Efficient Implementation of a Scalable Encryption Algorithm using FPGA

B. Praveen Kumar
Lecturer – Department of Information Technology
Velammal Institute of Technology
Chennai, India

P. Ezhumalai
Professor – Department of Computer Science
Rajalakshmi Engineering College
Chennai, India

Dr. S. Sankara Gomathi
Professor – Department of Information Technology
Rajalakshmi Engineering College
Chennai, India

ABSTRACT
Initially SEA is 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 present symmetric encryption algorithms result from a tradeoff between implementation cost and resulting performances. The proposed system is applicable where there are limited processing resources with high throughput requirements.

For this purpose, we propose a SEA loop Architecture with Behavior model (VHDL) coding. So the number of logic gates required is very less when compared with Gate level model. Because of less number of logic gates, time taken to execute the loop architecture is less. So we are achieving a faster execution time (Frequency in MHZ). The proposed design is parametric in the key and word size, provably secure against linear or differential cryptanalysis. 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.

General Terms
FPGA, Scalable Encryption algorithm

Keywords
FPGA – Field Programmable Gate Array, Computer security, DES - Data Encryption Standard, VHDL – Hardware Description Language**.**

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 straight forwardly 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.

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 re-implemented 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 remainder of the paper is organized as follows: Section II discusses the related works. Section III introduces the proposed work and section IV describes the performance analysis and Conclusions are presented in Section V.

2. RELATED WORK
All 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.

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 **Complete cipher** of the paper [1] is presented in the below algorithm. The cipher iterates an odd number nr of rounds. The following pseudo-C code encrypts a plain text 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=SEAn;b(P;K)
{
        %initialization:
    L0&R0 = P;
   KL0&KR0 = K;
        %keyscheduling:
   for i in 1 to nr / 2
   [KLi;KRi]= FK(KLi¡1;KRi¡1;C(i));
   switch KL nr/2, KR nr/2
   for i in nr/2 to nr - 1
   [KLi;KRi]= FK(KLi¡1;KRi¡1;C(r ¡ i));
        %encryption:
   for i in 1 to nr/2
   [Li;Ri]= FE(Li¡1;Ri¡1;KRi¡1);
   for i in nr/2+1 to nr
   [Li;Ri]= FE(Li¡1;Ri¡1;KLi¡1);
        %Final:
   C = Rnr&Lnr ;
   switch KLnr¡1, KRnr¡1;
}
```

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 proposed 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. PROPOSED WORK

In the paper [1] for all input parameters it achieves less Execution time (i.e.) Frequency in (MHZ) and also it occupied more working area or space. The Proposed system improves the Execution time by varying the key size and Word length parameter as input.

3.1 Behavior model or Algorithmic level model

This is the highest level of abstraction provided by HDL. A module can be implemented in terms of the desired design algorithm without concern of the hardware implementation details. Thus architectural evaluation takes place at an algorithmic level where the designers do not necessarily think in terms of logic gates flow but in terms of the algorithm they wish to implement in hardware.

3.2 Design

In the Proposed system we used Behavior Model – Hardware Description Language (VHDL). So the number of Slices (number of Flip Flops, Gates, Registers used) is less when compared with the Existing System, because in the Existing

System they used Gate level Model. So each and every blocks presented in the loop architecture was designed with the help of many logic Gates. So Area occupied was high and also used more internal register.

SEA achieves reduced throughput compared to other block-ciphers. So by designing the loop architecture as mentioned above we can achieve the quicker **Execution time**, number of **Slices** used is less, Working **area** is less and finally **Throughput** can be increased.

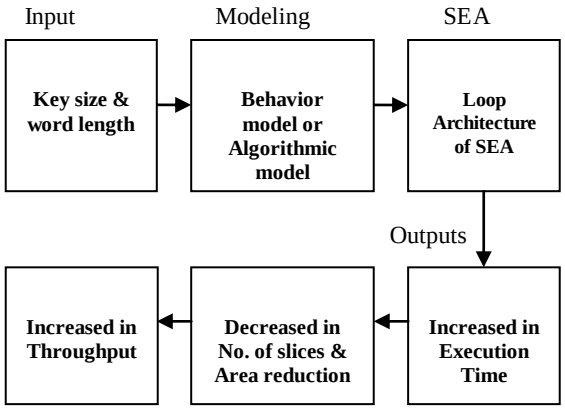

Figure 1: Flow Diagram

The Figure 1 shows the flow diagram to achieve the increase in the Throughput (Mbits/Sec), reduction in Area/slice and faster Execution Time.

3.3 Implementation Results

Implementation results were extracted after place and route with the ISE 9.1 i tool from Xilinx on XC4VSX25 VIRTEX 4 platform with speed grade – 12 and package FF668. 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 bits and word lengths of > 8.

Table 1: Implementation Result

n	b	# of slices	#of slices FFs	Freq (MHZ)	Thro/Area (Mbits/Sec)
48	8	171	162	310	1.050
72	12	236	214	296	0.708
96	16	282	256	246	0.537
108	18	376	280	241	0.525
126	8	369	318	251	0.493
132	12	399	336	274	0.454
144	8	407	366	276	0.418
152	11	429	388	277	0.398
160	8	443	398	265	0.356
164	11	462	413	275	0.311

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

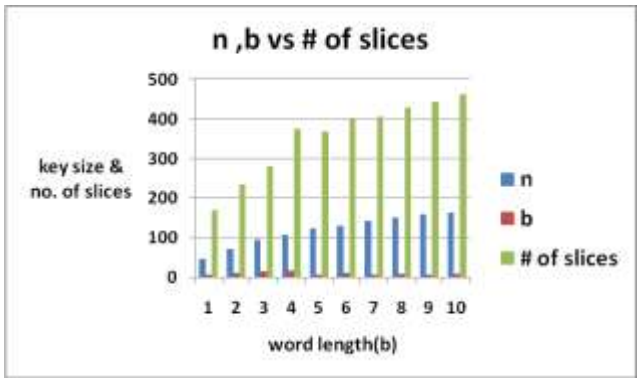

Figure 2: Reduction in slices w.r.to n & b

Figure 2, shows the reduction of slices with respect to the plain text key (n) and Word length (b). In the case of existing system the number of slices was more. To achieve less number of slices requirement we increased the key size and word length.

By varying the parameter **n (Key size)** there is an increase in throughput which is shown in the **Figure 3.** 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.

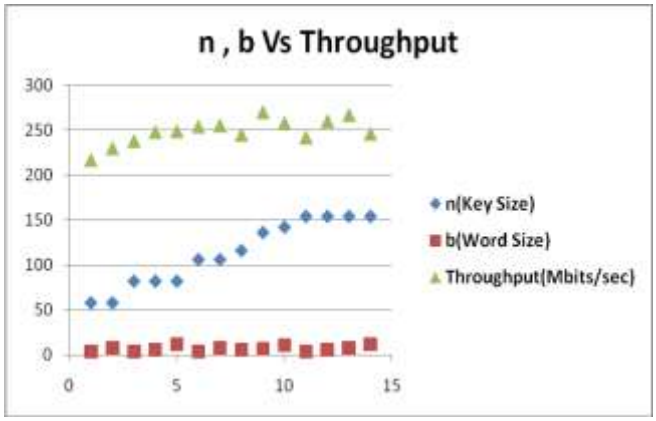

Figure 3 : Increase in Throughput

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 using the above mentioned factors in implementation we achieved the decrease in working space area and cost for the proposed architecture. The results achieved by the experimentation had been depicted in Figure 2, Figure 3 and Figure 4.

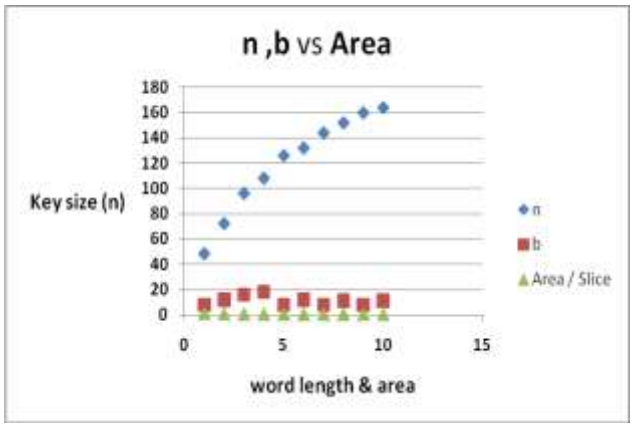

Figure 4: Reduced in Area / Slice

Figure 5 describes that, varying the key size results in the faster execution time in terms of frequency in MHZ.

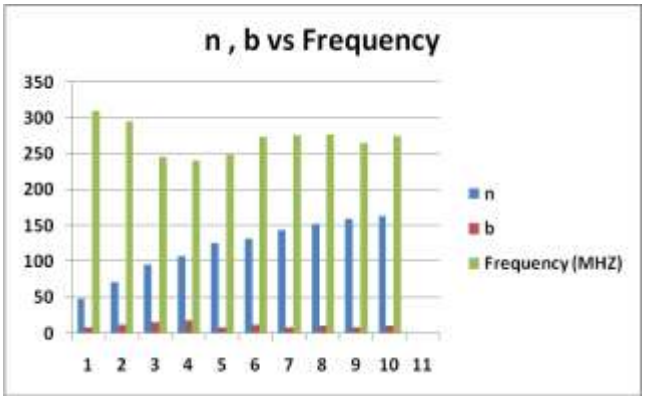

Figure 5 : Change in the Frequency (MHZ) w.r.to key size & word length

Hence by using the variable key size the systems obtains the improved performances in the throughput, frequency and reduction in the Area/slice and number of slices also.

4. PERFORMANCE ANALYSIS

SEA n, b is targeted for being implemented on low - cost processors, with little code size and a small instruction set. However, SEA n, b's simple structure makes it easy to implement on any processor. We propose a pseudo assembly code of an encryption / decryption design with on the fly key scheduling. The implementation objectives were, in decreasing order of importance

(1) Low RAM and registers usage

(2) Low code size and

(3) Speed. It is based on the following (very) reduced instruction set.

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.

5.1 Future Work

Scopes for further research include low power ASIC implementations purposed for RFIDs as well as further cryptanalysis efforts and security evaluations.

6. 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 AdvancedEncryption Standard Candidate Conference, pp 95-108, Rome, Italy, March 1999.

[13] A.Biryukov,D.Wagner,Slideattacks,in the proceedings of FSE1999, Lecture Notes in Computer Sciences, vol1 636, pp 245 - 259, Rome, Italy, March 1999 Springer-Verlag.

[14] E.Biham, A.Shamir, Differential Cryptanalysis of the Data Encryption Standard,1993, Springer Verlag.

[15] A. Lee, NIST Special Publication 800-21, Guideline for Implementing Cryptographyin the Federal Government, National Institute of Standards and Technology.

BIOGRAPHY

Praveen Kumar Bhuvaneswaran completed his M.E in Computer Science and Engineering at sri Venkateswara College of Engineering in the year of 2010. He received his B.E Degree

in Electronics and Communication Engineering in the Year (2008) at Sri Venkateswara College of Engineering, Sriperumbudur, Chennai. He is currently working as Lecturer in Velammal Institute of Technology , Chennai, India. My Area of interest is VLSI Architecture, Computer Networks, Data Mining, Image Processing 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 Faculty in the Department of Computer Science and Engineering, Rajalakshmi Engineering College , Chennai, Tamilnadu, India.

His research in reconfigurable architecture, Multi-Core Technology CAD – Algorithms for VLSI architecture, Theoretical computer science and mobile computing.

Dr. S.Sankara Gomathi received her PhD in Mobile and wireless communication from Anna University. She is currently a Professor of Rajalakshmi Engineering College, Chennai, India. GOMATHI served as a review committee member for various international Journals and conferences. She has organized various short term training program. She is a senior member of IACSIT, IAEngg and life member of ISTE. Her key research areas are Mobile Computing, Mobile security, Communication,Wireless sensor networks, Pervasive computing and networking.Currently she is working on energy conservation issues for mobile adhoc networks.