A NEW LIGHTWEIGHT CRYPTOGRAPHIC ALGORITHM

Leelavathy S R[1], Jothika S[2], Keerthana K[3], Amrutha A[4], Mamatha B R[5].
[1,2,3,4,5]Department of Computer Science and Engineering, Dr TTIT,KGF, Kolar Karnataka, India
leela48@gmail.com[1], jothikas208@gmail.com[2], keerthanak30398@gmail.com[3],
sweetieammu88@gmail.com[4], mamthaa.6066@gmail.com[5]

ABSTRACT

In this present world there is lot of importance in transferring and securing the digital data. But there is problem in securing the data, to avoid that we are going to use a combined algorithm. In this paper, we discuss about a hybrid algorithm using Hummingbird algorithm[1] this algorithm is implemented in vivado software, using verilog hardware description language.

Hummingbird algorithm is a latest ultra-lightweight cryptographic algorithm targeted for low cost smart devices. In this paper we are aiming to design an algorithm of low power and high speed, lightweight algorithm for securing data.

Keywords: encryption, lightweight cryptograph

1. INTRODUCTION

Cryptography is a method of securing information and it is used for encrypting and decrypting the information. Encryption is converting plain text to cipher text, where the translation of cipher text to the plain text is known as decryption. Cryptography is divided into secret key cryptography and public key cryptography and hash functions. In secret key cryptography[2] the same key will be used by the both sender and receiver for encryption and decryption. In public key cryptography the sender and receiver will be using dissimilar keys at both the sides Hummingbird algorithm is one of the recently proposed light weight cryptographic algorithms targeted for resource constrained devices like smart cards. The Hummingbird algorithm is the one of the recently presented ultra-light weight cryptographic algorithm.The size of the key and the internal state of Hummingbird provides adequate security level for many embedded applications.

2. Overview of Hummingbird

The design of hummingbird consists of 16-bit block size, 256-bit key size, and 80-bit internal state. The hummingbird algorithm[5] has good efficiency compared to all other algorithms and also it has smallest block size compared to all other algorithms. The the design of hummingbird algorithm includes initialization, encryption and decryption process.

Hummingbird Algorithm is as follows.

Input: A 16-bit data block m=(m0, m1 …… m15) and a 64-bit subkey k(i) such that k(i)=k1||k2||k3||k4

Output: A 16-bit data block m\`=(m\`0, m\`1 …… m\`15)

```
for j=1 to 4 do
    m<- m ⊕ kj
    A=m0||m1||m2||m3, B=m4||m5||m6||m7, C=m8||m9||m10||m11 and D=m12||m13||m14||m15
    m<- S1(A)||S2(B)||S3(C)||S4(D)
    m<- m ⊕ (m<<6) ⊕ (m<<10)
end for
m<- m ⊕ k1 ⊕ k3
A=m0||m1||m2||m3, B=m4||m5||m6||m7, C=m8||m9||m10||m11 and D=m12||m13||m14||m15
m<- S1(A)||S2(B)||S3(C)||S4(D)
m`<- m ⊕ k2 ⊕ k4
return m`=(m`0, m`1 …… m`15)
```

3. Processing of the message

The algorithm which is used for processing of the padded message is described next. First, the padded message needs to be divided into 512-bit blocks, denoted here as Mj where $j \geq 0$ is the index of the block. The algorithm processes one Mj at once, starting from $M0$, until all Mj have been processed. Five

32-bit registers, *A*, *B*, *C*, *D* and *E* are defined. At the beginning of processing of each *Mj* their values are set as follows: $A \leftarrow H0$, $B \leftarrow H1$, etc.

The algorithm consists of 80 steps. Let *t* denote the index of a step, i.e. $0 \leq t \leq 79$. First, a 32-bit message block *Wt* is derived for every step *t* from the 512-bit message block *Mj* using a message schedule. For $t < 16$, *Wt* is simply the *t*th 32-bit word of *Mj*. When $t \geq 16$, *Wt* are derived recursively with the following formula:$Wt = (Wt-3 \oplus Wt-8 \oplus Wt-14 \oplus Wt-16) <<< 1$

4. Hummingbird algorithm

Network security[3] consists of the policies and practices adopted to prevent and monitor unauthorized access. Cryptography is a technique in which we can encrypt data into cipher text for its secure transmission. Light weight cryptography: is a cryptographic algorithm tailored for implementation in constrained environments. Public key Cryptography uses same key for both encryption and decryption of data. Secret Key cryptography uses two keys private and public for communication between the sender and the receiver.

Hash function is a function which maps variable size data into fixed size data. Authentication is a form of encryption which simultaneously provides confidentiality, integrity. Network security involves the authorization of access to data in a network. Network security[6] covers a We are implementing a hybrid Cryptosystem with a new light weight Cryptographic Algorithm. Objective is to provide Authentication using Humming bird algorithm and hash will definitely lead to higher level of security. Hash functions have a significant role in cryptography. It maps messages of arbitrary length into a fixed length digest.

They were used to check the integrity of message initially but now they are being used in each and every field of online World They are the building blocks of today's digital world. The online banking transactions would not be secured enough without hash functions. Humming bird can be implemented with very small hardware and suitable for providing security in low cost ubiquitous devices.

The Encryption block uses the initialized status registers to encrypt the plain text into cipher text. Encryption undergoes the modulo 2^16 addition of register RS1 and plain text and undergoes through the block encryption using the secret key. This is repeated four times and the resulting cipher text is given to the decryption module making the encryption complete signal high. The decryption is just reverse of encryption. The input is cipher text and output is plain text Modulo 2^16 subtraction is used in decryption. Thus when both decryption complete and encryption complete signals are high the output is given in one clock cycle. The registers are updated for the next process.

The working of hummingbird algorithm. The 16-bit plaintext is given as the input along with the 256-bit key, which is segmented into four 64-bit subkeys. There are three blocks which perform initialization, encryption and decryption. In the initialization process, the registers are updated for encryption. After each set of plaintext and cipher text . the internal status registers are updated by the 16-bit LFSR. The initialization block and the encryption block consist of the block substitution box or S-Box and the linear permutation. The message is encrypted by the tag using the encryption process and the result as cipher text is sent to the reader. The reader decrypts the message using the same key

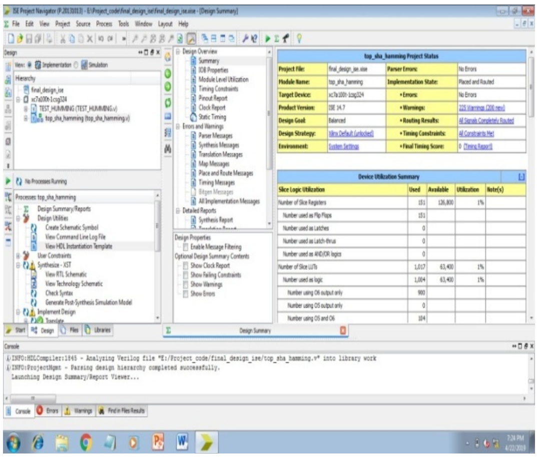
This architecture works on the encryption only and decryption only processor. The initialization block consists of four status registers and a ready pin. A 5-bit counter is used to count the number of clock cycles. When the data ready pin goes high, the counter starts counting and the status registers are initialized with some random values. The registers are then updated by undergoing encryption round of block cipher in next 16 clock cycles.

The Encryption block uses the initialized status registers to encrypt the plain text into cipher text. This is repeated four times and the resulting cipher text is given to the decryption module making the enc complete signal high. The decryption is just reverse of encryption. The input is cipher text and output is plain text.

Plain text refers to any message that is not encrypted. The data from plain text will be imported and the imported data will be given as input to the Encryption algorithm Hummingbird and Hash function. The encrypted code contains Cipher and Hash code.

The goal of encryption algorithm is to make it as difficult as possible to decrypt the generated cipher text without using the key. If a real good encryption algorithm is used, there is no technique better than methodology trying every possible keys, the longer the key, is more difficult to decrypt piece of cipher text without the key.

5. SIMULATION & RESULT

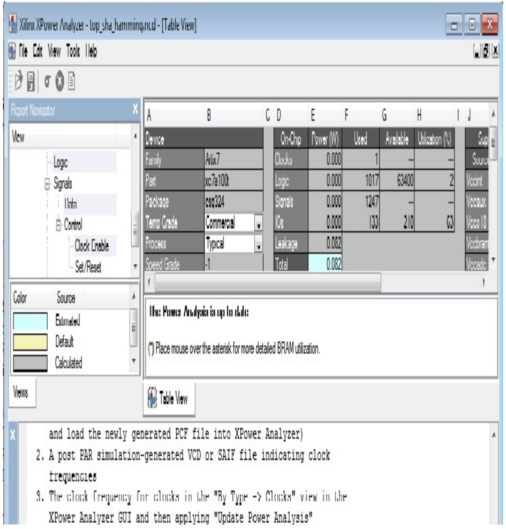


top_sha_hamming Project Status			
Project File:	final_design_ise.xise	Parser Errors:	No Errors
Module Name:	top_sha_hamming	Implementation State:	Placed and Routed
Target Device:	xc7a100t-1csg324	• Errors:	No Errors
Product Version:	ISE 14.7	• Warnings:	225 Warnings (200 new)
Design Goal:	Balanced	• Routing Results:	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	All Constraints Met
Environment:	System Settings	• Final Timing Score:	0 (Timing Report)

Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	151	126,800	1%	
Number used as Flip Flops	151			
Number used as Latches	0			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	1,017	63,400	1%	
Number used as logic	1,004	63,400	1%	
Number using O6 output only	900			
Number using O5 output only	0			
Number using O5 and O6	104			

```
INFO:HDLCompiler:1845 - Analyzing Verilog file "E:/Project_code/final_design_ise/top_sha_hamming.v" into library work
INFO:ProjectMgmt - Parsing design hierarchy completed successfully.
Launching Design Summary/Report Viewer...
```

Fig 1: working of vivado



Xilinx XPower Analyzer - top_sha_hamming.ncd - [Table View]

Device		On-Chip	Power (W)	Used	Available	Utilization (%)
Family	Artix7	Clocks	0.000	1	---	---
Part	xc7a100t	Logic	0.000	1017	63400	2
Package	csg324	Signals	0.000	1247	---	---
Temp Grade	Commercial	IOs	0.000	133	210	63
Process	Typical	Leakage	0.082			
Speed Grade	-1	Total	0.082			

The Power Analysis is up to date.

(*) Place mouse over the asterisk for more detailed BRAM utilization.

```
   and load the newly generated PCF file into XPower Analyzer)
2. A post PAR simulation-generated VCD or SAIF file indicating clock
   frequencies
3. The clock frequency for clocks in the "By Type -> Clocks" view in the
   XPower Analyzer GUI and then applying "Update Power Analysis"
```

Fig2 : inputs to Vivado

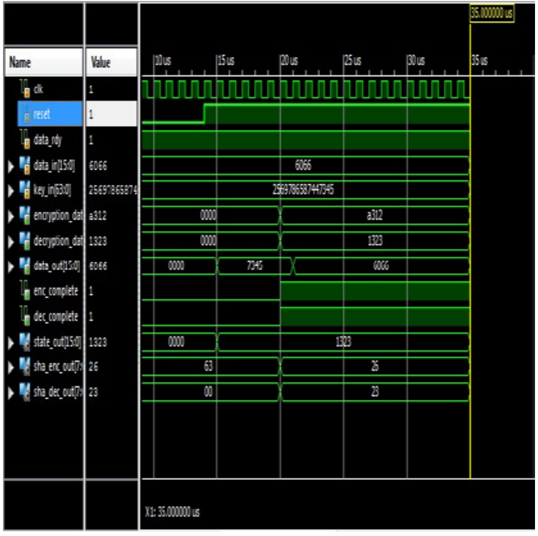


Fig3: Efficiency of humming bird

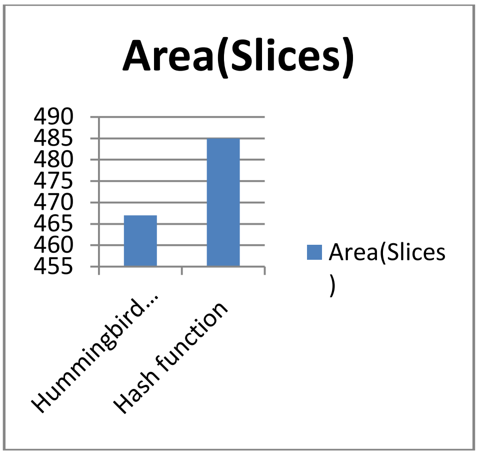


Fig4:Performance of Area in Hummingbird

Performance of Efficiency in Hummingbird

The results shows that the algorithm with Hashing can provide higher throughput and frequency than X.Fan and Bio Min by with the cost of higher resources. The Hash module that used for key authentication and security will take total area of 3659.673666 with power 184.637611

The above table shows the comparison of the proposed secured hummingbird design with key authentication using hash function with the design without the hash module. The design shows that the algorithm having improved Security and throughput with cost of small increase in area. This design has less power Consumption than the previous design with anefficiency of 0.3533

5. CONCLUSION

The design of Hummingbird Cryptographic Algorithm is based on an well-designed combination of a block cipher and stream cipher with 16 bit input size, 64-bit key size. Th key size of Hummingbird provides a security level which is suitable for resource constrained devices. The use of Hash algorithm SHA-1 make more secure. Various papers are discussed about hummingbird cryptographic algorithm on different platform like microcontroller based on ASIC, sparton-2 FPGA, sparton-3 FPGA, etc. In all of these, there is an enhanced research on reducing area, power requirement, & increasing speed with aim of giving better security to resource constrained devices like sensor nodes .

REFERENCES

[1] Abdullah, A. M., & Aziz, R. H. H. (2016, June). New Approaches to Encrypt and Decrypt

Data in Image using Cryptography and Steganography Algorithm., International Journal of Computer Applications, Vol. 143, No.4 (pp. 11-17).

[2] J. P. Aumasson, Quark: A Lightweight Hash, 2012.

[3] F. Xinxin, H. Honggang, G. Guang, E. M. Smith, and D. Engels, "Lightweight implementation of Hummingbird cryptographic algorithm on 4-bit microcontrollers," in internet Technology and Secured Transactions, 2009. ICITST 2009. International Conference for, 2009.

[4] K. Finkenzeller, RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification. Hoboken, NJ, USA: Wiley,2003

[5] Beaulieu, Ray, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan WeeksWeeks, and Louis Wingers. "The SIMON and SPECK Families of Lightweight Block Ciphers." Cryptology EPrint Archive. International Association for Cryptologic Research, 19 June 2013. Web. 11 Mar. 2015..

[6] Constantin, J., Burg, A., & Gürkaynak, F. K.(2012). Investigating the Potential of Custom Instruction Set Extensions for SHA-3Candidates on a 16-bit MicrocontrollerArchitecture. IACR Cryptology ePrint Archive,2012, 50.

[7] Xinxin Fan; Guang Gong; Lauffenburger, Hicks,“FPGA of the Hummingbird cryptographic algorithm”, 2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), pp.48 14 June 2010.

[8] X. Fan, G. Gong, K. Lauffenburger, and T. Hicks,“FPGA Implementations of the Hummingbird Cryptographic Algorithm”, IEEE International Symposium on Hardware-Oriented Security and Trust(HOST), 2010