COMPARATIVE ANALYSIS OF AES FINALIST ALGORITHMS AND LOW POWER METHODOLOGY FOR RC6 BLOCK CIPHER - A REVIEW

V. Tharun Deep[1], Dr. Venkata Siva Reddy. R[2]
Department of ECE
REVA ITM, Bangalore, India

Abstract: ***The Critical data which goes out into the environment has to be protected from cryptographic attacks, so the data has to be protected with the help of cryptographic algorithms. These algorithms secure the data by converting into cipher text for communication through internet and wireless methods. Low Power implementation of these algorithms is required because when these algorithms implemented in any embedded device could generate lot of power dissipation. This paper gives the comparison of different algorithms, principle design flow of RC6 algorithm and gives a brief overview on low power design methodology to the design.***
Index Terms: Cryptographic attacks, Cryptographic algorithms, Cipher text.

I. INTRODUCTION

These days with more and more technological advancements in the field of communication there is even more ever increasing threat to data which is being exposed in the environment of cryptographic attacks. With more advancement in internet applications there is a lot of critical data which is shared by the user and that has to be protected from illegal use by the hackers. As the growth of technology, communication through internet and wireless methods has become a revolutionary advancement of late [1]. So the never changing attribute which is of utmost prominence is the very basic necessity to protect the data from its unauthorized access of the information. With increasing inclination towards information security there was even more predilection in regard to security algorithms which acts as a barricade between the hacker and the critical data. As there were a lot of security algorithms which evolved out of the cause and were gaining its own appreciation at different fields of its use, the US government wanted to standardize a cryptographic algorithm which will be used universally by them called AES (Advanced Encryption Standards) This paper is in regard to different cryptographic algorithms which were shortlisted by NIST (National Institute of Standards and Technology) for the final round of selection for determining the AES. The different cryptographic algorithms that were shortlisted for the final round were

- Rijndael
- Twofish
- Serpent
- MARS
- RC6

This paper gives a detailed comparison on various evaluation criteria for the cryptographic algorithms and describes the implementation methodology of the cryptographic algorithm RC6 which contains the detailed briefing of Key Expansion Schedule, Encryption Process, and Decryption process. Also it gives the overview to obtain low power at different stages of synthesis flow.

II. COMPARATIVE ANALYSIS OF DIFFERENT BLOCK CIPHERS

A. RIJNDAEL:

Rijndael was developed by Joan Daemon and Vincent Rijimen which became the universally accepted AES.

Architecture:

- It is a symmetric key algorithm based on Fiestel structure.
- Key length can be variable either 128, 192 or 256 bits.
- It uses 128 bit plain text with variable 10, 12 or 14 rounds.
- It has no S-Boxes.
- Same algorithm is used in reverse for decryption.

Security:

It is mostly dependent on 256 variable key size, different attacks such as square attack, differential attack were practically not possible [2].

Encryption Speed:

Rijndael when implemented in Pentium II the raw encryption speed was very high compared to others with 2.54 u operations per cycle [3]. Assembly implementation is 44% slower than gcc implementations [4].

Hardware and software suitability:

It is the fastest in PA7000 and is more suited to Pentium II processor as it takes 5 less cycles in Pentium II for executing inner code [5].

Limitation:

Side channel attack is possible for this cipher [2].

B. TWOFISH:

Twofish was designed by Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, and Niels

Ferguson. The "extended Twofish team" who met to perform further cryptanalysis of Twofish and other AES contest entrants included Stefan Lucks, Tadayoshi Kohno, and Mike Stay.

Architecture:

- It is a symmetric key algorithm based on Fiestel structure.
- Key length can be variable either 128, 192 or 256 bits.
- It uses 128 bit plain text with 16 rounds.
- It has 4 S-Boxes.
- Same algorithm is used in reverse for decryption.

Security:
It has a complex key schedule, it is highly resistive to key related attacks. One half of the n-bit key is used as the actual encryption key and the other half of the n-bit key is used to modify the encryption algorithm [2].

Encryption Speed:
There is performance tradeoff between key-setup and encryption speed which is unique. The encryption time increases by less than 2600 clocks for 192 bit key and about 5200 for 256 bit key [4].

Hardware and software suitability:
It is used in all CPUs and it fits into smart cards, even those which have few registers. It fits in hardware in few gates.

Limitation:
It is susceptible to relative key attack and reduced round attack.

C. SERPENT:
It was designed by Ross Anderson, Eli Biham, and Lars Knudsen which was another finalist in the Advanced Encryption Standards (AES) Contest.

Architecture:

- It is a symmetric key algorithm based on Substitution permutation network structure.
- Key length can be variable either 128, 192 or 256 bits.
- It uses 128 bit plain text with 32 rounds.
- It has 8 S-Boxes.
- Same algorithm is used in reverse for decryption [6].

Security:
While designing extra assurance was allowed and hence 32 rounds of iteration was increased from 16 to make it protective from future discoveries of potential attacks [6]. There are attacks to break the 10^{th}, 11^{th} round, but still its not fully conquered.

Encryption Speed:
The encryption speed of this cipher was comparatively close to Rijndael. As it has 32 rounds therefore Rijndael is little faster.

Hardware and software suitability:
It is not suitable for small blocks or in small implementations as it has 32 rounds of iteration [7].

Limitation:
It is a bit slower due to the presence of 32 rounds and its complex to implement on small blocks [7].

D. MARS:
The design team of MARS included Don Coppersmith who was involved in creation of previous DES (Data Encryption Standard).

Architecture:

- It is a symmetric key algorithm based on Heterogeneous structure.
- Key length can be variable from 128 to 448 bits in multiples of 32-bit.
- It uses 128 bit plain text (Block size) with 32 rounds.
- It has a single S-Boxes.
- Same algorithm is used in reverse for decryption.

Security:
The Security is dependent on data rotations. It has 32 rounds of iteration so it offers improved security [2].

Encryption Speed:
Encryption speed which is number of cycles required for completion of the function in Rijndael is 1600 as compared to MARS which is 1276. The speed is less compared to AES [8].

Hardware and software suitability:
It is a complex algorithm and is difficult to implement in hardware [8].

Limitation:
Due to unique component behavior the simple round function of MARS is complex to analyze [2].

E. RC6:
It was designed by Ron Rivest, Matt Robshaw, Ray Sidney, and Yiqun Lisa Yin, it is a proprietary algorithm, patented by RSA Security.

Architecture:

- It is a private key algorithm based on Fiestel structure.
- Key length can be variable either 128, 192 or 256 bits.
- It uses 128 bit plain text with 20 rounds.
- It does not have S-Boxes.
- Same algorithm is used in reverse for decryption.

Security:
The security of RC6 is completely dependent on random series of output bits with 15 or less. A linear cryptanalysis attack can be launched for 16 rounds RC6, but requires 2^119 known plaintexts, which make the feasibility of such attack impossible. The RC6 algorithm is also strong against differential cryptanalysis, which worked with more than 12 rounds [2].

Encryption Speed:
When implemented in Pentium Pro processor 200MHz a performance of 250 clock cycles per block encrypted or decrypted was achieved, which makes it the fastest algorithm in target platforms. But it is 3 times slower in Pentium II processors [9].

Hardware and software suitability:
The algorithm is slower on many embedded platforms (Intel 386/486) and is therefore less attractive for low end 32 bit CPU's. It is best used in Pentium Pro processors where the module contains integer multiplication [9].

Limitation:
It is not suitable for 8-bit environments where integer multiplications instructions are not present [9]. In RC6, for a single class of weak keys, it is observed that full arbitrariness is not achieved for up to 17 rounds of the algorithm [2].

III. COMPARISON RESULTS

Based on the Architecture of these shortlisted algorithms it can be summed up through a table as shown below.

Characte-ristics / Algori--thm	**Type of Structure**	**Key length**	**Block size with number of rounds**	**S-Boxes**
Rijndael	Fiestel structure	Variable 128, 192 or 256 bits	128 bit with variable 10, 12 or 14 rounds	No
Twofish	Fiestel structure	Variable 128, 192 or 256 bits	128 bit with 16 rounds	Four
Serpent	Substitution permutation network structure	Variable 128, 192 or 256 bits	128 bit with 32 rounds	Eight
MARS	Heterogeneous structure	Variable 128 to 448 bits in multiples of 32-bit	128 bit with 32 rounds	One
RC6	Fiestel structure	Variable 128, 192 or 256 bits	128 bit with 20 rounds	No

Table 1. Comparison of architecture of different algorithms

On the other parameter with shows the security of all algorithms, it can be seen that security of Rijndael and RC6 are even and less compared to other algorithms of the group.

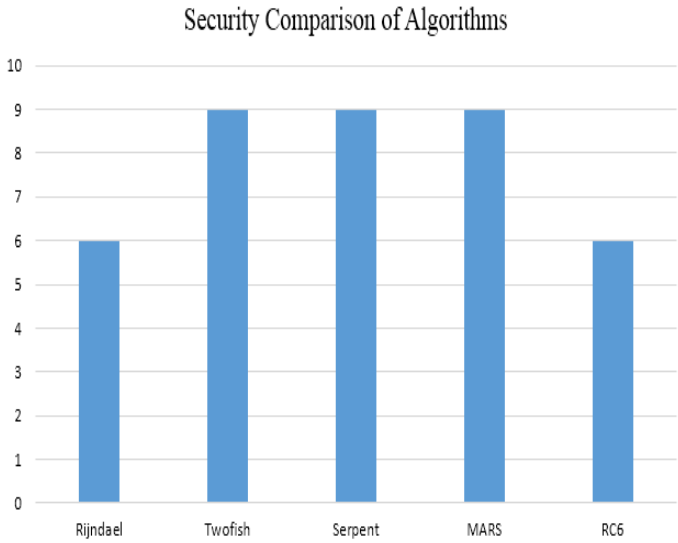

Figure 1. Security scale of different algorithms [19]

On encryption speed performance parameter is different for different platforms of use and on different implementations such as one algorithm could be of greater speed compared to another in C implementation and could be lagging behind on java implementation or assembly implementations. Therefore comparison on this parameter should be made from target platform and type of implementation of that specific algorithm.

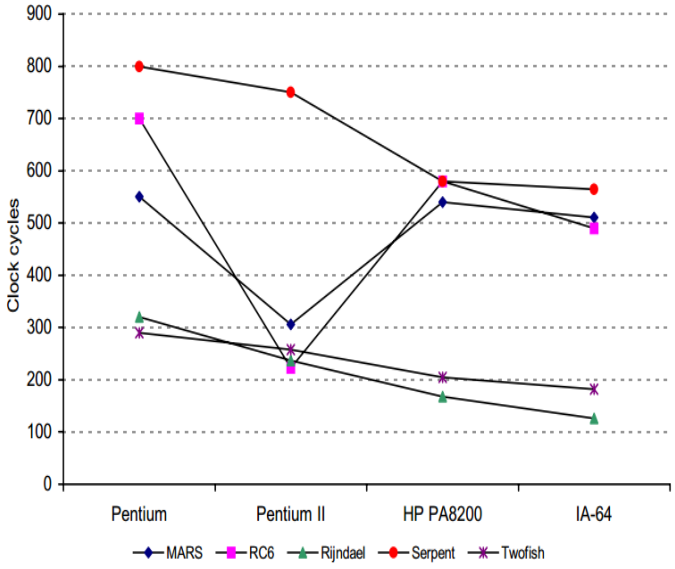

Figure 2 Encryption speeds for different algorithms in Assembly for 128 bit key [15]

IV. RC6 OVERVIEW

RC6 block cipher was designed by Ron Rivest, Matt Robshaw, Ray Sidney, and Yiqun Lisa Yinn from RSA Laboratories. It is an evolutionary advancement and improvement over the RC5 block cipher which was proposed and developed by the same company. RC6 makes use of data-dependent rotations and in order to be AES compliant the block cipher must handle 128-bit input and output blocks. Many additional features are included in RC6, which were not present in the RC5. The cipher uses four 32-bit registers for operations instead of two, as in RC5, which makes it possible to do two rotations per round rather than half round of RC5 algorithm. Integer multiplication is of 32 bit which makes it easy in most of the processors new and increases the diffusion achieved. This again leads to higher security, fewer rounds needed and an increased throughput. A version of RC6 is specified as RC6-w /r /b where w is the word size in bits, r is the number of rounds we encrypt and finally b is the length of the key we use to encrypt in 8-bit bytes. In the case of RC6 submitted for AES as a candidate the parameter are: w = 32, r = 20 and b =16, 24 or 32 for respectively 128, 196 and 256 bit keys.

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

V. PRINCIPLE

The main aspects in designing this algorithm are:

A. Key Expansion:

From the user given key with the length of b a number of internal subkeys are derived. If the key is not long enough it can be padded with zero bytes so as to achieve the required length. These subkeys are loaded into an array of c w words L[0, ..., c -1], that is the first byte is stored in L[0] and the high order byte, which can be zero padded if it is not of the required length, goes into L[c -1]. Now the subkeys are ready to be generated. The keys generated are stored into another array S [0, ..., 2r + 3].

The size of this array is 2r+4, and in the case of the AES candidate that is 2 × 20 + 4 = 44. RC6 uses, just as its predecessor the RC5, two ”magic” constants called Pw and Qw. Pw is derived from the binary expansion of e − 2, where e is the base of the natural logarithm and Qw is derived from the binary expansion of φ − 1, where φ (or Phi) is the Golden Ratio [9].

Key Expansion procedure is:

$S[0] = P_w$
for $i = 1$ **to** $2r + 3$ **do**
$S[i] = S[i-1] + Q_w$

$A = B\ i = j = 0$
$v = 3 \times max\{c, 2r+4\}$
for $s = 1$ **to** v **do**
{
$A = S[i] = (S[i] + A + B) \lll 3$
$B = L[j] = (L[j] + A + B) \lll (A + B)$
$i = (i + 1) \bmod (2r + 4)$
$j = (j + 1) \bmod c$
}

B. Encryption Process:

The encryption algorithm in RC6 is relatively simple. The plaintext, which is the input, is stored in four w -bit input registers called (A, B, C, D). Keys are being stored into an array S [0,...,2r + 3]. The ciphertext is the output and is being stored in (A, B, C, D) [9].The encryption algorithm consists of following steps:

RC6 begins with two initial steps:

- B is added with the subkey S [0]
- D is added with the subkey S [1]

Every round uses two subkeys, for each round i up to r the subkeys S [2i] and S[2i + 1] are being used, that is the first round uses S [2] and S [3]

A round can be described as:

- B and D are using the function f (x) = x(2x + 1) << log2w, which means that x(2x + 1) is left-shifted 5 bits (or log2w where w = 32)
- A = A ⊕ f(B) which is left-shifted with f(D) and added S[2i]
- C = C ⊕ f(D) which is left-shifted with f(B) and added S[2i + 1]
- The four quarters in the block are being rotated as: (A, B, C, D) = (B, C, D, A)

After the last round then:

- A is added with subkey S[2r + 2]
- C is added with subkey S[2r + 3]

It can be summarized as below:

$B = B + S[0]$
$D = D + S[1]$

for $i = 1$ **to** r **do**
{
$t = (B \times (2B + 1)) \lll \log_2 w$
$u = (D \times (2D + 1)) \lll \log_2 w$
$A = ((A \oplus t) \lll u) + S[2i]$
$C = ((C \oplus u) \lll t) + S[2i + 1]$
$(A, B, C, D) = (B, C, D, A)$
}

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

C. Decryption Process:
Decryption works in a similar way as encryption. The difference is that ciphertext is the input and plaintext is the output. The use of keys and rounds is the same as for encryption [9]. The decryption algorithm consists of following steps:
RC6 begins with two initial steps:

- C is subtracted with the subkey S [2r + 3]
- A is subtracted with the subkey S [2r + 2]

Every round uses two subkeys, for each round i downto 1 the subkeys S [2i] and S[2i + 1] are being used, that is the first round uses S [21] and S [20]

A round can be described as:

- The four quarters in the block are being rotated as: (A, B, C, D) = (D, A, B, C)
- D and B are using the same function as in described in the encryption, which is f (x) = x(2x+ 1) << log2w
- C = C − S[2i + 1] which is right-shifted with f(B) and the result is xor'ed with f(D)
- A = A − S[2i] which is right-shifted with f(D) and the result is xor'ed with f(B)

After the last round then:

- The subkey S[1] is subtracted D
- The subkey S[0] is subtracted B

It can be summarized as below:

$$
\begin{aligned}
&C = C - S[2r+3] \\
&A = A - S[2r+2] \\
&\\
&\textbf{for } i = r \textbf{ downto } 1 \textbf{ do} \\
&\{ \\
&\quad (A, B, C, D) = (D, A, B, C) \\
&\quad u = (D \times (2D+1)) \lll \log_2 w \\
&\quad t = (B \times (2B+1)) \lll \log_2 w \\
&\quad C = ((C - S[2i+1]) \ggg t) \oplus u \\
&\quad A = ((A - S[2i]) \ggg u) \oplus t \\
&\} \\
&\\
&D = D - S[1] \\
&B = B - S[0]
\end{aligned}
$$

Sources of power dissipation:

A. Static Dissipation:

- When all the transistors are off, both nMOS and pMOS have a gate –source threshold voltage below which the current drops exponentially, but at some cases the device works at low threshold voltage, this is sub threshold conduction [16].
- Leakage currents are formed due to of formation of reverse bias between diffusion regions and wells, this forms revers biased diodes and hence there is leakage current [16].

B. Dynamic Dissipation:
CMOS circuits dissipate power through charging and discharging the different load capacitances. Power is dissipated when switching from Vdd to load capacitance to charge it and then again discharge from load to ground.

C. Short Circuit Power Dissipation:
All transistors pMOS and nMOS will have finite rise time and fall time, so during transition both transistors will be on for a short duration for which there will be a current flowing from Vdd to ground [16].

Low Power Synthesis:
Different options available for low power synthesis are
A. Multi Vt
Multiple threshold voltage techniques are hvt, nvt and lvt. These libraries have different power dissipation for each component gates, so selection of these libraries reduces the dynamic power. So selecting a particular library is necessary [18].

B. Clock Gating
Clock gating is a popular technique used in many synchronous techniques for reducing dynamic power dissipation. Clock gating saves power by adding more logic to a circuit to prune the clock tree. Pruning the clock disables the portion of the circuitry so that the flip-flops in them do not have to switch states. Switching states consumes power. When not being switched, the switching power consumption goes to zero and only leakage power are incurred [18].

C. Avoid Cells consuming more power
For a given technology specific libraries a single design operation can be mapped by using various gates of the same library. A single library possess different gates which perform same operation but their characteristics (power, timing, area, capacitance, resistance) of those gates are different, so avoiding cells of more power are important [18].

This is the low power synthesis design flowchart:

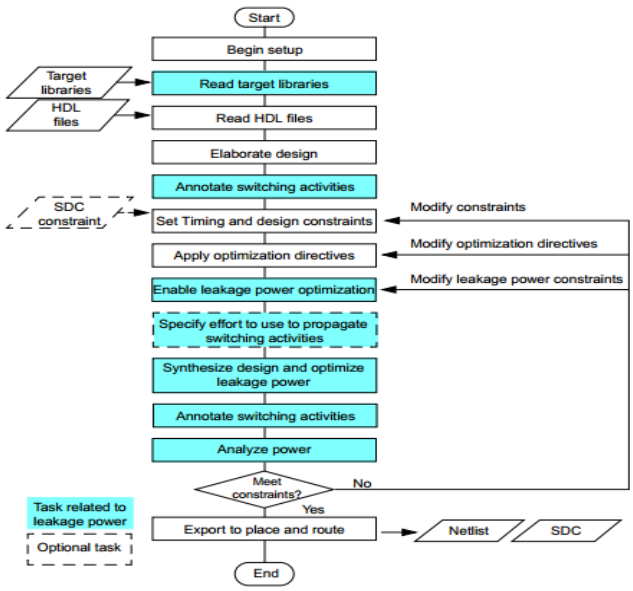

Figure 4. Low power synthesis flow chart

VI. CONCLUSION

In this paper five algorithms were analyzed based on different parameters such as architecture, security, encryption speed, hardware and software suitability and limitations, and based on these parameters each algorithm has its own set of usage advantages and its limitations on targeted platforms. It can be said that no specific algorithm is most suited for all applications with best possible outcome but rather it is more dependent on intention of its application on specific fields. RC6 is nevertheless deficient in any of its boundaries of its sweep over other profound algorithms which had their upvotes on the major areas of correspondence.

REFERENCES

[1] Murat Çakırolu, Cüneyt Bayilmi, Ahmet Turan Özcerit and Özdemir Çetin, "Performance evaluation of scalable encryption algorithm for wireless sensor networks in Scientific Research and Essays Vol. 5(9), pp. 856-861.

[2] Mansoor Ebrahim, Shujaat Khan, Umer Bin Khalid, "Symmetric Algorithm Survey: A Comparative Analysis", in International Journal of Computer Applications (0975 – 8887) Volume 61– No.20, January 2013.

[3] Yee wei law, Jeroen Doumen, and Pieter Hartel, "Survey and Benchmark of Block Ciphers for Wireless Sensor Networks", in ACM Transactions on Sensor Networks, Vol. 2, No. 1, February 2006.

[4] KazumaroAoki and Helger Lipmaa, "Fast Implementations of AES Candidates" K¨ uberneetika AS Akadeemia tee 21, 12618 Tallinn, Estonia.

[5] Helger Lipmaa, "AES Candidates: A Survey of Implementations" K¨uberneetika AS; Akadeemia 21, 12617 Tallinn, Estonia.

[6] Ross Anderson, Eli Biham, Lars Knudsen, "Serpent: A Proposal for the Advanced Encryption Standard", Cambridge University, England.

[7] Dr. Dobbs web page "http://www.drdobbs.com", http://www.drdobbs.com/security/the-twofish-encryption-algorithm/184410744.

[8] Mohan H. S.and A Raji Reddy, "Performance Analysis of AES and MARS Encryption Algorithms", IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 4, No 1, July 2011.

[9] Jens Vesti "The RC6 as a Candidate to the AES" School of Information Technology Deakin University June 8, 2003.

[10] Rajendra H. Rathod, Dr.C.A.Dhote, "A Literature Survey on performance evaluation of query processing on encrypted database", in IJECS ISSN:2319-7242 Volume 3 Issue 12 December 2014.

[11] Mansoor Ebrahim, Shujaat Khan, Umer Bin Khalid, "Symmetric Algorithm Survey: A Comparative Analysis", in International Journal of Computer Applications (0975 – 8887) Volume 61– No.20, January 2013.

[12] Arshiya I, Sekar R, "Securing M2m Post-Quantum Secret Key Cryptography" IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 9, Issue 1, Ver. III

[13] Johann Großschadl, Stefan Tillich, Christian Rechberger, Michael Hofmann, Marcel Medwed "Energy Evaluation of Software Implementations of Block Ciphers under Memory Constraints", Inffeldgasse 16a, A–8010 Graz, Austria.

[14] HossamEl-din H, Ahmed, HamdyM, Kalash, and Osama S.Farag Allah, "Encryption Efficiency Analysis and Security Evaluation of RC6 Block Cipher for Digital Images" World Academy of Science, Engineering and Technology International Journal of Computer, Information, Systems and Control Engineering Vol:1 No:2, 2007.

[15] Bruce Schnier and Doug Whiting, "Performance Comparision of five AES finalists", 7 April 2000.

[16] Kanika Kaur and Arti Noor, "Strategies & methodologies for low power vlsi designs: a review", in International Journal of Advances in Engineering & Technology, May 2011.

[17] Chetan Sharma, "Low power at different levels of vlsi design and clock disribution schemes" in Int. J. Comp. Tech. Appl., Vol 2 (1), 88-93.

[18] Raghava and sujatha, "Synthesis using 65 nanometer library of SPI to wishbone interface" in ICICE 18 June 2013.

[19] Crypto stock exchange.com /cryptography Beta http://crypto.stackexchange.com/questions/11104/how-exactly-was-the-finalist-chosen-in-the-nist-aes-competition.

