ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Flowchart Approach to Scalable Encryption Algorithm using Verilog HDL

Jadala Ravi Teja
M.Tech (VLSI & Embedded Systems),
Siddhartha Institute of Engineering and Technology.

Dr.Dasari Subba Rao, Ph.D
Associate Professor & HOD,
Siddhartha Institute of Engineering and Technology.

Abstract:

The Implementation of encryption/decryption algorithm is the most essential p art of the secure communication. In currently existing encryption algorithms there is a trade-off between implementation cost and resulting performances. SEA is a scalable encryption algorithm targeted for small embedded applications. It was initially designed for software implementations in controllers, smart cards or processors. In this letter, we investigate its performances in recent FPGA devices. For this purpose, a loop architecture of the block cipher is presented. Beyond its low cost performances, a significant disadvantage of the proposed architecture is its full flexibility for any parameter of the scalable encryption algorithm, taking advantage of generic Verilog HDL coding. The letter also carefully describes the implementation details allowing us to keep small area requirements. Finally, a comparative performance discussion of SEA with the Advanced Encryption Standard Randal and ICEBERG(a cipher purposed for efficient FPGA implementations) is proposed. It illustrates the interest of platform/context-oriented block cipher design and, as far as SEA is concerned, its low are requirements and reasonable efficiency.

I.INTRODUCTION:

Scalable encryption algorithm is targeted for small-embedded application with limited resources.SEA is a parametric block cipher for resource constrained systems (e.g. sensor networks, RFIDs) that has been introduced in [1]. It was initially designed as a low-cost encryption/authentication routine (i.e. with small code size and memory) targeted for processors with a limited instruction set(i.e. AND, OR, XOR gates, word rotation and modular addition).

Additionally and contrary to most recent block ciphers (e.g.the DES [2] and AES Randal [3], [4]), the algorithm takes the plaintext, key and the bus sizes as parameters and therefore can be straightforwardly adapted to various implementation contexts and/or security requirements. Compared to older solutions for low cost encryption like TEA (Tiny Encryption Algorithm) [5] or Yuval's proposal [6], SEA also benefits from a stronger security analysis, derived from recent advances in block cipher design/cryptanalysis. In practice, SEA has been proven to be an efficient solution for embedded software applications using micro controllers, but its hardware performances have not yet been investigated. Consequently, and as a first step towards hardware perform and analysis, this letter explores the features of a low cost FPGA encryption/decryption core for SEA.

In addition to the performance evaluation, we show that the algorithm's scalability can be turned into a fully generic Verilog HDL design, so that any text, key and bus size can be straightforwardly re-implemented without any modification of the hardware description language, with standard synthesis and implementation tools. In the rest of the letter, we first provide a brief description of the algorithm specifications. Then we describe the details of our generic loop architecture and its implementation results. Finally, we discuss some illustrative comparisons of the hardware performances of SEA, the AES Rijndael and ICEBERG (a cipher purposed for efficient FPGA implementations)with respect to their design approach (e.g. flexible vs.platform/context-oriented).

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

II. ALGORITHM DESCRIPTION:

A. Parameters and definitions:

SEA n,b operates on various text, key and word sizes. It is based on a Feistel structure with a variable number of rounds, and is defined with respect to the following parameters:

- n: plaintext size, key size.
- b: processor (or word) size.
- n_b = n/2b : number of words per Feistel branch.
- n_r: number of block cipher rounds.

As only constraint, it is required that n is a multiple of 6b (see[1] for the details). For example, using an 8-bit processor, we can derive a 96-bit block ciphers, denoted as SEA96,8.

Let x be a n/2 -bit vector. We consider two representations:

- Bit representation: $x_b = x(n/2-1) \ldots x(2)\, x(1)\, x(0)$.
- Word representation: $x_w = x_{nb-1}\, x_{nb-2} \ldots x2\, x1\, x0$.

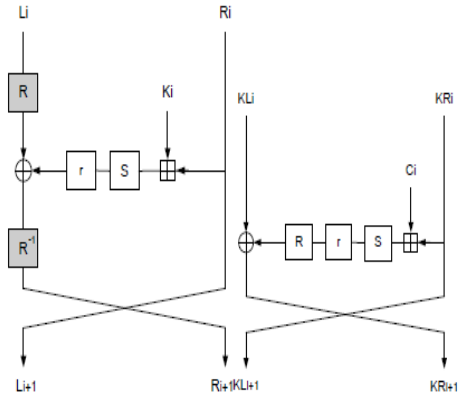

Fig. 1. Encrypt/decrypt round and key round.

B. Basic operations:

Due to its simplicity constraints, SEA n,b is based on a limited number of elementary operations (selected for their availability in any processing device) denoted as follows:

(1) bitwise XOR $\oplus$, (2) addition mod 2b $\boxplus$, (3) a 3-bitsubstitution box S := {0, 5, 6, 7, 4, 3, 1, 2} that can be applied bitwise to any set of 3-bit words for efficiency purposes. In addition, we use the following rotation operations:(4) Word rotation R, defined on nb-word vectors:

$$R : \mathbb{Z}_{2^b}^{n_b} \to \mathbb{Z}_{2^b}^{n_b} : x \to y = R(x) \Leftrightarrow \begin{array}{l} y_{i+1} = x_i, 0 \le i \le n_b - 2, \\ y_0 = x_{n_b - 1} \end{array}$$

(5) Bit rotation r, defined on n_b-word vectors:

$$r : \mathbb{Z}_{2^b}^{n_b} \to \mathbb{Z}_{2^b}^{n_b} : x \to y = r(x) \Leftrightarrow \begin{array}{l} y_{3i} = x_{3i} \ggg 1, \\ y_{3i+1} = x_{3i+1}, \\ y_{3i+2} = x_{3i+2} \lll 1, \end{array}$$

where $0 \le i \le nb/3 - 1$and >>>and <<<respectively represent the cyclic right and left shifts inside a word.

C. The round and key round:

Based on the previous definitions, the encrypt round FE, decrypt round FD and key round FK are pictured in Figure 1and defined as:

$$\begin{array}{ll} [L_{i+1}, R_{i+1}] = F_E(L_i, R_i, K_i) & \Leftrightarrow R_{i+1} = R(L_i) \oplus r(S(R_i \boxplus K_i)) \\ & L_{i+1} = R_i \\ [L_{i+1}, R_{i+1}] = F_D(L_i, R_i, K_i) & \Leftrightarrow R_{i+1} = R^{-1}\left(L_i \oplus r(S(R_i \boxplus K_i))\right) \\ & L_{i+1} = R_i \\ [KL_{i+1}, KR_{i+1}] = F_K(KL_i, KR_i, C_i) & \Leftrightarrow KR_{i+1} = KL_i \oplus R\left(r(S(KR_i \boxplus C_i))\right) \\ & KL_{i+1} = KR_i \end{array}$$

D. The complete cipher:

The cipher iterates an odd number nr of rounds. The following pseudo-C code encrypts a plaintext P under a key K and produces a cipher text C. P,C and K have a parametric bit size n. The operations within the cipher are performed considering parametric b-bit words.

C=**SEA** n,b(P,K)
{
% initialization:
L0&R0 = P;
KL0&KR0 = K;
% key scheduling:

for i **in** 1 **to** $\lfloor \frac{n_r}{2} \rfloor$
$[KL_i, KR_i] = F_K(KL_{i-1}, KR_{i-1}, C(i));$
switch $KL_{\lfloor \frac{n_r}{2} \rfloor}, KR_{\lfloor \frac{n_r}{2} \rfloor}$;
for i **in** $\lceil \frac{n_r}{2} \rceil$ **to** $n_r - 1$
$[KL_i, KR_i] = F_K(KL_{i-1}, KR_{i-1}, C(r-i));$

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

$$\begin{aligned}
&\% \text{ encryption:}\\
&\quad \textbf{for } i \textbf{ in } 1 \textbf{ to } \lceil \tfrac{n_r}{2} \rceil\\
&\qquad [L_i, R_i] = F_E(L_{i-1}, R_{i-1}, KR_{i-1});\\
&\quad \textbf{for } i \textbf{ in } \lceil \tfrac{n_r}{2} \rceil + 1 \textbf{ to } n_r\\
&\qquad [L_i, R_i] = F_E(L_{i-1}, R_{i-1}, KL_{i-1});\\
&\% \text{ final:}\\
&\quad C = R_{n_r} \& L_{n_r};\\
&\quad \text{switch } KL_{n_r-1}, KR_{n_r-1};\\
&\},
\end{aligned}$$

where & is the concatenation operator, KR⌊nr/2 ⌋is taken before the switch and C(i) is a nb-word vector of which all the words have value 0 excepted the LSW that equals i. Decryption is exactly the same, using the decrypt round FD.

III.IMPLEMENTATION OF A LOOP ARCHITECTURE:

A. Description

The structure of our loop architecture for SEA is depicted in figure 2, with the round function on the left part and the key schedule on the right part. Resource-consuming blocks are the S boxes and the mod2b adder; the Word Rotate and Bit Rotate blocks are implemented by swapping wires. According to the specifications, the key schedule contains two multiplexors allowing to switch the right and left part of the round key at half the execution of the algorithm using the appropriate command signal Switch. The multiple xor controlled by Half Exec provides the round function with the right part of the round key for the first half of the execution and transmits its left part instead after the switch. To support both encryption and decryption, we finally added two multiplexors controlled by the Encrypt signal. Supplementary area consumption will be caused by the two routing patches.

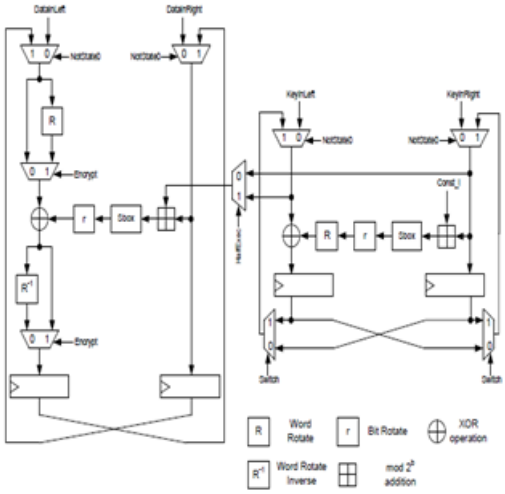

Fig. 2. Loop implementation of SEA.

The algorithm can easily beneficiate of a modular implementation, taking as only mandatory parameters the size of the plaintexts and keys n and the word length b. The number of rounds nr is an optional input that can be automatically derived from n and b according to the guidelines given in [1].From the data path description of Figure 2, a scalable design can then be straightforwardly obtained by using generic Verilog HDL coding. A particular care only has to be devoted to an efficient use of the mod 2^b adders in the key scheduling part.

In the round function, the mod 2^b adders are realized by using n_b b-bits adders working in parallel without carry propagation between them. However, in the key schedule, the signal Const_i (provided by the control part) can only take a value between 0 and $n_r/2$. Therefore, it may not be necessary to use nb adders. If $\log2(n_r/2) \leq b$, then a single adder is sufficient. If $\log2(n_r/2) > b$, then $\lceil \log_2(n_r/2)/2 \rceil$ adders will be required. In the next section, we detail the implementation results of this architecture for different parameters.

B. Implementation Results:

Implementation results were extracted after place and route with the ISE 9.2i tool from Xilinx on a xc4vlx25 VIRTEX-4 platform with speed grade -12.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

In order to illustrate the modularity of our architecture, we ran the design tool for different sets of parameters, with plaintext/key sizes n ranging from 48 to 144 bits and word lengths of 4, 6, 7, 8, and 12bits. For the control part, we used the recommended number of rounds.

$$n_r = [3\frac{n}{4} + 2(\frac{n}{2b} + \frac{b}{2})]^1$$

The computed implementation costs stand for both the operative and control parts. A summary of these results is presented in table I, where the area requirements (in slices), the work frequency and the throughput are provided. We observe that the obtained values for the work frequency are very close for all the implementations. Indeed, the critical path (passing through the key scheduling multiplexors, a mod 2b adder, the Round Function Box, a XOR operator and the multiplexor selecting between encryption or decryption patches) is very similar foray of our selected values for n and b.

TABLE I: IMPLEMENTATION RESULTS FOR SEA WITH DIFFERENT n AND b PARAMETERS

n	b	n_r	♯ of slices	♯ of slice FFs	Output every cycle	Freq (MHz)	Throughput (Mbits/sec)	Thr./Area Mbits/sec /slice
48	4	55	197	127	1/55	237	207	1.049
48	8	51	176	131	1/51	234	220	1.250
72	4	77	296	194	1/77	243	228	0.769
72	6	73	258	194	1/73	242	238	0.924
72	12	73	263	198	1/73	242	239	0.908
96	4	95	368	241	1/95	242	244	0.663
96	8	93	333	246	1/93	238	245	0.737
108	6	111	376	280	1/111	241	235	0.625
126	7	117	438	328	1/117	241	260	0.593
132	11	121	448	330	1/121	227	248	0.554
144	4	149	604	376	1/149	241	233	0.385
144	6	139	488	359	1/139	241	250	0.512
144	8	135	496	371	1/135	241	257	0.518
144	12	133	478	352	1/133	223	236	0.495

For a given n value, it is noticeable that increasing decreases the number of rounds n_r and therefore improves the throughput (since work frequencies are close in all our examples). Similarly, for our set of parameters, increasing b fore given n generally decreases the area requirements in slices. These observations lead to the empirical conclusion that, along as the b parameter is not a limiting factor for the work frequency, increasing the word size leads to the most efficient implementations for both area and throughput reasons.

C. Comparisons with other block ciphers:

For our comparative discussions, we reported a few implementation results of the AES Rijndael in Table II. We selected the implementations in [7], [8] and [9] because their design choices fit relatively well with those of the presented SEA architectures. Mainly, these cores do not take advantage of RAM blocks nor loop unrolling. The four first cores all correspond to loop architectures with a 128-bit data path. They respectively have no pipeline (Pipe0) or a 3-stage pipeline(Pipe3) and use LUT-based or distributed RAM-based Sboxes. The fifth referenced implementation [7] uses a 32-bit data path and consequently reduces the area requirements at the cost of a smaller throughput. Finally, [8] uses a 128-bit data path with a pipelined composite field description of the Sbox. As a matter of fact, a lot of other FPGA implementations of theses can be found in the open literature, e.g. taking advantage of different data path sizes, FPGA RAM blocks, pipelining, unrolling techniques, ..., e.g. [10], [11], [12] and [13].

Additionally, we compared these results with those obtained for ICEBERG, a block cipher optimized for reconfigurable hardware devices. Details on the ICEBERG architecture and different possible implementation tradeoffs are discussed in[14]. The reported result corresponds to a single-round loop architecture without pipeline. Compared to the AES Randal, ICEBERG is built upon a combination of 4-bit operations that perfectly fit into the FPGAs LUTs which intently results in Avery good ratio between throughput and area. The implementation results in Table II lead to the following observations. First, in terms of area requirements (for a data path size equal to the block size), SEA generally exhibits the smallest cost. Measuring the area efficiency with the bit per slice metric leads to a similar conclusion. Of course, the area requirements of, e.g. the AES Rijndael could still be decreased by using smaller data paths [15] and such a comparative table only serves as an indicator

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

rather than a strict comparison. However, in the present case, these results clearly suggest the low-cost purpose of our presented implementations. By contrast, looking at the throughput per area metric indicates that these low area requirements come with weak throughputs. This is of course mainly due to the high number of rounds in SEA. With this respect, it is interesting to compare SEA and ICEBERG since their implementation results clearly illustrate their respective context/platform-oriented design approach. Namely SEA is purposed for low cost applications while ICEBERG optimizes the throughput per slice. These numbers also confirm the differences between specialized algorithms and standard solutions. It must be underlined with this respect that the AES Randal still ranges relatively well in terms of hardware cost and throughput efficiency, compared to the investigated specialized solutions. Note also that SEA was initially purposed for low cost software implementations. While these design criteria turned out to allow low cost hardware implementations as well, it is likely that targeting a cipher specifically for low cost hardware would lead to even better solutions, e.g. [16]. Finally, it is also important to emphasize a number of advantages in SEA that cannot be found in other recent block ciphers, namely its simplicity, scalability (re-implementing SEA for a new block size does not require to re-write code),good combination of encryption and decryption and ability to derive keys "on the fly" both in encryption and decryption.

TABLE II: IMPLEMENTATION RESULTS OF OTHER BLOCK CIPHERS.

Algorithm	Device	n_r	E/D	# of slices	Freq (MHz)	Throughput (Mbits/sec)	Thr./Area Mbits/sec /slice	bit/slice
AES (Pipe0-LUT) [9]	xc2v400	10	no	2744	59	760	0.277	0.047
AES (Pipe0-Dist) [9]	xc2v400	10	no	1780	78	1000	0.562	0.072
AES (Pipe3-LUT) [9]	xc2v400	10	no	2909	148	1890	0.650	0.044
AES (Pipe3-Dist) [9]	xc2v400	10	no	1940	178	2280	1.175	0.066
AES [7]	xcv100e	10	yes	1125	161	215	0.191	0.114
AES [8]	xcv3200e	10	no	1769	167	2085	1.179	0.072
ICEBERG	xc4vlx25	16	yes	575	247	988	1.718	0.111
$SEA_{128,7}$	xcv3200e	117	yes	434	92	99	0.228	0.290
$SEA_{128,7}$	xc2v4000	117	yes	424	145	156	0.368	0.302
$SEA_{128,7}$	xc4vlx25	117	yes	438	241	260	0.594	0.288

IV SYNTHESIS AND SIMULATION RESULTS:

To investigate the advantages of using our technique in terms of area overhead against "Fully ECC"and against the partially protection, we implemented and synthesized for a Xilinx XC3S500E different versions of a32-bit, 32-entry, dual read ports, single write port register file. Once the functional verification is done, the RTL model is taken to the synthesis process using the Xilinx ISE tool. In synthesis process, the RTL model will be converted to the gate level net list mapped to a specific technology library. Here in this Spartan 3E family, many different devices were available in the Xilinx ISE tool. In order to synthesis this design the device named as "XC3S500E" has been chosen and the package as "FG320" with the device speed such as "-4". The corresponding schematics of the adders after synthesis is shown below.

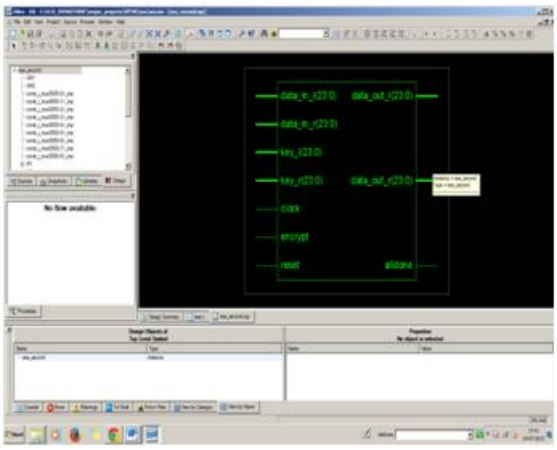

Fig.3. RTL schematic of SEA

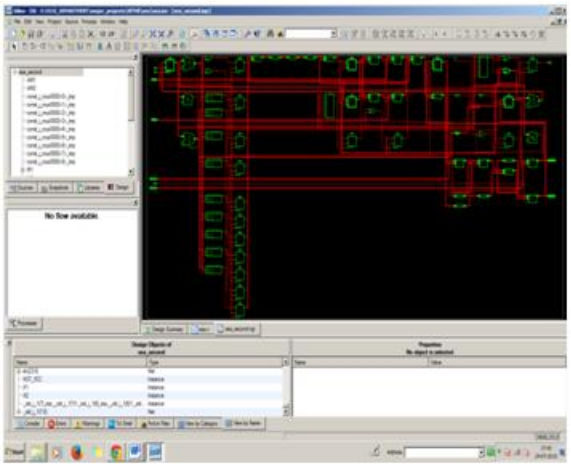

Fig.4. RTL schematic of Internal blocks of SEA

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

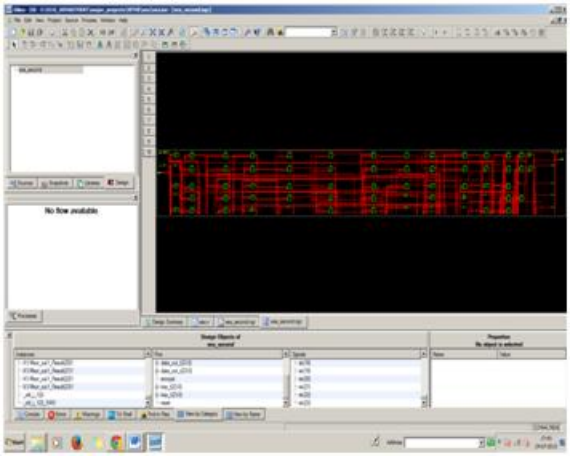

Fig.5. Technology schematic of SEA

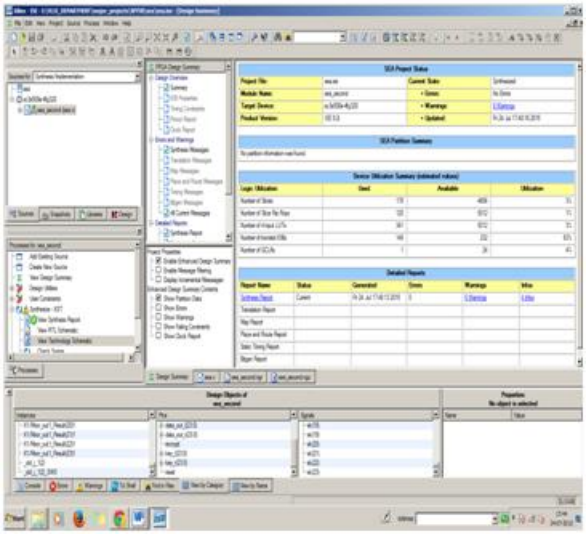

Fig.6.Synthesis report of SEA

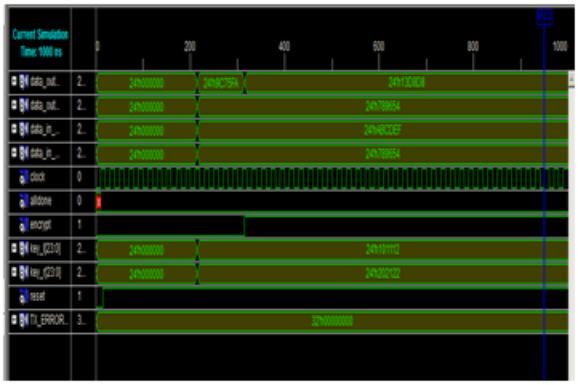

Fig.7.Simulation of SEA

V. CONCLUSION:

This letter presented FPGA implementations of a scalable encryption algorithm for various sets of parameters. The presented parametric architecture allows keeping the flexibility of the algorithm by taking advantage of generic Virology HDL coding. It executes one round per clock cycle, computes the round and the key round in parallel and supports both encryption and decryption at a minimal cost. Compared to other recent block ciphers, SEA exhibits a very small area utilization that comes at the cost of a reduced throughput. Consequently, it can be considered as an interesting alternative for constrained environments. Scopes for further research include low power SIC implementations purposed for RFIDs as well as further cryptanalysis efforts and security evaluations.

REFERENCES:

[1]F.-X. Standard, G. Piret, N. Gershen feld, and J.-J.Quisquater, "SEA:A Scalable Encryption Algorithm for Small Embedded Applications," in the Proceedings of CARDIS 2006, ser. LNCS, vol. 3928, Taragona, Spain, 2006, pp. 222–236.

[2]Data Encryption Standard, NIST Federal Information Processing StandardFIPS 46-1, Jan. 1998.

[3]J. Daemen, V. Rijmen, The Design of Rijndael. Springer-Verlag, 2001.

[4]Advanced Encryption Standard, NIST Federal Information Processing Standard FIPS 197, Nov. 2001.

[5]D. Wheeler and R. Needham, "TEA, a Tiny Encryption Algorithm," in the Proceedings of Fast Software Encryption - FSE 1994, ser. LNCS, vol. 1008, Leuven, Belgium, Dec. 1994, pp. 363–366.

[6]G. Yuval, "Reinventing the Travois: Encryption/MAC in 30 ROM Bytes," in the Proceedings of Fast Software Encryption - FSE 1997, ser. LNCS, vol. 1267, Haifa, Israel, Jan. 1997, pp. 205–209.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

[7]N. Pramstaller and J. Wolker storfer, "A Universal and Efficient AES Coprocessor for Field Programmable Logic Arrays," in the Proceedings of FPL 2004, LNCS, vol. 3203, Leuven, Belgium, Aug. 2004, pp. 565–574.

[8]F.-X. Standaert, G. Rouvroy, J.-J.Quisquater, and J.-D. Legat, "Efficient Implementation of Rijndael Encryption in Reconfigurable Hardware: Improvement sand Design Tradeoffs," in the Proceedings of Cryptographic Hardware and Embedded Devices - CHES 2003, ser. LNCS, vol. 2779,Cologne, Germany, Sep. 2003, pp. 334–350.

[9]J. Zambreno, D. Nguyen, and A. Choudhary, "Exploring Area/Delay Tradeoffs in an AES FPGA implementation," in the Proceedings of FPL2004, ser. LNCS, vol. 3203, Leuven, Belgium, Aug. 2004, pp. 575–585.

[10]K. Gaj and P. Chodowiec, "Fast Implementation and Fair Comparisonof the Final Candidates for Advanced Encryption Standard Using Field Programmable Gate Arrays," in Topics in Cryptology - CT-RSA 2001,LNCS., vol. 2020, San Fransisco, USA, pp. 84-99.

[11]G. P. Saggese, A. Mazzeo, N. Mazzocca, and A. G. M. Strollo,"An FPGA-Based Performance Analysis of the Unrolling, Tiling, and Pipelining of the AES Algorithm," in the Proceedings of FPL 2003, ser.LNCS, vol. 2778, Lisbon, Portugal, Sep. 2003, pp. 292–302.

[12]A. J. Elbirt, W. Yip, B. Chetwynd, and C. Paar, "An FPGA Implementation and Performance Evaluation of the AES Block Cipher CandidateAlgorithm Finalists," in AES Candidate Conference, 2000, pp. 13–27.

[13]K. Jarvinen, M. Tommiska, J. Skytta, "Comparative Survey of High-Performance Cryptographic Algorithm Implementations on FPGAs," IEE Proceedings on Information Security, vol. 152, Oct. 2005, pp. 3–12.

[14]F.-X. Standaert, G. Piret, G. Rouvroy, and J.-J.Quisquater, "FPGA Implementations of the ICEBERG Block Cipher," in the Proceedings of ITCC 2005, vol. 1, Las Vegas, USA, Apr. 2005, pp. 556–561.

[15]M. Feldhofer, J. Wolkerstorfer, and V. Rijmen, "AES Implementationon a Grain of Sand," in IEE Proceedings on Information Security, vol.152. IEE, Oct. 2005, pp. 13–20.

[16] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B.-S. Koo, C. Lee,D. Chang, J. Lee, K. Jeong, J. Kim, and S. Chee, "HIGHT: a New Block Cipher Suitable for Low-Resource Devices," in The Proceedings of Cryptographic Hardware and Embedded Devices - CHES 2006, ser.LNCS, vol. 4249, Yokohama, Japan, Oct. 2006, pp. 13–20.

Author's Details:

Jadala Ravi Teja

Is an M.Tech (VLSI & EMBEDDED SYSTEMS) student In Department of Electronics & Communication Engineering ,from Siddhartha Institute of Engineering and Technology, Ibrahimpatnam, Hyderabad, Telangana. His interest of fields in VLSI & Embedded systems and Networking.

Dr. D Subba Rao

Is a proficient Ph.D person in the research area of Image Processing from Vel-Tech University, Chennai along with initial degrees of Bachelor of Technology

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

in Electronics and Communication Engineering (ECE) from Dr. S G I E T, Markapur and Master of Technology in Embedded Systems from SRM University, Chennai. He has 13 years of teaching experience and has published 12 Papers in International Journals, 2 Papers in National Journals and has been noted under 4 International Conferences. He has a fellowship of The Institution of Electronics and Telecommunication Engineers (IETE) along with a Life time membership of Indian Society for Technical Education (ISTE). He is currently bounded as an Associate Professor and is being chaired as Head of the Department for Electronics and Communication Engineering discipline at Siddhartha Institute of Engineering and Technology, Ibrahimpatnam, Hyderabad.