RESEARCH ARTICLE OPEN ACCESS

Design And Implementation Of Tiny Encryption Algorithm

Kiran Kumar.V.G*, Sudesh Jeevan Mascarenhas**,Sanath Kumar***, Viven Rakesh J Pais ****
*(Associate Professor, Department of E & C Engineering, Sahyadri College of Engineering Adyar, Mangalore)
** (Department of E & C Engineering, Sahyadri College of Engineering Adyar, Mangalore-7)
** *(Department of E & C Engineering, Sahyadri College of Engineering Adyar, Mangalore-7)
**** (Department of E & C Engineering, Sahyadri College of Engineering Adyar, Mangalore-7)

ABSTRACT

Over the recent years, several smart applications like RFID’s, sensor networks, including industrial systems, critical infrastructures, private and public spaces as well as portable and wearable applications in which highly constrained devices are interconnected, typically communicating wirelessly with one another, working in concert to accomplish some task. Advanced safety and security mechanisms can be very important in all of these areas. Light weight cryptography enables secure and efficient communication between networked smart objects. This proposed system focuses on the FPGA implementation of light weight cryptographic algorithm Tiny Encryption Algorithm TEA to adapt with many real time constraints such as memory, data loss and low cost. The proposed scheme uses Linear Feedback Shift Register to generate the random key making it more secure for sensitive information transfer in many real-time applications. In this study,operation of this cryptosystem is analyzed by implementing the cryptographic algorithm TEA with the key generation unit in FPGA Spartan 3E. We have also compared the results with the IDEA.

Keywords - Light weight cryptography, Linear feedback shift register,Tiny Encryption Algothim RFID.

I. INTRODUCTION

Data Security is a primary issue in any wireless cryptographic protocol, a cryptographic algorithm is an essential part in network security.

One of the state-of-the-art techniques is “Lightweight Cryptography (LWC)”. Lightweight cryptography is a cryptographic algorithm or protocol tailored for implementation in constrained environments like RFID’s, sensor networks, healthcare, the Internet of Things, cyber-physical systems, distributed control systems, indicators, measuring devices, custom controllers, smart power system etc.

In today’s era of pervasive computing, FPGA systems are deployed in a broad scope of areas, like RFID’s, sensor networks, healthcare, the Internet of Things, cyber-physical systems, measuring devices, custom controllers, smart power system etc..

An important strategy of the functionality of these schemes is the data storage, data access and transmission of private, raw or even critical information. Consequently, the confidentiality and integrity of the resources and services of said devices constitute a prominent issue that must be debated during their conception. There are a variety of cryptographic mechanisms which are used to safeguard the confidentiality and integrity of stored and transmitted information. In the context of FPGA systems, nonetheless, the problem at hand is exacerbated by the resource-constrained nature of the devices, in concurrence with the unrelenting need for smaller size, lower power and lower output prices.

In this paper, we focus on the TEA (block cipher) which allows feasibility for the key generation and these generated keys are used for cryptographic applications with reduced hardware complexity.

An emerging requirement of the cryptographic algorithm is the security and a very clever design of an algorithm is essential.

In this paper, architecture and the FPGA implementation of the Tiny encryption Algorithm (TEA) and a Key generation using Linear Feedback are proposed. The system works for the both encryption and decryption processes and has been optimized for low hardware resources and for high–speed operation. The proposed architecture uses FPGA Spartan 3E and the performances are analyzed by implementing the KGU using LFSR with TEA which offers moderate security and simplicity in the implementation processes.

With the proposed architecture, we focus on proving that the TEA algorithm can easily be implemented in hardware. The report is formed as follows: In Section II the TEA algorithm and Key Generation Unit is described, a thorough analysis of the architecture and the FPGA implementation is created. Carrying out analysis of the architectures is made in Section III and finally some conclusions are given in Section IV.

II. TINY ENCRYPTION ALGORITHM

The Tiny Encryption Algorithm (TEA) is simple and small, but fast and cryptographically strong. The original spec was designed by Wheeler and Needham.The Tiny Encryption Algorithm (TEA) is a block cipher notable for its simplicity of description and implementation, typically a few lines of code. TEA has fast execution time, and needs minimal storage space. TEA is very secure. There accept been no known successful crypt analyses of TEA. It's conceived to be as secure as the IDEA algorithm, designed by Massey and Xuejia Lai. It uses the same mixed algebraic groups technique as IDEA, but it's very much simpler, hence faster.

TEA is a Feistel cipher which uses operations from mixed (orthogonal) algebraic groups - XOR, ADD and SHIFT in this instance. This is a really ingenious way of providing Shannon's twin properties of diffusion and confusion which are necessary for a secure block cipher, without the explicit need for P-boxes and S-boxes respectively. It encrypts 64 data bits at a time using a 128-bit key. It seems highly resistant to differential cryptanalysis, and achieves complete diffusion (where a one bit difference in the plaintext will cause about 32 bit differences in the cipher text) after only six cycles.

The mixing portion of TEA seems unbroken but related key attacks are possible, even though the construction of 2^{32} text under two related keys seems impractical is one of the weaknesses. The second failing is that the effective length of the keys is 126 bits not 128 does affect certain potential applications but not the simple cipher decipher mode.

A. Encryption Methodology

At the encryption site, TEA takes 64 (block size) data bits time using a 128-bit key with 32 rounds. TEA is an iteration cipher, where each round i has plain text inputs P0 [i-1] and P1 [i-1], which is derived from the previous round. The subkey K[*i*] is derived from the 128 bit overall K and it uses a constant delta (∂), is the derivative of the golden number ratio to ensure that the sub keys are distinct. The figure 1 shows the architecture for TEA

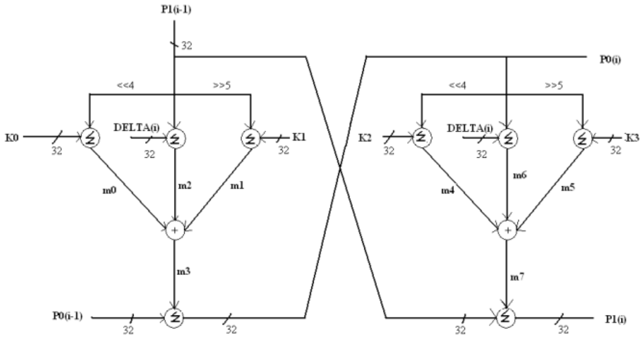


Fig -1: TEA encryption process

The value of the delta function is defined as

∂ = (√5 −1) * 2^31 = 9E3779B9h

The outputs of each iteration are given by

P0[*i*] = P0[*i-1*] ∑ F(P1[*i-1*], K[0, 1], DELTA[*i*])

P1[*i*] = P1[*i-1*] ∑ F(P0[*i-1*], K[2, 3], DELTA[*i*])

The round function F is defined by

F([P,K[j,k],DELTA[i]) = ((P<<4) ∑ K[j]) XOR (P ∑DELTA[i]) XOR ((P>>5) ∑K[k])

The single TEA round function performs the simple mixed orthogonal algebraic functions such as Right/Left shifts, Integer addition and exclusive – or operations.

The steps carried out in round function:

- The one half P1 [i-1] of the block cipher is Left shifted by 4 times and Right shifted 5 times.
- The left shifted block is added with the sub key K0 and right shifted block is added with the sub key K1.
- It is also added to the constant delta value DELTA[i] which is the multiples of delta, where i represents the number of iterations.
- The results are then Ex–ORed and added with the other half of the block cipher P0[i-1] which produces one half of the block cipher MO for the next iteration.
- Similar operations are performed for the next half round function with the above result.

Finally, the Ex–OR ed result is added with the first half of the block cipher P1[i-1] to produce the next half block M1 for the further rounds.

B. Decryption Methodology

Similar operations are performed for decryption process which is described in figure 2.

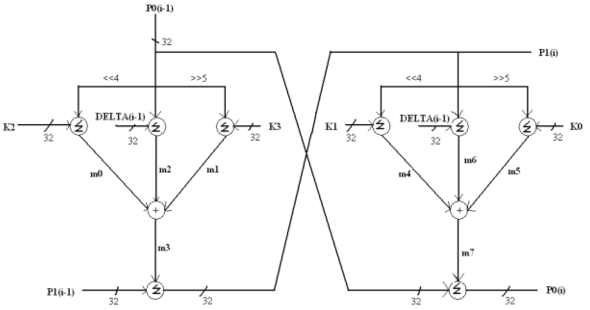


Fig -2: TEA decryption process

In this case, the constant delta value DELTA(i-1) is "C6EF3720", where 'i' represents the number of iterations. In each iteration, the delta value "9E3779B9" is subtracted with the constant delta value. To decrypt the encrypted data the reverse operation of the encryption process can be done since TEA uses Fiestel structure.

C. Key generation using LFSR

In this proposed technique we generate the key sequences using pseudorandom number generator using the linear feedback shift register (LFSR) as shown in Figure 3.

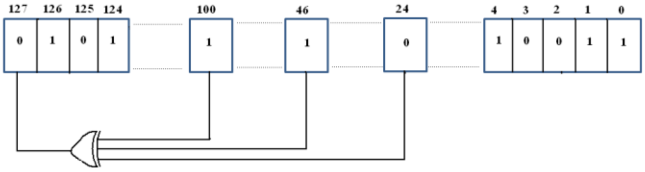


Fig -3: Key generation using a linear feedback shift register

The LFSR is a shift register which initially takes a starting seed value of 128 bits which acts as the secret key between the sender and the receiver and is driven by the XORing of some bits of the overall shift register value as shown. The positions in the LFSR which affect the next state are called as the tap position which also acts as the secret key between the sender and the receiver. The resultant output obtained by XORing the 100, 46, 24th bits and is then fed back to the 127th bit this generates a new key stream each time.

III. PERFORMANCE ANALYSIS AND RESULTS

The proposed architecture has been implemented by using Verilog. All the internal components of the design were synthesized placed and routed using XILINX FPGA devices. Figure 4 shows the simulation result for both encryption and decryption.

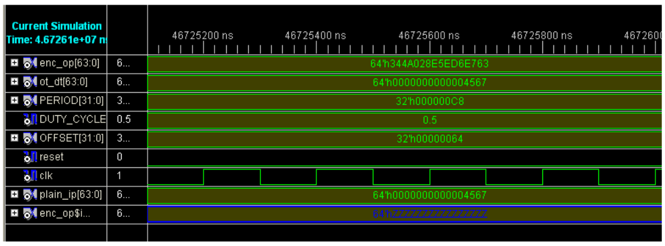
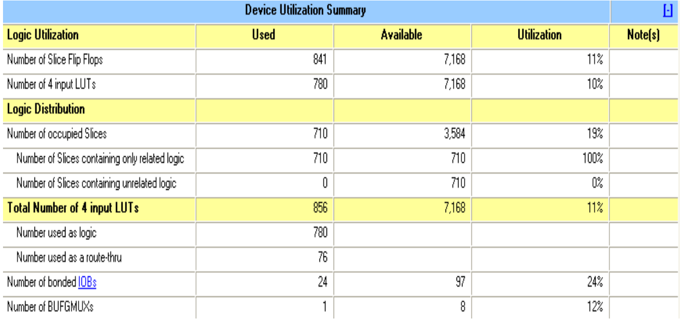

```
Current Simulation
Time: 4.67261e+07 ns        46725200 ns   46725400 ns   46725600 ns   46725800 ns   4672600
enc_op[63:0]      6...   64'h344A028E5ED6E763
ot_dt[63:0]       6...   64'h0000000000004567
PERIOD[31:0]      3...   32'h000000C8
DUTY_CYCLE        0.5    0.5
OFFSET[31:0]      3...   32'h00000064
reset             0
clk               1
plain_ip[63:0]    6...   64'h0000000000004567
enc_op$i...       6...   64'hZZZZZZZZZZZZZZZZ
```

Fig.4: Simulation result for encryption and decryption.

FPGA implementation of this design has been done using Xilinx XC3s400-5tq144, the corresponding device utilization summary is tabulated in TABLE 1. The place and route were done using Xilinx ISE 8.1.i package. The final results are given in Table 3 for Spartan-3E



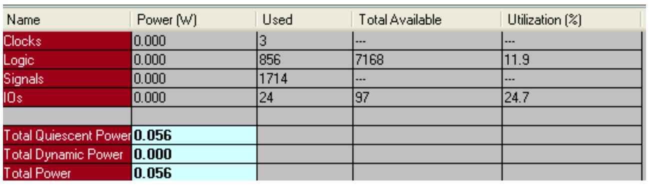
Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	841	7,168	11%	
Number of 4 input LUTs	780	7,168	10%	
Logic Distribution				
Number of occupied Slices	710	3,584	19%	
Number of Slices containing only related logic	710	710	100%	
Number of Slices containing unrelated logic	0	710	0%	
Total Number of 4 input LUTs	856	7,168	11%	
Number used as logic	780			
Number used as a route-thru	76			
Number of bonded IOBs	24	97	24%	
Number of BUFGMUXs	1	8	12%	

Table 1: TEA architecture synthesis results.

The power estimation for fully sub-pipelined architecture of 128 bits-length, having 32 round units using XPower Analyzer has been done and results are tabulated in TABLE2.

Figure 5 shows the real time implementation using FPGA and 7 segment LED.

The throughput, low-cost and flexibility of our solution make it perfectly practical for cryptographic embedded applications.



Name	Power (W)	Used	Total Available	Utilization (%)
Clocks	0.000	3	---	---
Logic	0.000	856	7168	11.9
Signals	0.000	1714	---	---
IOs	0.000	24	97	24.7
Total Quiescent Power	**0.056**			
Total Dynamic Power	**0.000**			
Total Power	**0.056**			

Table 2: Power Analysis using XPower Analyzer .

Fig -3: Real time implementation using FPGA and 7 segment LED.

Table 3 shows the comparison results of Tiny encryption Algorithm using idea encryption. It shows that TEA is faster and better throughput and lesser area.

Algorithm	Clock Frequency (MHz)	Throughput (Mbps)	Area (mm^2)
TEA	59.3MHz	19.52	1.25
IDEA	37	215	1.54

Table 3: Comparision of TEA with IDEA .

IV. CONCLUSIONS

In this paper, we have presented an FPGA implementation of efficient pipelined TEA architecture which includes both encryption and decryption. Also sub pipelining architecture helped us to get higher throughput than earlier implementations. The design is modeled using *Verilog HDL* and simulated with the help of *Xilinx ISE*. Synthesis is done by using Spartan 3E XC3s400. The TEA has been implemented using Spartan XC3s400 so that the cost can be reduced . The application includes various wired and wireless communications which require secure data transfer

REFERENCES

[1] Thomas Eisenbarth, Sandeep Kumar,Christof Paar and Axel Poschmann and Leif Uhsadel " A Survey of Lightweight-Cryptography Implementations" IEEE Design & Test of Computers-November–December 2007.

[2] Rahul Ranjan and I. Poonguzhali, "VLSI Implementation of IDEA Encryption Algorithm"- *Mobile and Pervasive Computing (CoMPC–2008)*

[3] Bhasker J., Verilog VHDL Synthesis, A Practical Primer,Star Galaxy Publishing, Lucent Technologies, 1998.

[4] Senthil Kumar, Manjupriya.M- "Microcontroller Based Cryptosystem With Key Generation Unit" -International Conference on Computer, Communication and Electrical Technology – ICCCET2011, 18th & 19th March, 2011.

[5] P.Israsena, Thailand IC Design Incubator (TIDI) National Electronics & Computer Technology Center (NECTEC), "Design and implementation of low power hardware encryption for low cost secure RFID using TEA", 2005 IEEE ICICS.

[6] Xilinx. Spartan-3E data FPGA family data sheet DS312 July 19-2013,available from http://www.xilinx.com.

[7] Issam Damaj, Samer Hamade and Hassan Diab "Efficient tiny hardware cipher under Verilog", in High Performance Computing & Simulation Conference, 2008.

[8] M. Thaduri, S. -M. Yoo*, R. Gaede, "An efficient VLSI implementation of the IDEA encryption algorithm using VHDL" Microprocessors and Microsystems 29 (2005) 1–7-ELSEVIER.

[9] Shruti Hathwalia, Meenakshi Yadav, "Design and Analysis of 32 bit Linear Feedback Shift Register using VHDL"- International Journal of Engineering Research and Applications.Vol.4 Issue 6,June-2014 pp99-102.