Improving the Performance of a Scalable Encryption Algorithm (SEA) using FPGA

***Praveen Kumar. B[1], P. Ezhumalai[2], P. Ramesh[3,] Dr.S. Sankara Gomathi [4], Dr . P. Sakthivel[5]**
[1, 3] PG Scholars., [2]Assistant professor, Dept of Computer Science.
[1 2 3]Sri Venkateswara College of Engineering, Anna University. Sriperumbudur -602 105.

[4] R.M.K Engineering College , Dept. of Information Technology ,Chennai .
[5] College of Engineering. Guindy .Anna University. Chennai .

Abstract
The present symmetric encryption algorithms result from a Tradeoff 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 Paper we proposed a system that investigates its performances in recent field-programmable gate array (FPGA) devices. The proposed system is applicable where there are limited processing resources and throughput requirements. For this purpose, we propose low-cost encryption routines (*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). The proposed design is parametric in the text, key and processor size, provably secure against linear or differential cryptanalysis, allows efficient combination of encryption/decryption and on-the-fly" key derivation. Beyond its low cost performances, a significant advantage of the proposed architecture is its full flexibility for any parameter of the scalable encryption algorithm, taking advantage of generic VHDL coding.
Keywords:
Computer security, DES - Data Encryption Standard, VHDL – Hardware Description Language, FPGA – Field Programmable Gate Array.

1. INTRODUCTION

Scalable encryption algorithm (SEA) is a parametric block cipher for resource constrained systems (example sensor networks RFIDs). 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). This 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.
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.

Manuscript received February 5, 2010
Manuscript revised February 20, 2010

Consequently, and as a first step towards hardware performance analysis, this paper explores the features of a low-cost field-programmable gate array (FPGA) encryption or decryption core for SEA. In addition to the performance evaluation, we show that the algorithm's scalability can be turned into a fully generic VHDL design, so that any text, key, and bus size can be straightforwardly reimplemented without any modification of the hardware description language, with standard synthesis and implementation tools.
Present block ciphers, like the Advanced Encryption Standard and Rijndael rather focus on finding a good tradeoff between cost, security and performances. While this approach is generally the most convenient, there exist contexts where more specialized ciphers are useful. As a motivating Example, ICEBERG is targeted for the hardware Implementations and shows significant Efficiency improvements on these platforms compared to other algorithms. Embedded applications such as building infrastructures present a significant opportunity and challenge for such new cryptosystems.

The work discussed here is a currently undergoing Final year project. The remainder of the paper is organized as follows: Section II discusses the related works. Section III introduces the power consumption and section IV describes the proposed work and Conclusions are presented in Section V.

2. RELATED WORK

A detailed literature survey gave an insight into various related ideas. However, they did not encompass all the parameters which we have proposed for analysis.
Scalable encryption algorithm (SEA) is a parametric block cipher for resource constrained systems (e.g., sensor networks, RFIDs) that has been introduced in [1]. SEA **n** and **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;
• nb: number of words per Feistel branch;
• nr number of block cipher rounds.

Note: As an only constraint, it is required that n is a multiple of **6b.**

Let x be a n=2-bit vector. We consider the following two representations.

• **Bit representation**: xb = x((n=2) 1) _ x(2) x(1) x(0).
- - - - **(1)**

• **Word representation**: xW = xn 1 xn 2 _ x2 x1 x0.
- - - - **(2)**

The number of rounds nr is an optional input that can be automatically derived from n and b according to the guidelines given in [2].
A summary of the paper [1] is presented in the below Figures, where the area requirements (in slices), the throughput in Figure [1] are provided. We observe that the obtained values for the work frequency are very close for all the implementations and also they compared with all other block ciphers in Figure 2.

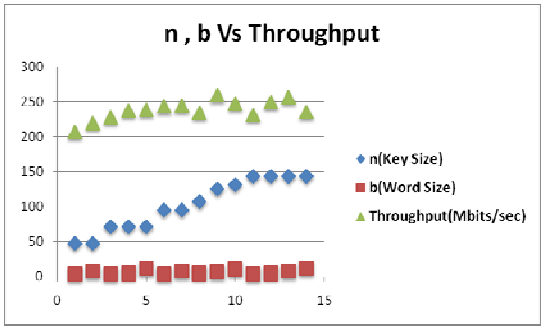

Figure 1: Throughput vs n and b

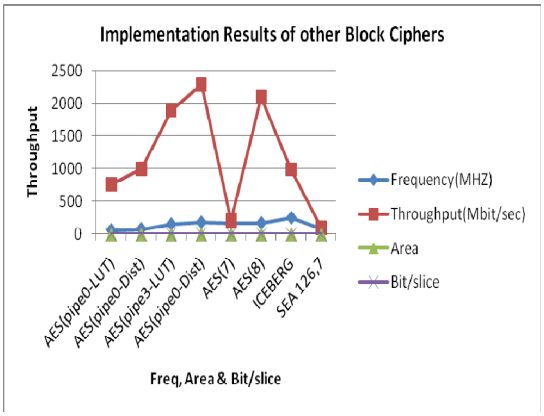

Figure 2: Comparison with other Block Ciphers

In the paper [2], we consequently consider a general context where we have very limited processing resources (*e.g.* a small processor) and throughput requirements. It yields design criteria such as: low memory requirements, small code size, limited instruction set.
In addition, they propose the exibility as another unusual design principle. In opposition, SEA **n and b** allows to obtain a small encryption routine targeted to any given processor, the security of the cipher being adapted in function of its key size. Both of encryption and decryption result in an improved efficiency and are particularly relevant in contexts where the same constrained device has to perform encryption and decryption operations (*e.g.* authentication).
In the paper [3] they discussed the implementation of AES and concluded that AES minimizes mean power consumption .The design of AES hardware implementation used flexible methodology which put forth a lot of possible optimization ideas. All ideas were evaluated regarding their impact on the silicon size and on the power efficiency. The evaluation is based on synthesis results and circuit-level simulations. These in-depth analyses ensure that our circuit achieves the ambitious requirements for passively powered devices

A closely related work is also presented in paper [4], which studies the AES implementation on Xilinx vertex family using FPGA's and also they have shown that FPGAs can be used very efficiently for high-speed implementations of cryptographic algorithms and also it can be efficiently implemented on FPGAs for applications with various requirements. Both very high performance and low area requirements can be efficiently achieved using the methods presented in the paper [4] .

3. POWER CONSUMPTION

The distribution of the current consumption is as follows. The RAM circuit consumes 52% of the power, followed by the controller, which uses 18%. The data path requires the remaining power. The S-Box needs 14%, the MixColumns circuit 8%, and the rest of the circuit 8%. The rest is consumed by the output Multiplexer, the temporary storage register, the XOR gates, and the Rcon module.

4. PROPOSED WORK

In the paper [1] for all input parameters it achieves Throughput almost closer to each other. The Proposed system improves the throughput by varying the key size and plain text size.

A. DESIGN

In the Design the 4 – bit Look Up Table (LUT) from the ICEBERG is integrated with the loop architecture of SEA to increase the throughput and reduce the area as shown in the Figure 3. Compared to the AES Rijndael, ICEBERG is built upon a combination of 4-bit operations that perfectly fit into the **FPGAs LUTs** which intently results in a very good ratio between **throughput and area.**
SEA achieves reduced throughput compared to other block-ciphers. By merging ICEBERG architecture with SEA loop architecture. The system can achieve the improvement in the throughput. Because SEA exhibits low cost and Area size but ICEBERG exhibits improved in Throughput.

B. LOOK UP TABLE (LUT)

A lookup table is a data structure, usually an array or associative array, often used to replace a runtime computation with a simpler array indexing operation. The savings in terms of processing time can be significant since retrieving a value from memory is often faster than undergoing an 'expensive' computation or Input/output operation.

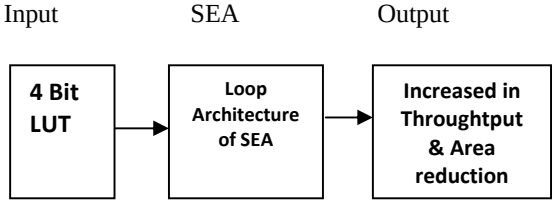

Figure 3 : Flow Diagram

The Figure 3 shows the flow diagram to achieve the increased in the Throughput (Mbits/Sec) and some reduction in Area/slice.

Table 1: Key size vs. Area and Throughput

n(Key Size)	b(Word Size)	Area / Slice /Mbits/Sec	Throughput (Mbits/sec)
58	4	1.039	217
58	8	1.15	230
82	4	0.659	238
82	6	0.814	248
82	12	0.808	249
106	4	0.563	254
106	8	0.637	255
116	6	0.525	245
136	7	0.493	270
142	11	0.454	258
154	4	0.285	242
154	6	0.412	260
154	8	0.418	267
154	12	0.395	246

The value obtained from the experimental is presented in table 1, which shows that increase in the key size, reduces the Area/Slice and also improves the throughput.

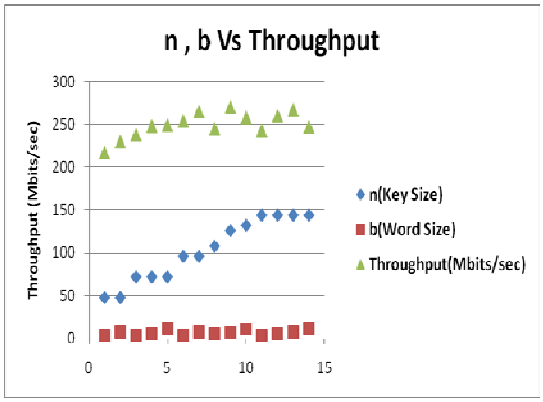

Figure 4 : Increasing Throughput

The existing algorithm deals with the fixed plain text and fixed key size which provides less security. To achieve more security over the traditional algorithm the proposed architecture implements the variable key size and plain text.
By varying the parameter **n (Key size)** the throughput is increased as shown in the Figure 5. Similarly, for our set of parameters, increasing b for a given n generally decreases the area requirements in slices. These observations lead to the empirical conclusion that as long 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.

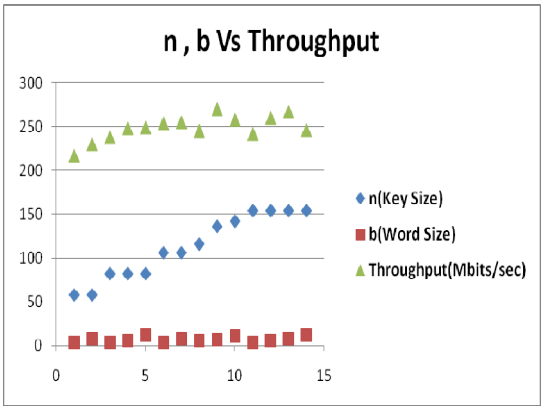

Figure 5 : Change in the Key size

Low power ASIC implementation. (by reducing the no. of gates).(i.e) It should be implemented on as low a level as possible in order to guarantee maximum performance with minimum resources. By implementing the above mentioned factors we can decrease the working space area

and cost for the proposed architecture. Figure 4 and Figure 5 and 6 shows the proposed system result.

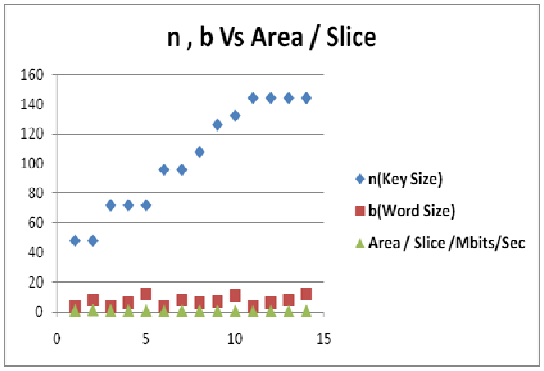

Figure 6: Reduced in Area / Slice

Figure 7 describes that, varying the key size results in the reduction of Area / Slice.

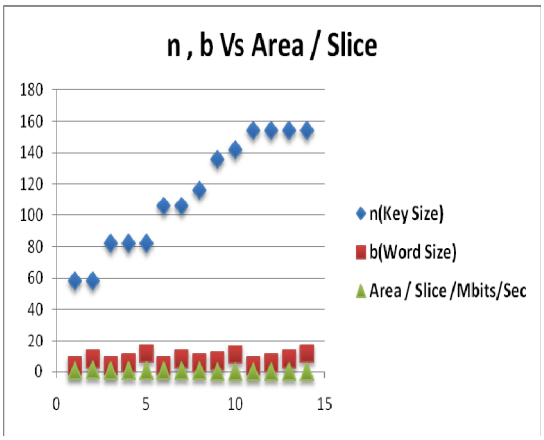

Figure 7 : Change in the Area / slice w.r.to Key size

Hence by using the variable key size and plain text size , the systems obtains the improved performance in throughput and reduction in the Area/slice.

5. CONCLUSION

This paper 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 VHDL 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 ASIC implementations purposed for RFIDs as well as further cryptanalysis efforts and security evaluations.

REFERENCES

[1] F. Mace, F . -X. Standaert, and J.-J Quisquater "FPGA implementation(s) of a Scalable Encryption Algorithm," in IEEE Transaction on very large scale integration (VLSI) systems ,VOL.16, NO. 2, FEBRUARY 2008

[2] F.-X. Standaert, G. Piret, N. Gershenfeld, 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.*

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

[4] 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.*

[5] Data Encryption Standard, FIPS PUB 46-3, Oct. 1999.

[6] K. Gaj and P. Chodowiec, "Fast implementation and fair comparison of the final candidates for advanced encryption standard using field programmable gate arrays," *in Proc. Topics Cryptol. (CT-RSA), 2001, pp. 84–99.*

[7] J. Zambreno, D. Nguyen, and A. Choudhary, "Exploring area / delay tradeoffs in an AES FPGA implementation," in *Proc. FPL*, 2004, pp. 575–585.

[8] K. Gaj and P. Chodowiec, "Fast implementation and fair comparison of the final candidates for advanced encryption standard using field programmable gate arrays," in *Proc. Topics Cryptol. (CT-RSA)*, 2001, pp. 84–99.

[9] 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 *Proc. FPL*, 2003, pp. 292–302.

[10] A. J. Elbirt, W. Yip, B. Chetwynd, and C. Paar, "An FPGA implementation and performance evaluation of the AES block cipher candidate algorithm finalists," in *Proc. AES Candidate Conf.*, 2000, pp. 13–27

[11] F.-X.Standaert, G.Piret G.Rouvroy, and J.-J. Quisquater, "FPGA implementations of the ICEBERG block cipher," *in Proc. ITCC, 2005, pp. 556–561.*

[12] G. Hachez, F. Koeune, J.-J. Quisquater, *cAESar Results: Implementation of Four AES Candidates on Two Smart Cards*, in the proceedings of the Second Advanced Encryption Standard Candidate Conference, pp 95-108, Rome, Italy, March 1999.

Praveen Kumar B. I am currently doing my Final year M.E in Computer Science and Engineering in Sri Venkateswara College of Engineering. I received my B.E Degree in Electronics and Communication engineering in the Year (2008), Anna University, Chennai, India. My Area of interest is VLSI Architecture ,Computer Networks and Network Security.

Ezhumalai Periyathambi received the B.E degree in Computer Science and engineering from Madras University, Chennai, India in 1992 and Master Technology (M.Tech.) in computer science and Engineering from J N T University, Hyderabad, India in 2006. He is currently working towards the Ph.D degree in Department of Information and Communication, Anna University, Chennai, India. He is working as assistant Professor in the Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, sriperumbudur, Chennai, Tamilnadu, India. His research in reconfigurable architecture, Multi-Core Technology CAD – Algorithms for VLSI architecture, Theoretical computer science and mobile computing.

Ramesh. P. I am currently doing my Final year M.E in Computer Science and Engineering in Sri Venkateswara College of Engineering. I received my B . E Degree in Electrical and Electronics engineering in the Year (2006), M.G.R University, Chennai, India. My Area of interest is VLSI Architecture, Embedded System, Computer Networks and Security