 4 x 4 matrix as well.

The original submission allowed for the RoundKey to have the same number of columns as the State matrix, but because only 128 bit data blocks are used in the FIPS Standard the State and RoundKey both have a fixed four-column structure. The operation of AddRoundKey is as follows:

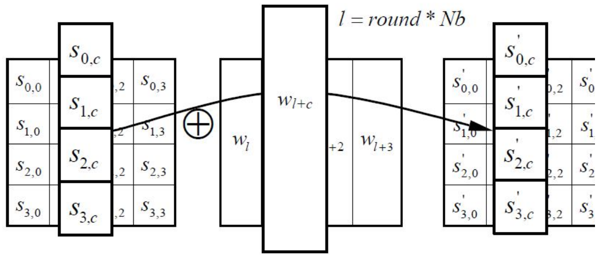

Fig.5.AddRoundkey( )

The inverse of the RoundKey addition is the same operation as the AddRoundKey for encryption. The XOR allows it to be its own inverse.

2.3 Key Schedule

Each RoundKey is derived from the original cipher key through the Key Schedule. The cipher Key is expanded into a large array of words, the size of which is subject to the specified key length.

TABLE II.KEY SCHEDULE GENERATION

Cipher Key(Nb)	Expanded Key(Words)
4	44
6	52
8	60

3. KECCAK-256(SECURE HASHING ALGORITHM)

Keccack was selected as a finalist, mainly due to its high security margin, its high throughput and throughput-to-ratio and the simplicity of its design. Keccak uses the sponge construction model. In a sponge construction, the internal stage registers are divided into two portions. A sponge construction has two phases, absorb and squeeze. During the absorption phase, the input message will be XORed with the data stored inside the first portion. The XORed value will be updated along with the data from the second portion of the internal state registers through the round function. During the squeeze phase, the data stored inside the first portion will be used as a portion of the output. New output values will be present inside those registers as they were updated through the round function in each cycle. Unlike the structures mentioned previously which truncate the internal state registers for output, a sponge construction can accommodate an output of any arbitrary size by updating its internal state registers.

The Keccak has a round function module and a round constant generation module. The round function module of Keccak is a permutation-substitution network using S-Boxes that are 5-bits wide. The round generation module generates round constants based on a 5-bit round number.

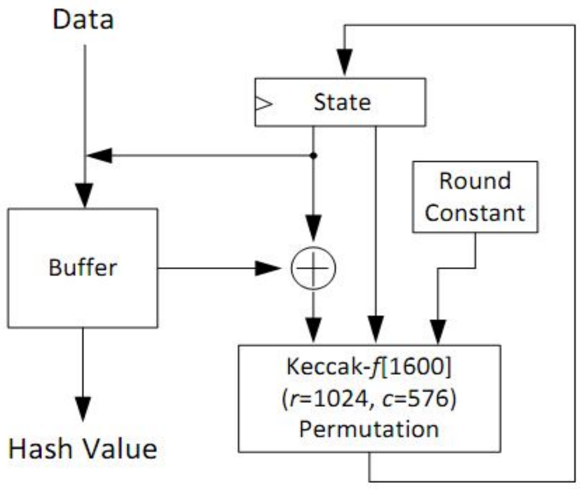

Fig.6.Structure of the Keccak-256 core

3.1 Performance

Keccak is an average performer in software across most platforms for long messages. In constrained environments, Keccak is below-average in performance, average in ROM usage, and below-average in RAM usage. In hardware, Keccak is among the top performers in throughput-to-area ratio (especially for the 256-bit variant). The area usage is average, but the throughput is outstanding, because of the very high level of parallelization inherent in the design.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

4. AES COMBINED WITH HASHING ALGORITHM

Hash value is generated to an description of the AES.It has key length as SHA-256.AES-hash is a secure hash mode, rijndael is used in 256 bit key and 256 block mode.The hash value is that is nearly impossible to derive the original input data without knowing the data used to create the hash value.

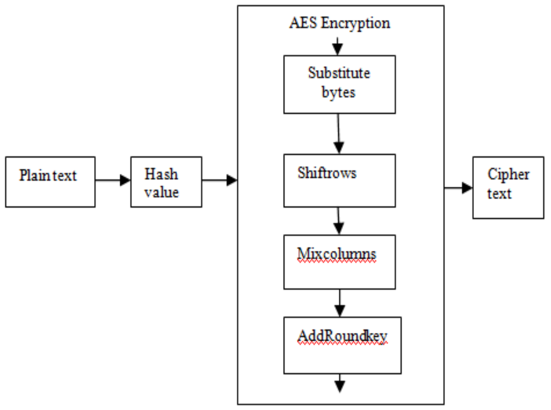

Fig.7.AES-Hash for Secure Cryptographic Algorithm

In this proposed methodology hash value is generated using a keccak algorithm, And then the Generated hash key is given as the input to the AES Encryption algorithm. It also provide a high security and throughput.The throughput of a design is calculated as follows

$$TP=\frac{\text{Block size}*\text{Maxfreq}}{\text{Latency}} \qquad (4)$$

The block size is the amount of data the hashing algorithm will process at a time. It is fixed by the algorithm and is not a design parameter for the hardware designer. The latency shown here is the core latency, which is the number of cycles it takes to hash a message. The maximum throughput is another important metric people focus on in the SHA-3 project because it indicates how much data a design can process over a fix amount of time. Maximum throughput is important because this number weighs block size, a characteristic of the algorithm; maximum frequency, a characteristic of the hardware design performance, and latency; a characteristic of the hardware design architecture

5.

5. SIMULATION RESULTS

The design has been coded by verilog HDL.All the results are synthesized and simulated using Xilinx are shown in figure.

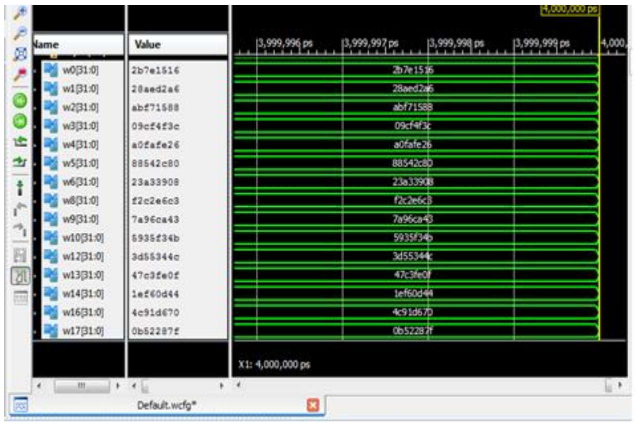
```
Name        Value
w0[31:0]    2b7e1516
w1[31:0]    28aed2a6
w2[31:0]    abf71588
w3[31:0]    09cf4f3c
w4[31:0]    a0fafe26
w5[31:0]    88542c80
w6[31:0]    23a33908
w8[31:0]    f2c2e6c3
w9[31:0]    7a96ca43
w10[31:0]   5935f34b
w12[31:0]   3d55344c
w13[31:0]   47c3fe0f
w14[31:0]   1ef60d44
w16[31:0]   4c91d670
w17[31:0]   0b52287f
```

Fig.8.Timing simulation of AES Key Expansion

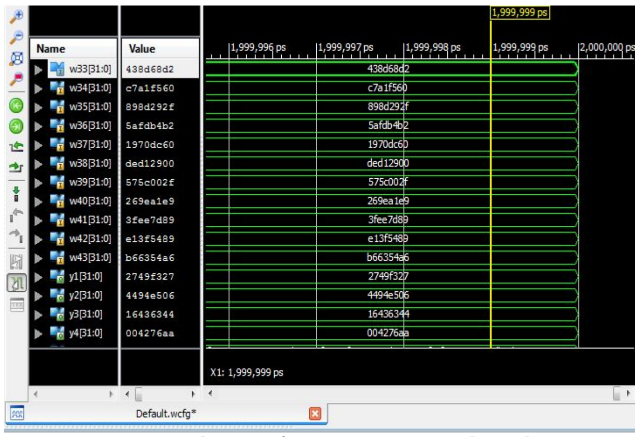
```
Name        Value
w33[31:0]   438d68d2
w34[31:0]   c7a1f560
w35[31:0]   898d292f
w36[31:0]   5afdb4b2
w37[31:0]   1970dc60
w38[31:0]   ded12900
w39[31:0]   575c002f
w40[31:0]   269ea1e9
w41[31:0]   3fee7d89
w42[31:0]   e13f5489
w43[31:0]   b66354a6
y1[31:0]    2749f327
y2[31:0]    4494e506
y3[31:0]    16436344
y4[31:0]    004276aa
```

Fig.9.Timing Simulation of AES encryption algorithm

6. CONCLUSION

Nowadays security is the most important one. We provide the better security using a cryptographic algorithm. To encrypt the plain text using a key to produce cipher text. Rijindael algorithm has high

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

security margin compared to an other AES candidates. It has 32bit uniform across CPUs, easily fit on small smart cards. It should be efficient across all platform in hardware implementation. The designed core supports both encryption and decryption standards. Its functionality has been verified using simulation, by taking various inputs and is synthesized by using Xilinx ISE 12.1.

This work can be further extended to implement an AES finalists of MARS and Serpent Algorithm, Then compare the performance of above three Algorithm and propose a new novel algorithm.

REFERENCES

[1] Alan Kaminsky, Michael Kurdziel, Stanisław Radziszowski, (2010) "An overview of cryptanalysis research for the advanced encryption standard", IEEE, the 2010 military communications conference unclassified program cyber security and network management, pp. 1310.

[2] Bryan M.sobczyk,"AES implementations optimized for midrange FPGAs", scholarly paper.

[3] Daemon.J, Rijmen.V (March 1999), AES Proposal: Rijndael, Version 2.

[4] Elbirt AJ, Paar C," An FPGA Implementation and Performance Evaluation of the Serpent Block Cipher" Electrical and Computer Engineering Department, Worcester Polytechnic Institute 100 Institute Road Worcester, MA 01609, USA.

[5] Francois Xavier, Gael Rouvroy, Jean Jacques,"Effiecient Implementation of Rijndael Encryption in reconfigurable Hardware: improvements and design tradeoffs",Ucl crypto group,laboratorie de microelectronique,Belgium.

[6] Federal Information Processing Standards Publication 197 (November 2001), Announcing the Advanced Encryption Standard (AES) issued by the NIST.

[7] Matsui.M,(1994) "Linear cryptanalysis method for DES cipher", Eurocrupt, Lncs765, pp.386-397, Springer.

[8] Nechtvatal.J,Barker.E,Bassham.L,Burr.W,Dworkin.M,Foti.J,Ro back.E, Computer Security Division, Information Technology Laboratory, NIST, U.S. Department of Commerce, (October 2000) Report on the Development of the Advanced Encryption Standard (AES), printed by the NIST.

[9] Norbet Pramstaller,Elisabeth Oswald,Stefan Mangard,"A Masked AES ASIC Implementation".In proceedings of Institute of Applied Information Processing & communication(IAIK).

[10] Namin YU and Heys Howard.M,"A compact ASIC Implementation of the Advanced Encryption Standard with concurrent error detection",In proceedings of Electrical and Computer Engineering,Memorial university of Newfoundland.

[11] Rijmen.V,"Efficient Implementation of the Rijndael S-Box". Katholieke Universiteit Leuven, Dept. ESAT.

[12] Rudra.A, Dubey.P.K, Jutla.C.S, Kumar.V, Rao.J.R, and Rohatgi.P,(2001) "Efficient Implementation of Rijndael Encryption with composite Field Arithmetic."In proceedings of the cryptographic hardware and embedded systems conference, lecture notes in computer science vol2162, pp.171-185, Paris, France.

[13] Riaz.M and Heys H.M,"The FPGA Implementation of RC6 and CAST-256 Encryption algorithm".In proceedings of Electrical and Computer Engineering,Memorial university of Newfoundland.